# Shell, Terminals and Syntax

## Assignment Objectives:

On completing this assignment, you should be able to:

* Identify the difference between the terminal and shell
* Identify correct syntax for Linux commands.

# Part 1

## Instructions:

Follow the instructions below. When indicated answer the questions or provide the necessary screenshots.

## Task

**Task 1**

Sign in to [vShpere](https://infoadc.mccinfo.net/) and open your virtual machine. (If you are not sure how to do this, review the handout found in Canvas in the Virtual Lab folder.) Login to the **sysadmin** account using the password netlab123.

**Task 2**

Open the terminal app. You will do this the same way you did in previous assignments.

**Task 3**

You are logged in as a normal user.

You will see this prompt: sysadmin@CentOS8:~$

The prompt shows you (1) the user who is logged into the current session, (2) the name of the computer logged into followed by a colon, (3) where the user is placed in the filesystem – the tilde (~) is a shortcut to the user’s home directory, and (4) the symbol for the prompt ($).

**Task 4**

Now switch user (su) to the superuser, root, by typing the command, su - root . The password is Passw0rd (that is a zero, not the letter “o”)

You will see this prompt: [root@CentOS8:~]#

What is different about the prompt for superuser or root?

**Your answer below:**

**Task 5**

Close the root user’s session and go back to being sysadmin by typing: “exit”. Notice the prompt change.

**Task 6**

Whenever you su to another user it is recommended to use the dash because it switches you to that user’s environment. Otherwise, you become the new user, but are still in the previous user’s environment. If you need to become root, you need root’s environment variables, profile, permissions, etc.

**Task 7**

There is a better way to run root-level commands than by switching to root, by using the sudo command (switch user and do). For example, attempt to install a program called **curl** by typing: sudo **yum install -y curl** , then press enter. You should get a permission denied error. You are trying to run a command as a normal user and not allowed to run this command. Now type the same command with **sudo** at the beginning, ***sudo yum install -y curl*** . You will then be prompted for the sysadmin’s password which is netlab123.

With sudo, sysadmin has root privileges only to do the **yum install**. It works because sysadmin is in the sudo group, the list of users allowed to use sudo. Will Adams, the user you logged in as in previous assignments was not part of the sudo group and could not use sudo.

Type the command, id sysadmin . Notice the groups of which sysadmin is a member. Now type the command, id wadams. Notice how wadams has less access and is not part of the sudo group. However sysadmin is part of the sudo group.

NOTE: YUM is a package manager. In this case, we used “yum install” to get the finger package from the Internet and to install the finger program on our computer.

**Task 8**

A shell is a program that provides a command-line interface. It interfaces between you and the operating system kernel. It accepts input from the keyboard - it is waiting in the background, listening for your keystrokes. It acts upon the commands you enter, interpreting them and sending instructions to the kernel. It is your command interpreter.

A shell is started when terminal window is opened. The function of a shell is to make it easy for users to work with files and to make it possible to pass parameters to a program being started. The way the shell work is (it’s behavior), if a command succeeds, no response from shell; if a command fails, error message given.

For example, type some gibberish in the command line, such as **asdfkdjf** . You should receive an error. The shell should respond back to you letting you know that our gibberish was not a known command. However, if we type an actual command it will execute the command without an error.

The shell also allows a user to create scripts, which is a program consisting of a list of commands stored in a text file. Shell scripting is a topic all its own and covered in a later module.

**Task 9**

There are several shells (command lines) available. The most commonly used shell in Linux is The Bourne Again shell (also known as BASH). BASH we are using for this course.

When you first boot Linux and sign in, your default shell is loaded. You can identify your default shell by typing: **echo $SHELL** . This will give you the absolute path or (system location) of the shell you are using.

What is your default shell’s Absolute path?

**Your answer below:**

**Task 10**

BASH has several features such as, a) Sophisticated prompt control, b.)Command line completion, c) Command line history, d) Command line editing - the vi editor is standard on Linux distributions. BASH has several built in commands such as: ls, cd, cp, cat, date, mkdir, rm, mv.

BASH also has built in instruction manuals for its commands. You access the manuals by using the man command followed by the command you want to see the manual for. For example if you type, man ls, will open the manual for the ls command. It will open the manual in a text viewer (similar to a ebook reader program). To navigate you used keys on the keyboard. To move down by line use the down arrow. To move up, use the up arrow key. **Press “q” to quit** and return to the command line.

Other manuals you might want to look at are, man bash, man cp, man cat, etc.

**Task 11**

Syntax are the rules and structure of commands. Here is the basic command-line syntax (called SYNOPSIS in the man pages):

command [-options] [option arguments] [command arguments]

NOTE: Everything is square brackets is optional [ ], which means for this command you don’t have to use any options or arguments.

Example: ls -la /home/sysadmin

a. In this example, the command is ls

b. the [-options] are l (for **long** list) and a (for list **all** files, including hidden files). Optional because it is in [ ].

c. [command argument] is /home/sysadmin. Optional because it is in [ ].

This command will list all files in the /home/sysadmin directory. It will list them in the long format (the -l option) and list all files including hidden files (the a option).

**Type the command, man ls, to view the manual for ls.**

Notice the SYNOPSIS which describe the syntax of the command and the DESCRIPTION which gives an alphabetical list of all the many options available to the ls command. Notice the -a option. Files that start with . (dot) are hidden.

What other options are available for the ls command? (Remember “q” to quit)

**Your answer below:**

**Task 12**

Open the manual for the lpr command to answer the following questions. What does the lpr command do? When you add the -P option to the lpr command, what does it do?

**Your answer below:**

**Task 13**

Linux is case-sensitive and only accepts command names in lowercase letters (with very few exceptions). Commands are divided into two categories: user-level commands stored in /bin and system-level commands stored in /sbin or /usr/sbin. The “s” stands for system and “bin” stands for binaries (executable programs), thus /sbin holds administrative tools, executable programs, which are not usually run by normal users.

At the shell prompt, type the **clear** command with all lower case letters and press <enter>. Now type the command **Clear** with a capital C.

What error did you recieve when you typed Clear with a capital? Explain why you received this error.

**Your answer below:**

**Task 14**

Command options modify the command’s behavior. When they are included, they are usually preceded by “-” (called a dash, not hyphen or minus sign). Most options are designated by a single lowercase letter.

Example: ls -l

-l (dash L) is the option for the ls command. It changes the ls command to list files in the long format.

More than one option can be specified after the single dash.

Example: ls -la

-la is a combination of the -l and -a option (shows all hidden files). This command will list files in long format and all hidden files. The longer way to run this command is ls -l -a, which does the same thing.

But, some options are a single uppercase letter and some are complete words. If the option is a complete word, it is preceded by a double dash (--)., but not after a double dash.

Example of command that use complete words as options: uname --help or ls --help

Some options have both a single-character and a word, for instance: ls -a and ls --all do the same thing.

Some commands break the normal syntax rules, such as the ps command: ps aT doesn’t need a dash, has an uppercase option.

So if the there are so many syntax rules for commands, how can you know which option to use, or use a dash or no dash or use two dashes, etc. **LOOK at the manual (man ls)**. Sometimes on a forum when you ask a question, someone might tell you to RTFM which means affectionately to “Read the F\*\*\*\*\*\* Manual”. That’s what the manpages (manuals) are for.

What should you do if you need to know the syntax of a command or know which option to use for a command?

**Your answer below:**

**Task 15**

Commands often need to operate on something, and that something is called the **argument**. Arguments are generally filenames (paths), for example, cp command needs two arguments, what does it need to copy and where does it need to be copied to. For example, cp file.txt practice The cp command is going to copy the file.txt to the practice directory. If you did include file.txt, how would it know what to copy. In addition if it didn’t include practice and would it know where to copy the file.txt file to. It would make any sense to just use cp by itself.

The argument for the following command, **ls -la /etc/hosts**, is /etc/hosts.ls is the command, -la are the options and /etc/hosts is the argument.

What is the argument for the following made up command, foo -xy file1? What is the option? What is the command?

**Your answer below:**

**Task 16**

One of the cool things about the command line shell is being able to use metacharacters to combine programs or manipulate programs to do other things. For instance, type the command, clear, then enter (it clears the screen). Then type the command, date, then enter (it gives the date and time). Then type the command whoami, then enter (displays your user name).

We will now use the metacharacter ; (semicolon) to combined all of the command we just typed. They the following,

**clear; date; whoami**

Type these as-is with the semicolons. All 3 command should run.

Combine the commands, clear, pwd and cal, then paste a screenshot of the output.

**Your screenshot below:**

**Task 17**

Other metacharacters can be used and are listed below. We will discuss these metacharaters later.

| Metacharacter | Description |
| --- | --- |
| $ | Shell variable content |
| ~ | Special home directory variable |
| ; | Command termination |
| < << > >> | Input/Output redirection |
| | | Command piping |
| $( ) | Command substitution |
| & | Command background execution |
| \* ? [ ] | Shell wildcards |
| " ' \ | Metacharacter quotes, escape key |
| { } | Brace Expansion |
| ( ) { } | Command grouping |

**Task 18**

Describe what you like or dislike about this lab. Did you find this easy or challenging? Explain why.

**Your answer below:**

**Task 19**

After going through this lab, do you feel you better understand command line syntax and how to use manpages (manuals) on the command line? Explain why or why not.

**Your answer below:**

## When Completed

1. Save this document with your answers.
2. Submit this completed document in the assignment Dropbox.