KOÇ UNIVERSITY

COMP 350
Selected Topics - Introduction to DevOps

Lecture 2
Basic Shell Commands

Hakan Ayral

Filesystem navigation

- pwd—Print name of current working directory.
- cd—Change directory.
- ls—List directory contents.

Table 2-1: cd Shortcuts

Shortcut	Result
`cd`	Changes the working directory to your home directory.
`cd -`	Changes the working directory to the previous working directory.
`cd ~username`	Changes the working directory to the home directory of *username*. For example, `cd ~bob` changes the directory to the home directory of user *bob*.

Filesystem navigation

- ***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 that directory is */usr/bin*.
    - This means from the root directory (represented by the leading slash in the pathname) there is a directory called *usr* that contains a directory called *bin*.
- ***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 filesystem 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.

Facts about file names

- **Filenames that begin with a period character are hidden**.
  - This only means that ls will not list them unless you say ls -a.
    - Some applications place their configuration and settings files in your home directory as hidden files.
- **Filenames and commands in Linux, as in Unix, are case sensitive.**
  - The filenames *File1* and *file1* refer to different files.
- **Linux has no concept of a "file extension" like some other operating systems**.
  - You may name files any way you like. The contents and/or purpose of a file is determined by other means. Although Unix-like operating systems don't use file extensions to determine the contents/purpose of files, some application programs do.
- Though Linux supports long filenames that may contain embedded spaces and punctuation characters, **try not to use punctuation characters in file names** other than period, dash (hyphen), and underscore.
  - *Most importantly, do not embed spaces in filenames*. Embedding spaces in filenames will make many command line tasks more difficult.

Options and Arguments

- Commands are often followed by one or more ***options*** that modify their behavior and, further, by one or more ***arguments***, the items upon which the command acts.
- Most commands look something like this:

```
command -options arguments
```

- Most commands use options consisting of a single character preceded by a dash, such as -l.
- Many commands also support ***long options***, consisting of a word preceded by two dashes.
- Many commands allow **multiple short options** to be strung together.
- `hayral@computer1:~$ ls -lt` → in this example, the ls command is given two options,
  - the ***l*** option to produce long format output, and
  - the ***t*** option to sort the result by the file's modification time.

Options for *ls* command

Table 3-1: Common ls Options

Option	Long Option	Description
`-a`	`--all`	List all files, even those with names that begin with a period, which are normally not listed (i.e., hidden).
`-d`	`--directory`	Ordinarily, if a directory is specified, `ls` will list the contents of the directory, not the directory itself. Use this option in conjunction with the `-l` option to see details about the directory rather than its contents.
`-F`	`--classify`	This option will append an indicator character to the end of each listed name (for example, a forward slash if the name is a directory).
`-h`	`--human-readable`	In long format listings, display file sizes in human-readable format rather than in bytes.
`-l`		Display results in long format.
`-r`	`--reverse`	Display the results in reverse order. Normally, `ls` displays its results in ascending alphabetical order.
`-S`		Sort results by file size.
`-t`		Sort by modification time.

```
hayral@Computer1:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
hayral@Computer1:~$ ls -l
total 32
drwxr-xr-x 2 hayral hayral 4096 Feb 28 20:57 Desktop
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Documents
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Downloads
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Music
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Pictures
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Public
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Templates
drwxr-xr-x 2 hayral hayral 4096 Feb 21 21:44 Videos
hayral@Computer1:~$ ls -a
.              Downloads     .sudo_as_admin_successful
..             .gnupg        Templates
.bash_history  .gtkrc-2.0    .vboxclient-clipboard.pid
.bash_logout   .gtkrc-xfce   .vboxclient-display-svga-x11.pid
.bashrc        .local        .vboxclient-draganddrop.pid
.cache         .mozilla      .vboxclient-seamless.pid
.cinnamon      Music         Videos
.config        Pictures      .Xauthority
Desktop        .profile      .xsession-errors
Documents      Public        .xsession-errors.old
hayral@Computer1:~$
```

Determining file type with *file*

- use the ***file*** command to determine a file's type.
- filenames in Linux do not reflect a file's contents.
- You can use the file command as:

`file` *`filename`*

- When invoked, the file command will print a brief description of the file's contents.

```
hayral@Computer1:~$
hayral@Computer1:~$ cd Desktop/
hayral@Computer1:~/Desktop$ ls *.jpg
index.jpg
hayral@Computer1:~/Desktop$ file index.jpg
index.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segme
nt length 16, baseline, precision 8, 225x224, components 3
hayral@Computer1:~/Desktop$
```

Inspect file contents with *less*

- ***less*** is a command to view text files.
  - many files that contain human-readable text.
  - ***less*** provides a convenient way to examine them.
- Why would we want to examine text files?
  - Because many of the files that contain system settings (called *configuration files*) are stored in this format, being able to read them gives us insight about how the system works.
  - In addition, many of the actual programs that the system uses (called *scripts*) are stored in this format. In later chapters, we will learn how to edit text files in order to modify system settings and write our own scripts, but for now we will just look at their contents.

Table 3-3: less Commands

Command	Action
PAGE UP or b	Scroll back one page.
PAGE DOWN or Spacebar	Scroll forward one page.
Up Arrow	Scroll up one line.
Down Arrow	Scroll down one line.
G	Move to the end of the text file.
1G or g	Move to the beginning of the text file.
/*characters*	Search forward to the next occurrence of *characters*.
n	Search for the next occurrence of the previous search.
h	Display help screen.
q	Quit less.

LESS IS MORE

The less program was designed as an improved replacement of an earlier Unix program called more. Its name is a play on the phrase "less is more"—a motto of modernist architects and designers.

less falls into the class of programs called *pagers*, programs that allow the easy viewing of long text documents in a page-by-page manner. Whereas the more program could only page forward, the less program allows paging both forward and backward and has many other features as well.

Common Directories

- On the right you can find the list of common directories on a linux installation along with their descriptions.

Directory	Comments
/	The root directory, where everything begins.
/bin	Contains binaries (programs) that must be present for the system to boot and run.
/boot	Contains the Linux kernel, initial RAM disk image (for drivers needed at boot time), and the boot loader. Interesting files: • */boot/grub/grub.conf* or *menu.lst*, which are used to configure the boot loader • */boot/vmlinuz*, the Linux kernel
/dev	This is a special directory that contains *device nodes*. "Everything is a file" also applies to devices. Here is where the kernel maintains a list of all the devices it understands.
/etc	The */etc* directory contains all of the system-wide configuration files. It also contains a collection of shell scripts that start each of the system services at boot time. Everything in this directory should be readable text. Interesting files: While everything in */etc* is interesting, here are some of my all-time favorites: • */etc/crontab*, a file that defines when automated jobs will run • */etc/fstab*, a table of storage devices and their associated mount points • */etc/passwd*, a list of the user accounts
/home	In normal configurations, each user is given a directory in */home*. Ordinary users can write files only in their home directories. This limitation protects the system from errant user activity.
/lib	Contains shared library files used by the core system programs. These are similar to DLLs in Windows.
/lost+found	Each formatted partition or device using a Linux file-system, such as ext3, will have this directory. It is used in the case of a partial recovery from a filesystem corruption event. Unless something really bad has happened to your system, this directory will remain empty.
/media	On modern Linux systems the */media* directory will contain the mount points for removable media such as USB drives, CD-ROMs, etc. that are mounted automatically at insertion.
/mnt	On older Linux systems, the */mnt* directory contains mount points for removable devices that have been mounted manually.
/opt	The */opt* directory is used to install "optional" software. This is mainly used to hold commercial software products that may be installed on your system.

Directory	Comments
/proc	The */proc* directory is special. It's not a real filesystem in the sense of files stored on your hard drive. Rather, it is a virtual filesystem maintained by the Linux kernel. The "files" it contains are peepholes into the kernel itself. The files are readable and will give you a picture of how the kernel sees your computer.
/root	This is the home directory for the root account.
/sbin	This directory contains "system" binaries. These are programs that perform vital system tasks that are generally reserved for the superuser.
/tmp	The */tmp* directory is intended for storage of temporary, transient files created by various programs. Some configurations cause this directory to be emptied each time the system is rebooted.
/usr	The */usr* directory tree is likely the largest one on a Linux system. It contains all the programs and support files used by regular users.
/usr/bin	*/usr/bin* contains the executable programs installed by your Linux distribution. It is not uncommon for this directory to hold thousands of programs.
/usr/lib	The shared libraries for the programs in */usr/bin*.
/usr/local	The */usr/local* tree is where programs that are not included with your distribution but are intended for system-wide use are installed. Programs compiled from source code are normally installed in */usr/local/bin*. On a newly installed Linux system, this tree exists, but it will be empty until the system administrator puts something in it.
/usr/sbin	Contains more system administration programs.
/usr/share	*/usr/share* contains all the shared data used by programs in */usr/bin*. This includes things like default configuration files, icons, screen backgrounds, sound files, etc.
/usr/share/doc	Most packages installed on the system will include some kind of documentation. In */usr/share/doc*, we will find documentation files organized by package.
/var	With the exception of */tmp* and */home*, the directories we have looked at so far remain relatively static; that is, their contents don't change. The */var* directory tree is where data that is likely to change is stored. Various databases, spool files, user mail, etc. are located here.
/var/log	*/var/log* contains *log files*, records of various system activity. These are very important and should be monitored from time to time. The most useful one is */var/log/messages*. Note that for security reasons on some systems, you must be the superuser to view log files.