Linux Directory Structure

Now that you are able to connect to the server and have been introduced to the interface you will be using, it's time to learn about the directory layout. Understanding the directory structure will help you in the future when you are searching for components on the system. It can help you answer questions like:

> *Where are programs located?*
>
> *Where do configuration files live?*
>
> *Where might I find the log files for this application?*

Common Directories

Here are the most common top level directories that you need to be aware of and may interact with as a normal user.

Dir	Description
`/`	The directory called "root." It is the starting point for the file system hierarchy. Note that this is not related to the root, or superuser, account.
`/bin`	Binaries and other executable programs.
`/etc`	System configuration files.
`/home`	Home directories.
`/opt`	Optional or third party software.
`/tmp`	Temporary space, typically cleared on reboot.
`/usr`	User related programs.
`/var`	Variable data, most notably log files.

Comprehensive Directory Listing

Here is a comprehensive list of top level directories that you may find on various Linux systems. Some subdirectories are included to help clearly define the purpose of the top level directory. You may never interact with many of these directories. Some of these directories will be on every system you encounter like `/usr`. Other directories are unique to specific Linux distributions. You can safely skim over this list and refer back to it if or when you have a practical need to do so.

Dir	Description
`/`	The directory called "root." It is the starting point for the file system hierarchy. Note that this is not related to the root, or superuser, account.
`/bin`	Binaries and other executable programs.
`/boot`	Files needed to boot the operating system.
`/cdrom`	Mount point for CD-ROMs.
`/cgroup`	Control Groups hierarchy.
`/dev`	Device files, typically controlled by the operating system and the system administrators.
`/etc`	System configuration files.
`/export`	Shared file systems. Most commonly found on Solaris systems.

`/home`	Home directories.
`/lib`	System Libraries.
`/lib64`	System Libraries, 64 bit.
`/lost+found`	Used by the file system to store recovered files after a file system check has been performed.
`/media`	Used to mount removable media like CD-ROMs.
`/mnt`	Used to mount external file systems.
`/opt`	Optional or third party software.
`/proc`	Provides information about running processes.
`/root`	The home directory for the root account.
`/sbin`	System administration binaries.
`/selinux`	Used to display information about SELinux.
`/srv`	Contains data which is served by the system.
`/srv/www`	Web server files.
`/srv/ftp`	FTP files.
`/sys`	Used to display and sometimes configure the devices and busses known to the Linux kernel.
`/tmp`	Temporary space, typically cleared on reboot. This directory can be used by the OS and users alike.
`/usr`	User related programs, libraries, and documentation. The sub-directories in /usr relate to those described above and below.
`/usr/bin`	Binaries and other executable programs.
`/usr/lib`	Libraries.
`/usr/local`	Locally installed software that is not part of the base operating system.
`/usr/sbin`	System administration binaries.
`/var`	Variable data, most notably log files.
`/var/log`	Log files.

Unix Specific Directories

Linux is often found in environments with other Unix variants. If you ever have a need to log into a Unix server you may see some of the following Unix specific directories.

Dir	Description
`/devices`	Device files, typically controlled by the operating system and the system administrators.
`/kernel`	Kernel and kernel modules. (Solaris)
`/platform`	Platform specific files. (Solaris)
`/rpool`	ZFS root pool directory. (Solaris)
`/net`	Used to mount external file systems. (HP-UX)
`/nfs4`	Used to mount the Federated File System domain root. (Solaris)
`/stand`	Files needed to boot HP-UX.

Note that you may encounter other top level directories that have not been listed above. However, those were most likely created by the system administrator.

Application Directory Structures

Applications can follow the same conventions employed by the operating system. Here is a sample directory structure of an application named `apache` installed in `/usr/local`.

Dir	Description
`/usr/local/apache/bin`	The application's binaries and other executable programs.
`/usr/local/apache/etc`	Configuration files for the application.
`/usr/local/apache/lib`	Application libraries.
`/usr/local/apache/logs`	Application log files.

Here is what it might look like if it was installed in `/opt`.

Dir	Description
`/opt/apache/bin`	The application's binaries and other executable programs.
`/opt/apache/etc`	Configuration files for the application.
`/opt/apache/lib`	Application libraries.
`/opt/apache/logs`	Application log files.

A common alternative to placing all the application subdirectories in `/opt/app-name` is to also use `/etc/opt/app-name` and `/var/opt/app-name`. Here is what that might look like for our example `apache` application.

Dir	Description
`/etc/opt/apache`	Configuration files for the application.
`/opt/apache/bin`	The application's binaries and other executable programs.
`/opt/apache/lib`	Application libraries.
`/var/opt/apache`	Application log files.

Sometimes applications that are not part of the standard operating system are installed in a shared manor and are not given their own subdirectory. For example, if apache was installed directly into `/usr/local` its binaries would live in `/usr/local/bin/` and its configuration would live in `/usr/local/etc`. Apache may not be the only locally installed software so it would share that space with the other installed applications.

Another common practice is to create a directory structure based on a company, organization, or team name. For example, if you work at the Acme Corporation you may find a directory named `/opt/acme` or `/usr/local/acme`. Sometimes scripts and utilities are installed directly in that structure and other times there are segregated into their own subdirectories. Here's an example.

Dir	Description
`/opt/acme`	Company top level directory.
`/opt/acme/bin`	Binary programs created by or installed by the Acme Corporation.

Alternatively you may see something like this.

Dir	Description
`/opt/acme`	Company top level directory.

`/opt/acme/apache`	The top level directory for Acme's installation of apache.
`/opt/acme/apache/bin`	The apache binary programs.

Here are variations on the same idea, but based on a team within the company.

Dir	Description
`/opt/web-team`	The web support team's top level directory.
`/opt/acme/web-team`	The web support team's top level directory.
`/usr/local/acme/web-team`	The web support team's top level directory.

Example Top Level Directory Listings

Here is a listing of the top level directories for a few different Linux servers. Listing files and directories with the `ls` command will be covered later in the course.

Red Hat Enterprise Linux 6 (RHEL)

```
[bob@rhel6 ~]$ ls -1 /
bin
boot
cgroup
dev
etc
home
lib
lib64
lost+found
media
mnt
opt
proc
root
sbin
selinux
srv
sys
tmp
usr
var
```

SUSE Linux Enterprise Server 11 (SLES)

```
[bob@sles11 ~]$ ls -1 /
bin
boot
dev
etc
home
lib
lib64
lost+found
media
mnt
opt
proc
root
sbin
selinux
srv
sys
```

```
tmp
usr
```

Ubuntu 12.04 LTS

```
[bob@ubuntu12 ~]$ ls -1 /
bin
boot
dev
etc
home
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
selinux
srv
sys
tmp
usr
var
```