

Looking Around

Now that you know how to move from working directory to working directory, we're going to take a tour of your Linux system and, along the way, learn some things about what makes it tick. But before we begin, I have to teach you some tools that will come in handy during our adventure. These are:

- [ls](#) (list files and directories)
- [less](#) (view text files)
- [file](#) (classify a file's contents)

ls

The `ls` command is used to list the contents of a directory. It is probably the most commonly used Linux command. It can be used in a number of different ways. Here are some examples:

Examples of the ls command	
Command	Result
<code>ls</code>	List the files in the working directory
<code>ls /bin</code>	List the files in the /bin directory (or any other directory you care to specify)
<code>ls -l</code>	List the files in the working directory in long format
<code>ls -l /etc /bin</code>	List the files in the /bin directory and the /etc directory in long format
<code>ls -la ..</code>	List all files (even ones with names beginning with a period character, which are normally hidden) in the parent of the working directory in long format

These examples also point out an important concept about commands. Most commands operate like this:

command -options arguments

where *command* is the name of the command, *-options* is one or more adjustments to the command's behavior, and *arguments* is one or more "things" upon which the command operates.

In the case of **ls**, we see that **ls** is the name of the command, and that it can have one or more options, such as **-a** and **-l**, and it can operate on one or more files or directories.

A Closer Look At Long Format

If you use the `-l` option with `ls`, you will get a file listing that contains a wealth of information about the files being listed. Here's an example:

```

-rw----- 1 bshotts bshotts      576 Apr 17 1998 weather.txt
drwxr-xr-x 6 bshotts bshotts    1024 Oct  9 1999 web_page
-rw-rw-r-- 1 bshotts bshotts  276480 Feb 11 20:41 web_site.tar
-rw----- 1 bshotts bshotts     5743 Dec 16 1998 xmas_file.txt

```

The diagram illustrates the structure of the file listing output. It consists of a table with six columns and six rows. The columns are labeled as follows:

- File Name**: The first column, containing file names like `weather.txt`, `web_page`, `web_site.tar`, and `xmas_file.txt`.
- Modification Time**: The second column, containing dates and times like `Apr 17 1998`, `Oct 9 1999`, `Feb 11 20:41`, and `Dec 16 1998`.
- Size (in bytes)**: The third column, containing file sizes like `576`, `1024`, `276480`, and `5743`.
- Group**: The fourth column, containing group names like `bshotts`.
- Owner**: The fifth column, containing owner names like `bshotts`.
- File Permissions**: The sixth column, containing permission strings like `-rw-----`, `drwxr-xr-x`, `-rw-rw-r--`, and `-rw-----`.

The diagram uses dashed lines to connect the column headers to the corresponding data in the table. The headers are located at the bottom of the table, and the data is located at the top. The connections are as follows:

- File Name** connects to the first column.
- Modification Time** connects to the second column.
- Size (in bytes)** connects to the third column.
- Group** connects to the fourth column.
- Owner** connects to the fifth column.
- File Permissions** connects to the sixth column.

File Name

The name of the file or directory.

Modification Time

The last time the file was modified. If the last modification occurred more than six months in the past, the date and year are displayed. Otherwise, the time of day is shown.

Size

The size of the file in bytes.

Group

The name of the group that has file permissions in addition to the file's owner.

Owner

The name of the user who owns the file.

File Permissions

A representation of the file's access permissions. The first character is the type of file. A "-" indicates a regular (ordinary) file. A "d" indicates a directory. The second set of three characters represent the read, write, and execution rights of the file's owner. The next three represent the rights of the file's group, and the final three represent the rights granted to everybody else. I'll discuss this in more detail in a later lesson.

less

less is a program that lets you view text files. This is very handy since many of the files used to control and configure Linux are human readable.

What is "text"?

There are many ways to represent information on a computer. All methods involve defining a relationship between the information and some numbers that will be used to represent it. Computers, after all, only understand numbers and all data is converted to numeric representation.

Some of these representation systems are very complex (such as compressed multimedia files), while others are rather simple. One of the earliest and simplest is called *ASCII text*. [ASCII](#) (pronounced "As-Key") is short for American Standard Code for Information Interchange. This is a simple encoding scheme that was first used on Teletype machines to map keyboard characters to numbers.

Text is a simple one-to-one mapping of characters to numbers. It is very compact. Fifty characters of text translates to fifty bytes of data. Throughout a Linux system, many files are stored in text format and there are many Linux tools that work with text files. Even the legacy operating systems recognize the importance of this format. The well-known NOTEPAD.EXE program is an editor for plain ASCII text files.

The **less** program is invoked by simply typing:

```
less text_file
```

This will display the file.

Controlling less

Once started, **less** will display the text file one page at a time. You may use the Page Up and Page Down keys to move through the text file. To exit **less**, type "q". Here are some commands that **less** will accept:

Command	Action
Page Up or b	Scroll back one page
Page Down or space	Scroll forward one page
G	Go to the end of the text file
1G	Go to the beginning of the text file
/characters	Search forward in the text file for an occurrence of the specified <i>characters</i>
n	Repeat the previous search
h	Display a complete list less commands and options
q	Quit

file

As you wander around your Linux system, it is helpful to determine what kind of data a file contains before you try to view it. This is where the **file** command comes in. **file** will examine a file and tell you what kind of file it is.

To use the **file** program, just type:

```
file name_of_file
```

The **file** program can recognize most types of files, such as:

Various kinds of files

File Type	Description	Viewable as text?
ASCII text	The name says it all	yes
Bourne-Again shell script text	A bash script	yes
ELF 32-bit LSB core file	A core dump file (a program will create this when it crashes)	no
ELF 32-bit LSB executable	An executable binary program	no
ELF 32-bit LSB shared object	A shared library	no
GNU tar archive	A tape archive file. A common way of storing groups of files.	no, use tar tvf to view listing.
gzip compressed data	An archive compressed with gzip	no
HTML document text	A web page	yes
JPEG image data	A compressed JPEG image	no
PostScript document text	A PostScript file	yes

RPM	A Red Hat Package Manager archive	no, use rpm -q to examine contents.
Zip archive data	An archive compressed with zip	no

While it may seem that most files cannot be viewed as text, you will be surprised how many can. This is especially true of the important configuration files. You will also notice during our adventure that many features of the operating system are controlled by shell scripts. In Linux, there are no secrets!