Computer Architecture

Lab 3 (Command Line Basics)

Overview

This lesson covers the most essential commands every user should know. These basic commands cover topics like getting help, navigating directories, and listing files. Before you begin, there are three important rules you need to know about the command line:

1. Unix, Linux, and BSD systems are case (and space) sensitive. This means that a file named MyFile is not same as myfile as it would be on a DOS or Windows system.
2. There is no "recycle bin" or "trash can" when working in the command line environment. When files are deleted on the command line, they instantly disappear forever.
3. You should always practice new commands on a testing system that is not used in a production environment. This minimizes the chances of an accident that can take down an important system.

Glossary of terms used in this lesson:

Argument	One or more variable input items used by a command line program.
Binary	A compiled program or data file.
Flag	Synonym for *Option.*
Man Page	Online manual for shell commands.
Option	A modifier that alters the default operation of a command.
Parameter	Synonym for *Argument.*
Recursively	Includes all subdirectories when executing a command.
Shell Script	A grouping of commands in a plain text executable file.
Source Code	Uncompiled code for a program.
Switch	Synonym for *Option.*
Working Directory	Your current location within the directory tree.

Commands, Options, and Arguments

A Linux command normally consists of three parts: the command itself, the command options, and its arguments. For instance, the following example shows what a Linux command looks like:

```
ls -l /etc/hosts
```

This example consists of three parts, ls, which is the command; -l, which is the option; and /etc/hosts, which is the argument. The command itself is the character string you type to activate a certain task. Most commands have options as their second part. By using these options, you modify the behavior of the commands. Apart from options, many Linux commands have arguments. These are additional specifications that you can add to the command to tell it more precisely what to do.

Commands covered in this lesson:

Command	Purpose
`man`	Online manual for command line programs.
`whatis`	Display a description of the specified command.
`ls`	List the contents of a directory.
`pwd`	Display the current/working directory.
`cd`	Change (navigate) directories.
`tree`	Display the contents of a directory in a tree hierarchy format.
`find`	Search for files and directories.
`locate`	Search the locate database for files and directories.
`whereis`	Display the location of binary files, manual pages, and source code for the specified command.
`file`	Display the file type of the specified file.
`stat`	Display extended information about a file system, file, or directory.
`date`	Display or set the system clock.
`cal`	Display a calendar on the command line.
`history`	Display commands that have recently been executed.
`clear`	Clear the contents of the current screen.
`logout`	Logout of the system.
`exit`	Exit the current shell.

man

Purpose: Online manual for command line programs.

Usage syntax: man [OPTIONS] [COMMAND/FILE]

The man command displays the manual (often referred to as the man page) for the specified command. Each manual page provides detailed information about a command's options and usage. The man command is your most valuable resource for help on the command line. Always look to the manual pages for any command if you are unsure of its proper usage syntax.

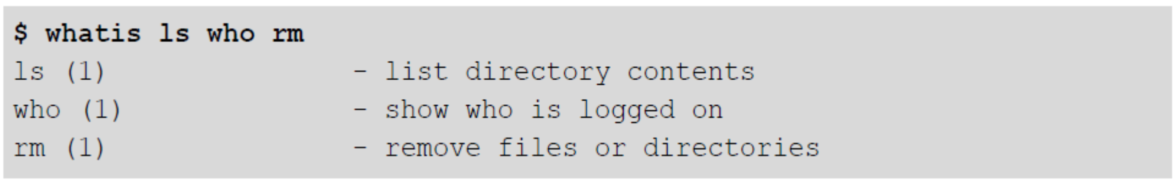
whatis

Purpose: Display a description of the specified command.

Usage syntax: whatis [OPTIONS] [COMMAND]

whatis displays a brief description of the specified command. This can be used as a helpful reminder of a command's purpose without having to refer to the man command.

Multiple commands can be used with one whatis query to display the description of each individual command. For example, typing whatis ls who rm would display the description of all three commands at once as demonstrated in the next example.



```
$ whatis ls who rm
ls (1)               - list directory contents
who (1)              - show who is logged on
rm (1)               - remove files or directories
```

Viewing the manual description of multiple commands

ls

Purpose: List the contents of a directory.

Usage syntax: ls [OPTIONS] [DIRECTORY/FILE]

Executing the ls command displays a simple list of files in the current directory. To see more information about the files in a directory you can use command line options to activate additional features, as demonstrated in the next example.

```
$ ls -l
-rw-r--r-- 1 nick sales 35068 2009-05-19 08:41 Notes.txt
-rw-r--r-- 1 nick sales    23 2009-05-19 08:43 ShoppingList.txt
-rw-r--r-- 1 nick sales    37 2009-05-19 08:43 ToDoList.txt
```

Using the -l option with the ls command

In this example, the -l option is used to produce a detailed list of files including the permissions, owner, group, and modification date/time. The table below describes the output of the ls -l command.

Permissions	Number of Links	Owner & Group	Size	Modification Date	File or Directory
-rw-r--r--	1	nick sales	35068	2009-05-19 08:41	Notes.txt

Description of fields displayed with the ls -l command

Most command line programs have numerous options available. The ls command is no exception. By combining these options you can activate multiple features at the same time.

Common usage examples:

`ls`	Display a basic list of files in the current directory
`ls [DIRECTORY]`	Display a basic list of files in the specified directory
`ls -l`	List files with details
`ls -la`	List hidden files
`ls -lh`	List file sizes in "human readable format" (KB, MB, etc.)
`ls -R`	Recursively list all subdirectories
`ls -d [DIRECTORY]`	List only the specified directory (not its contents)

pwd

Purpose: Display the current/working directory.

Usage syntax: pwd

```
$ pwd
/home/nick
```

Using the pwd command the display the current directory

The pwd command (short for Print Working Directory) displays your current location within the file system. In the above example, executing pwd displays /home/nick as the current working directory.

cd

Purpose: Change (navigate) directories.

Usage syntax: cd [DIRECTORY]

```
$ cd /etc
$ pwd
/etc
```

Using the cd command to navigate to the /etc directory

The cd command (short for Change Directories) changes your location within the file system to the specified path. In the above example, executing cd /etc makes /etc the new working directory.

The cd command interprets directory paths relative to your current location unless you manually specify a full path (such as cd /etc as used in the first example.) The next example demonstrates using cd to change directories relative to the current location.

```
$ pwd
/home/nick
$ cd documents
$ pwd
/home/nick/documents
```

Using the cd command to navigate to a directory relative to the current location

In this example, the starting directory is /home/nick. Typing cd documents makes /home/nick/documents the new working directory. If you were starting in a different location you would have to type the full path (i.e. cd /home/nick/documents) to achieve the same results. Since the previous location was /home/nick, typing the full path is not necessary.

Tip	*Executing the* `cd` *command with no options returns you to your home directory regardless of your current location.*

Common usage examples:

`cd [DIRECTORY]`	Navigate to the specified directory
`cd`	Navigate to the user's home directory
`cd -`	Go back to the previous working directory
`cd ..`	Navigate up one level in the directory tree

tree

Purpose: Display the contents of a directory in a tree hierarchy format.

Usage syntax: tree [OPTIONS] [DIRECTORY]

The tree command displays a directory listing in tree form. This is useful for visualizing the layout of a directory structure. In the above example, executing tree -d -L 2 displays 2 directory levels (relative to the current location) in tree form.

find

Purpose: Search for files and directories.

Usage syntax: find [PATH] [OPTIONS] [CRITERIA]

```
# find / -name hosts
/etc/avahi/hosts
/etc/hosts
/usr/share/hosts
```

Using the find command to locate files with the word "hosts" in their name

The find command performs a raw search on a file system to locate the specified items. You can search for files using a number of characteristics - the most common being file name, owner, size, or modification time. The above example displays the results of a search for files that contain the word "hosts" in their file name.

Common usage examples:

`find [PATH] -name [NAME]`	Find files with the specified name
`find [PATH] -user [USERNAME]`	Find files owned by the specified user
`find [PATH] -size [FILESIZE]`	Find files larger than the specified size
`find [PATH] -mtime 0`	Find files modified in the last 24 hours

locate

Purpose: Search the locate database for files and directories.

Usage syntax: locate [OPTIONS] [DIRECTORY/FILE]

```
$ locate hosts
/etc/avahi/hosts
/etc/hosts
/usr/share/hosts
```

Searching the locate database for files that contain the word "hosts" in their name

The locate command displays the location of files that match the specified name. While similar to the find command, locate is significantly faster because it searches a database of indexed filenames rather than performing a raw search of the entire file system. A disadvantage of the locate command is the fact that it lacks the ability to search for advanced characteristics such as file owner, size, and modification time.

whereis

Purpose: Display the location of binary files, manual pages, and source code for the specified command.

Usage syntax: whereis [OPTIONS] [COMMAND/FILE]

```
$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
```

Displaying the file locations of the ls program using whereis

whereis displays the file locations for the specified command. In the above example, whereis displays the binary file and manual page location for the ls command.

file

Purpose: Display the file type of the specified file.

Usage syntax: file [OPTIONS] [FILE]

The file command displays information about the contents of the specified file. Microsoft Windows systems often use a file extension (such as .txt, .exe, .zip, etc.) to identify the type of data found in a file.

Unix, Linux, and BSD files rarely include an extension which can make identifying their file type a challenge. The file command is provided to resolve this problem.

```
$ file /bin/bash
/bin/bash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
$ file /etc/hosts
/etc/hosts: ASCII English text
$ file /home/nick/backup.tgz
backup.tgz: gzip compressed data, from Unix, last modified: Tue May 19
22:29:29 2009
$ file /dev/cdrom
/dev/cdrom: symbolic link to 'sr0'
$ file /dev/sr0
/dev/sr0: block special
```

Using the file command to identify several different types of files

The above example displays results for several file types commonly found on Unix, Linux, and BSD systems. The table below displays more information about these file types.

Type	Description
Ascii Text Files	Plain text files
Binary Files	Executable programs such as those located in the `/bin` and `/usr/bin` directories
Compressed Files	Files compressed through the **`compress`** or **`gzip`** programs
Device Files	Special virtual files that represent devices
Links	Links (AKA shortcuts) that point to other files or directories

Basic file types found on Unix, Linux, and BSD systems

stat

Purpose: Display extended information about a file system, file, or directory.

Usage syntax: stat [OPTIONS] [FILE/DIRECTORY]

```
$ stat /etc/hosts
  File: '/etc/hosts'
  Size: 266            Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d     Inode: 788         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/   root)   Gid: (    0/    root)
Access: 2009-05-25 20:47:14.916626707 -0500
Modify: 2009-05-25 20:46:57.512623325 -0500
Change: 2009-05-25 20:46:57.512623325 -0500
```

Displaying information for the /etc/hosts file using the stat command

The stat command displays extended information about files. It includes helpful information not available when using the ls command such as the file's last access time and technical information about the file's location within the file system. The example above displays the stat output for the /etc/hosts file. The next example displays the stat output for the /etc directory itself.

date

Purpose: Display or set the system clock.

Usage syntax: date [OPTIONS] [TIME/DATE]

```
$ date
Wed Jun 10 20:33:27 CDT 2009
```

Output of the date command

The -s option can be used to set the time/date on the system as demonstrated in the next example.

```
# date -s "07/10/2009 11:30"
Fri Jul 10 11:30:00 CDT 2009
```

Setting the time and date

When setting both the time and date you must use "MM/DD/YYYY HH:MM" format. To set the time only you can simply use date -s HH:MM.

cal

Purpose: Display a calendar on the command line.

Usage syntax: cal [OPTIONS] [MONTH] [YEAR]

```
$ cal
      May 2009
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

Displaying a calendar for the current month

The cal command displays a simple calendar on the command line. Executing cal with no arguments will display a calendar for the current month, as shown in the above example.

history

Purpose: Display commands that have recently been executed.

Usage syntax: history [OPTIONS]

```
$ history 10
  686  man uptime
  687  cat /etc/hosts
  688  ls -l
  689  uptime
  690  dmesg
  691  iostat
  692  vmstat
  693  ping google.com
  694  tracepath google.com
  695  history 10
```

Display 10 lines of command history

The history command displays a user's command line history. Executing the history command with no arguments will display the entire command line history for the current user. For a shorter list, a number can be specified as an argument. Typing history 10, for example, will display the last 10 commands executed by the current user as shown in the above example.

clear

Purpose: Clear the contents of the current screen.

Usage syntax: clear

The clear command clears the contents of the terminal screen. This is useful for uncluttering the display after you have executed several commands and are preparing to move to the next task.

logout

Purpose: Log out of the system.

Usage syntax: logout

```
$ logout

Ubuntu 9.04

login:
```

Results of the logout command

The logout command logs your account out of the system. This will end your terminal session and return to the login screen.

exit

Purpose: Exit the current shell.

Usage syntax: exit [CODE]

The exit command is similar to the logout command with the exception that it does not run the logout script located in the user's home directory. The above example shows the results of exiting the shell and returning to the login prompt.