A Guided Tour

It's time to take our tour. The table below lists some interesting places to explore. This is by no means a complete list, but it should prove to be an interesting adventure. For each of the directories listed below, do the following:

- **`cd`** into each directory.
- Use **`ls`** to list the contents of the directory.
- If there is an interesting file, use the **`file`** command to determine its contents.
- For text files, use **`less`** to view them.

Directory	Description
`/`	The root directory where the file system begins. The root directory will probably contain only subdirectories.
`/boot`	This is where the Linux kernel and boot loader files are kept. The kernel is a file called **`vmlinuz`**.
`/etc`	The **`/etc`** directory contains the configuration files for the system. All of the files in **`/etc`** should be text files. Some points of interest are: **`/etc/passwd`** The **`passwd`** file contains the essential information for each user. This is where user accounts are defined. **`/etc/fstab`** The **`fstab`** file contains a table of devices that get mounted when the system boots. This file defines the system's disk drives. **`/etc/hosts`** This file lists the network host names and IP addresses that are intrinsically known to the system. **`/etc/init.d`** This directory contains the scripts that start various system services at boot time.
`/bin, /usr/bin`	These two directories contain most of the programs for the system. The **`/bin`** directory has the essential programs that the system requires to operate, while **`/usr/bin`** contains applications for the system's users.
`/sbin, /usr/sbin`	The **`sbin`** directories contain programs for system administration, mostly for use by the superuser.
`/usr`	The **`/usr`** directory contains a variety of things that support user applications. Some highlights: **`/usr/share/X11`** Support files for the X Window system **`/usr/share/dict`**

	Dictionaries for the spelling checker. Yes, Linux comes with a spelling checker. See **look** and **aspell**. **/usr/share/doc** Various documentation files in a variety of formats. **/usr/share/man** The man pages are kept here.
/usr/local	**/usr/local** and its subdirectories are used for the installation of software and other files for use on the local machine. What this really means is that software that is not part of the official distribution (which usually goes in **/usr/bin**) goes here. When you find interesting programs to install on your system, they should be installed in one of the **/usr/local** directories. Most often, the directory of choice is **/usr/local/bin**.
/var	The **/var** directory contains files that change as the system is running. This includes: **/var/log** Directory that contains log files. These are updated as the system runs. It's a good idea to view the files in this directory from time to time, to monitor the health of your system. **/var/spool** This directory is used to hold files that are queued for some process, such as mail messages and print jobs. When a user's mail first arrives on the local system (assuming it has local mail, a rare occurrence on modern machines that are not mail servers), the messages are first stored in **/var/spool/mail**
/lib	The shared libraries (similar to DLLs in that other operating system) are kept here.
/home	**/home** is where users keep their personal work. In general, this is the only place users are allowed to write files. This keeps things nice and clean :-)
/root	This is the superuser's home directory.
/tmp	**/tmp** is a directory in which programs can write their temporary files.
/dev	The **/dev** directory is a special directory, since it does not really contain files in the usual sense. Rather, it contains devices that are available to the system. In Linux (like Unix), devices are treated like files. You can read and write devices as though they were files. For example **/dev/fd0** is the first floppy disk drive, **/dev/sda** is the first hard drive. All the devices that the kernel understands are represented here.
/proc	The **/proc** directory is also special. This directory does not contain files. In fact, this directory does not really exist at all. It is entirely virtual. The **/proc** directory contains little peep holes into the kernel itself. There are a group of numbered entries in this directory that correspond to all the processes running on the system. In addition, there are a number of named entries that permit access to the current configuration of the system. Many of these entries can be viewed. Try viewing **/proc/cpuinfo**. This entry will tell you what the kernel thinks of the system's CPU.

/media	Finally, we come to **/media**, a normal directory which is used in a special way. The **/media** directory is used for *mount points*. As we learned in the second lesson, the different physical storage devices (like hard disk drives) are attached to the file system tree in various places. This process of attaching a device to the tree is called *mounting*. For a device to be available, it must first be mounted. When your system boots, it reads a list of mounting instructions in the **/etc/fstab** file, which describes which device is mounted at which mount point in the directory tree. This takes care of the hard drives, but we may also have devices that are considered temporary, such as optical disks and USB storage devices. Since these are removable, they do not stay mounted all the time. The **/media** directory is used by the automatic device mounting mechanisms found in modern desktop oriented Linux distributions. To see what devices and mount points are used, type **mount**.

Interesting directories and their contents

A weird kind of file...

During your tour, you probably noticed a strange kind of directory entry, particularly in the **/lib** directory. When listed with **ls -l**, you might have seen something like this:

```
lrwxrwxrwx     25 Jul  3 16:42 System.map -> /boot/System.map-4.0.36-3
-rw-r--r-- 105911 Oct 13  2018 System.map-4.0.36-0.7
-rw-r--r-- 105935 Dec 29  2018 System.map-4.0.36-3
-rw-r--r-- 181986 Dec 11  2019 initrd-4.0.36-0.7.img
-rw-r--r-- 182001 Dec 11  2019 initrd-4.0.36.img
lrwxrwxrwx     26 Jul  3 16:42 module-info -> /boot/module-info-4.0.36-
-rw-r--r--  11773 Oct 13  2018 module-info-4.0.36-0.7
-rw-r--r--  11773 Dec 29  2018 module-info-4.0.36-3
lrwxrwxrwx     16 Dec 11  2019 vmlinuz -> vmlinuz-4.0.36-3
-rw-r--r-- 454325 Oct 13  2018 vmlinuz-4.0.36-0.7
-rw-r--r-- 454434 Dec 29  2018 vmlinuz-4.0.36-3
```

Notice the files, `System.map, module-info` and `vmlinuz`. See the strange notation after the file names?

Files such as this are called *symbolic links*. Symbolic links are a special type of file that points to another file. With symbolic links, it is possible for a single file to have multiple names. Here's how it works: Whenever the system is given a file name that is a symbolic link, it transparently maps it to the file it is pointing to.

Just what is this good for? This is a very handy feature. Let's consider the directory listing above (which is the `/boot` directory of an old system). This system has had multiple versions of the Linux kernel installed. We can see this from the files `vmlinuz-4.0.36-0.7` and `vmlinuz-4.0.36-3`. These file names suggest that both version 4.0.36-0.7 and 4.0.36-3 are installed. Because the file names contain the version it is easy to see the differences in the directory listing. However, this would be confusing to programs that rely on a fixed name for the kernel file. These programs might expect the kernel to simply be called `"vmlinuz"`. Here is where the beauty of the symbolic link comes in. By creating a symbolic link called `vmlinuz` that points to `vmlinuz-4.0.36-3`, we have solved the problem.

To create symbolic links, we use the **ln** command.

Further Reading

- To learn more about the organization of the Linux filesystem, consult the Filesystem Hierarchy Standard