2 – Navigation



## – Navigation

The first thing we need to learn (besides just typing) is how to navigate the file system 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 File System 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 file system is called the *root direc- tory*. The root directory contains files and subdirectories, which contain more files and subdirectories and so on and so on.

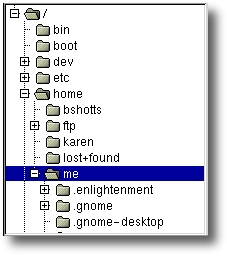
Note that unlike Windows, which has a separate file system tree for each storage device, Unix-like systems such as Linux always have a single file system tree, regardless of how many drives or storage devices are attached to the computer. Storage devices are attached (or more correctly, *mounted*) at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

### The Current Working Directory

Most of us are probably familiar with a graphical file manager which represents the file system tree as in Figure 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 file system tree we need to think of it in a different way.

Imagine that the file system is a maze shaped like an upside-down tree and we are able to



*Figure 1: File system tree as shown by a graphical file manager*

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.



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 and it is the only place a regular user is allowed to write files.

### Listing The Contents Of A Directory

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



Listing The Contents Of A Directory

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

### Changing The Current Working Directory

To change your working directory (where we are standing in our tree-shaped maze) we use the cd command. To do this, type cd followed by the *pathname* of the desired work- ing 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 different 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 the directory is usr/bin/. This means from the root directory (represented by the leading slash in the pathname) there is a directory called "usr" which contains a di- rectory called "bin".



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 file system 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:



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:



Or, with a relative pathname:



Two different methods with 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 using an absolute pathname:



Or, with a relative pathname:



Now, there is something important that I must point out here. In almost all cases, you can

Changing The Current Working Directory omit the "./". It is implied. Typing:

does the same thing. In general, if you do not specify a pathname to something, the work- ing 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*

cd Changes the working directory to your home directory.



cd - Changes the working directory to the previous working directory.



cd ~*user\_name* Changes the working directory to the home directory of *user\_name*. For example, cd ~bob will change the directory to the home directory of user “bob.”







### Summing Up

In this chapter we saw how the shell treats the directory structure of the system. We learned about absolute and relative pathnames and the basic commands that are used to move about that structure. In the next chapter we will use this knowledge to go on a tour of a modern Linux system.

**Screenshots:**