Linux Overview

Sam Gallagher

18 Sept 2017

1 Flavors and Distros

2 Terminal

The **terminal** is one of the most valuable tools you have in linux. The terminal itself is a unix program called *tty*, which stands for *teletype* terminal. A teletypewriter was the predecessor to the modern keyboard, and the terminology has stuck ever since. Because a proper teletype terminal would involve a teletypewriter and video terminal, modern terminals are actually *terminal emulators*.

There are many terminal emulators available, depending on your distribution of linux. You can easily install a new terminal emulator with extra features if you are not satisfied with your distribution's default terminal. Some of the most common terminal emulators are:

- GNOME Terminal (Default for GNOME)
- Konsole (Default for KDE)
- xterm (Default for X Window System)
- rxvt
- Guake
- Yakuake
- Xfce
- mintty (Default Cygwin terminal)

The first three in the list are default terminals that come preinstalled, while the rest are popular for the features and workflow they provide. For example, drop down terminals are activated with a particular hotkey, some terminals allow multiple screens, etc. The term *console* can generally be used interchangeably with terminal, the difference being the history of the terms.

A **shell** is a generic term for an interface which a user sends commands through. In unix (and therefore linux) a shell typically refers to the software interface between the user and the operating system kernel and other low level components. The shell acts as a scripting language and a command line language, and it works very closely with the terminal emulator to allow user interaction with the system. For example, when a user presses a key in a terminal emulator, the terminal receives the keypress and the shell converts this into a command (such as backspace, or appending a letter). Essentially, the shell receives parameters from devices or programs, and issues commands to a terminal or to the operating system.

The most common shell in unix is the **bash** shell which is derived from the Bourne shell. When a command is issued to the terminal such as

```
$ tar -xvf mycompfile.tar
```

The command is supplied in Bash format, with a command (tar), flags (-x, -v, and -f) and an input file (mycompfile.tar). A bash script is shell script, a file with commands supplied in the bash language. This begins with the location of the bash program, and then continues with the commands.

```
#!/bin/sh
echo "Hello world"
```

Note that the bash program is called ”sh” and is located in ”/bin/”. We'll talk more about files and the filesystem in the next section.

There are many commands used in the terminal to navigate the file system, modify files and programs, change permissions, and so on. Some of the most fundamental commands are:

- cd, change directory
- ls, list files
- pwd, print working directory
- man, read the manual for a program
- cp, copy a file
- mv, move a file
- mkdir, Make a directory
- chmod, change mode (for a file or folder, refers to permissions)
- chown, change owner
- rm, remove a file
- cat, stands for ”concatenate”, prints out a file
- grep, search a file or output
- wget, web get, downloads a remote file
- less, prints a file in its own viewer to browse without taking up terminal space
- tail, prints the tail, or end, of a file or output
- w, lists active users
- sudo, stands for superuser do, gives admin privilege for the current command

Familiarizing yourself with these commands and their options is essential to using linux, or any unix-based operating system.

2.1 Commands

2.1.1 cd

Change directory.

Usage:

```
cd
cd [directory]
cd ~
cd .
cd ./
cd ..
cd ../
cd ../../../
cd -
```

2.1.2 ls

List directory.

Usage:

2.1.3 pwd

Print working directory.

Usage:

2.1.4 cp

Copy file.

Usage:

2.1.5 mv

Move file.

Usage:

2.1.6 rm

Remove file.

Usage:

2.1.7 cat

Concatenate file to terminal.

Usage:

2.1.8 grep

Globally search a Regular Expression and Print.

Usage:

2.1.9 chmod

Change file Mode.

Usage:

2.1.10 chown

Change file Owner

Usage:

2.1.11 wget

Web get.

Usage:

2.1.12 less

Print a file in a buffer.

Usage:

2.1.13 tail

Print the tail end of a file.

Usage:

2.1.14 w

List active users.

Usage:

2.1.15 sudo

Superuser do.

Usage:

3 Filesystem

The linux/unix filesystem is quite different from the Windows filesystem. There are no ”drives” like C:, but rather there is a root directory called / which is the root of all files and directories in the system, including system files, user home directories, and libraries and programs. The filesystem is accessed by **users** which are part of **groups** that control who has access to what, the *permissions*. Just like users have passwords, groups can have passwords, and groups can have owners as well. We’ll see more about users, groups, and permissions later on.

The general structure of a unix filesystem is as follows.

```
/
├── bin/
├── boot/
├── cdrom/
├── dev/
├── etc/
├── home/
│   └── sam/
├── lib/
├── media/
├── opt/
├── proc/
├── root/
├── run/
├── sbin/
```

```
|
├── srv/
├── sys/
├── tmp/
├── usr/
└── var/
```

This is the root directory, /, and the entire filesystem stems from this point. The user that logs on typically finds him or herself in the /home/username/ directory, where username is the user's username, like 'sam' above. The root, /, is the **mount point** for the file system, and the file system is said to be **mounted** when it is in use. This is because a hard disk could contain many different formats of filesystems - for example a Windows filesystem and a unix filesystem. A given filesystem is simply a section in memory, and *mounting* the filesystem means telling the operating system what type of filesystem it is, where it is, and how to use it.

A brief review of directories listed above and their contents and usage is in order.

/bin/ Essential command binaries

/boot/ Bootloader files and programs

/dev/ Virtual filesystem for devices

/etc/ System-wide configuration files

/home/ All user home directories

/lib/ Static and dynamic library files for linux

/media/ Mount points for USB drives and other media

/mnt/ Temporarily mounted filesystems

/opt/ Optional application packages

/proc/ Virtual filesystem for processes, hardware status, etc

/root/ Home directory for root user

/run/ Run-time variable data

/sbin/ Essential system binaries

/srv/ Site-specific data served by the system

/sys/ Information about devices, drivers, some kernel features

/tmp/ Temporary files

/usr/ Secondary hierarchy for read-only user data

/var/ Variable files, expected to change continuously, such as logs, etc

4 Software

5 Devices and Drivers