# Linux Lab 1 – Basics

There is a bit of a learning curve when you get started with any command line interface (CLI). This assignment has more reading than normal. Hang in there and do it because it is important. Much of this course requires that you be comfortable with the command line.

# Hand In

When I first created this lab, my intent was that you would paste the results from your Linux terminal into your document with answers, for every step. That proved to be tedious, so just hand in the answers to steps 20 and 21. Be sure to do all the steps at a real Linux terminal; otherwise, you won’t learn the material and will have problems with later lessons.

# Reading

Listen to, or read, CyberAces Module 1 Linux, Session 3 Core Commands (<https://tutorials.cyberaces.org/tutorials/view/1-1-3.html>)   
Do the exercises in the CyberAces module!

Read Chapters 1 - 4, and 6 in *The Linux Command Line*. Books will be handed out in class, or you can access a digital version on the class Canvas site in Unit 0, LinuxCommandLine.zip. Or, you can download a copy of the book at <https://sourceforge.net/projects/linuxcommand/files/TLCL/19.01/TLCL-19.01.pdf/download>.

Pay attention to the information about these commands, as you will use them in the lab:  
cd  
ls  
pwd  
mkdir  
echo  
cp  
rm  
rmdir

The following commands are covered later in the book, so here is a brief description of what they do:  
touch [file] This command updates the access times for a file. If the file does not exist, the command creates a new, empty file. The command is often used as an easy way to create an empty file.

ps This command lists the processes (apps or the like) that are running on the computer.

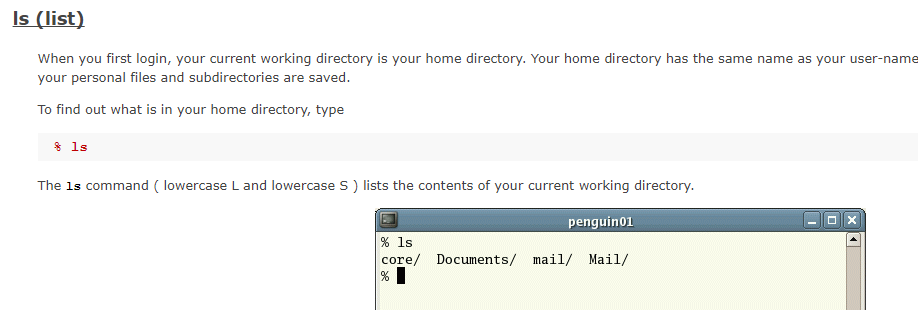
netstat This command lists the network connections the computer has. Note that it will list many STREAM connections, which are used to communicate between processes on the computer. Usually we filter them out. Netstat has been deprecated and may not be present on your VM. The replacement for netstat is ss.

ss This is the replacement for netstat. The syntax is like netstat as well.

grep This is a powerful command for searching for strings in files and output.

Take special note of the gray boxes on pages 11, 17, 28, and 31. They have helpful information.

# Extra Help

If you are nervous about using the Linux command line you can go to <http://www.ee.surrey.ac.uk/Teaching/Unix/> and do Tutorials One and Two. Note: You should have your Linux VM running, with an open terminal window. Whenever the text on the web site is in red, you are supposed to do that yourself on your own VM. You don’t need to hand in anything for this, but it may make the lab easier. If you already know the Linux commands and are bored, you can skip this.

# Basic File Commands

1) Open a terminal

2) Change to the root (/) directory.

3) Do an ls command while you are in the root of the file system tree to see what's there. Note that there is a directory at the root (/) called "root". This is the home directory for the root user, not the root of the tree.

4) Change directory to home, then do "ls" to see what is there. You probably have only your user home directory, named the same as your username. Change directory into your home directory, the one named after your user.

5) Use pwd (print working directory) to see the path to the directory you are in. Hopefully, it is /home/<your username>.

6) In your home directory, do ls, then ls -l, and compare them. What extra information do you see?

7) In your home directory, do ls -la. This shows a lot of hidden directories that contain configuration information for your profile and programs. (Forensics information about what the user has done can be gleaned from these directories.) How are the hidden directories/files named differently from those that are not hidden?

8) Change directory to your desktop. Note that cd desktop does not work, but cd Desktop does. Linux is case sensitive, Windows is not.

9) Create two new directories on your desktop using mkdir. You should see the new directories appear on your desktop.

10) Create two new files inside the first directory you made. Create the first file using the touch command and the second by using the echo command and redirection.

11) Copy the files to the second directory.

12) Delete the first directory. Remember that you will have to empty it first.

13) Open Firefox and download an image (picture). By default, it will be put in your /home/<username>/Downloads directory.

14) Change directory to ~/Downloads. What does that do, and what does the ~ do?

15) List the contents of your Downloads directory. Hopefully, your picture is there.

16) Run the command, file [name of your picture]. Does the information seem correct?

17) Get a list of processes running on your machine by using ps aux. To see just one page at a time, pipe the output from ps aux into more, using this: ps aux | more. Try it again using ps aux | less. (less is an improved version of more.)

18) Our terminal is a bash shell, so we should have a bash process in there somewhere. You can filter it from the noise by piping the output from ps aux into the great Linux search program grep. Try   
ps aux | grep bash.

19) Note: The ss application has replaced netstat, although you will see netstat referenced in many books and blogs. If you want to practice with netstat, you will have to install it by typing:  
sudo apt install net-tools in Ubuntu  
sudo yum install net-tools in CentOS  
(this is optional, ss works fine.)

Use ss -l (-l is a lower-case L, for listening) to get a list of listening network ports. Unix uses network sockets or streams for internal communications so this output will be very noisy. You can filter out the internal connection noise by just looking for TCP and UDP connections, which are connections with other computers.

Use ss –help to see the options that ss allows. What would the command ss -nat do? Once you have made a guess, execute ss -nat to see what it does.

You should find that ss -nat lists all connections (-a) but limits the display to TCP connections (-t). The -n causes ss to show port numbers as numbers and not by names (53 vs. dns, for example.)

Now execute an ss command that will show all UDP ports that are open.

Note: We will discuss TCP, UDP, and port numbers in great detail when we reach the networking modules.

20) Execute the following commands:

[john@localhost ~]$ ls

centfile.txt file other pwgen~ Templates

Desktop JohnPublicKey Pictures pwlist Unsaved Document 1~

Documents kalifile.txt Public ssltest.py Videos

Downloads Music pwgen stuff-file

[john@localhost ~]$ mkdir testdir

[john@localhost ~]$ cd testdir/

[john@localhost testdir]$ touch empty file

[john@localhost testdir]$ ls -l empty

-rw-rw-r--. 1 john john 0 Aug 25 13:30 empty

[john@localhost testdir]$ ls -l file

-rw-rw-r--. 1 john john 0 Aug 25 13:30 file

[john@localhost testdir]$ cd ..

[john@localhost ~]$ rmdir testdir

rmdir: failed to remove `testdir': Directory not empty

[john@localhost ~]$

What did the command, touch empty file, do?

Why did rmdir fail? What would you do to correct it?

Come up with two ways to delete the testdir in one line. (Hint: One way I can think of involves a ";". Another involves rm. Any help methods, including Google, are fair game.) Test your answers to be sure they work.

21) Why did this fail?  
[john@localhost ~]$ echo 'lots of stuff here' > Myfile  
[john@localhost ~]$   
[john@localhost ~]$   
[john@localhost ~]$ cat MyFile  
cat: MyFile: No such file or directory  
[john@localhost ~]$

# Hand In

Hand in the answers to steps 20 and 21. Be sure to do all the steps at a real Linux terminal; otherwise, you won’t learn the material and will have problems with later lessons.