

# OPS105 Lab 01 - Exploring A Linux File System

Complete this lab by the beginning of the next class.

## Objectives

To use BASH shell commands in Linux as a regular user to:

- determine system characteristics - `uname`
- list the contents of a directory - `ls`
- display the contents of a file – `more`, `less` and `cat`
- determine the type of a file - `file`
- list the present working directory - `pwd`
- create/delete a directory - `mkdir`, `rmdir`
- move from one directory to another - `cd`
- copy files - `cp`
- delete a file - `rm`

## Required materials

- Access to a lab PC (Matrix)
- USB flash memory stick

# Lab Instructions

1. We want to work from the command line in this lab. To get to the Matrix command line, boot the lab PC to Matrix. Once logged in, you will need to find a **terminal** program and start it. The name will vary, depending on what's available on the host.

Once a terminal is started, notice the command prompt. This prompt indicates that the **shell** program (running in the terminal) is ready to take your command.

2. At the command prompt type in the **uname -a** command. **uname** is a command to display information about your machine, and **-a** is an option for this command which displays "all" the information the command has gathered about the machine. Try the **uname** command without the option.
3. The **man** command shows you the manual page for a selected command, for example **man uname** will show the manual for the **uname** command you used earlier.

Use the **man** command to find out what the **uname -a** command displays, and fill in the table below: (Note: depending on the system in question, not all information may be available)

Kernel version	
Machine hardware name	
Processor type	
Operating system	

4. You should recall from the lecture that the Unix/Linux file system resembles a tree-like structure. You can use path names as arguments for some commands, to apply the specific command to a directory, file or a group of files and/or directories.

At the command prompt, type the command: **less /etc/passwd**.

You'll see a number of lines that look similar to the following:

**mailman:x:72:67:GNU mailing list manager:/var/lib/mailman:/bin/bash**

Each line in the **/etc/passwd** file is a record containing the login account information for a user, with record consisting of seven fields separated by a colon (:). The 6th field stores the **home directory** for the user named in the first field. When you login to a Linux/Unix system remotely, or open up a terminal on a lab PC, the shell program listed in the 7th field of the user account record starts up and sets its current working directory to the user's home directory.

Note that some user accounts have their shell and/or home directory set to a non-existent item. Also, the **passwd** file does not actually store passwords (despite its name) – this information is stored somewhere else encrypted and regular users do not have direct access to it.

6. Type the command **pwd** and find the shell's present working directory (or current working directory). If you haven't changed directories yet, it should be the full path name of your home directory as shown in the **/etc/passwd** file. The shell is always "in" a specific directory, and you can move around a Linux files system by changing to different directories.

Type the command **cd /** and press return. You should not see any output.

Type **pwd** and press return again. What is the current directory now?

Type **cd**, press enter, then type **pwd** and press enter. Where are you now?

7. To list files and directories in the present working directory, type the command **ls** and make note of the output. The command **ls** lists only the name of the files in the current directory. To list the name of the files in another directory, supply the directory as an argument to the **ls** command.

Try listing the following directories: **/etc**, **/tmp**, **/root**.

The **ls** command can also be used to list other file attributes besides the name of a file. It can also list the owner of the file, which group owns the file, the size of the file (expressed in number of bytes), last modification date/time, and access permissions for the file.

Try the command **ls -l** and make note of the output.

The **-l** in the command **ls -l** is called an *option*. The **ls** command supports many options. You can get a list of options by using the man page for the **ls** command.

Use the manual to find the answers for the following questions:

Who is/are the author(s) of the <b>ls</b> program?	
What license is used by this program?	
What option tells <b>ls</b> to show file size in human-readable format?	
What does the option <b>-F</b> do?	
What option tells <b>ls</b> to list all files?	

8. There are several shell commands which can be used to display the contents of a text file. We will study three of them: they are cat,less, and more.

Before we try to use any of these commands to display a file, we must first make sure that the file we want to display contains only regular text. There is another shell command called file, which is used to test and determine the type of data stored in a file.

Use the **file** command to determine the type of the following files:

File Name	File Type
/etc/passwd	
/dev/lp0	
/dev/sda	
/bin/ls	

Once you know which of the files are plain-text, experiment with the cat,more and less commands on them. Try using the commands with the /bin/ls file.

9. So far, we only worked with absolute path names (always starting with the root directory). While this is straightforward, sometimes it may be easier to use a relative path name. Your goal is to gain enough experience that you can make a choice in a specific situation, to accomplish a given task with the least amount of typing.

The shell offers special symbols, assisting us with defining path names. They are “.” (present working directory) and “..” (parent directory). You may need to use these symbols at the beginning of relative path names.

There is also the “~” symbol, sometimes referred to as “relative-to-home path”.

Whenever used with a command, it represents the absolute path to your home directory, for example: **/home/student**.

The same symbol can be used to refer to home directories of other users, using the following syntax: ~someuser. Notice that there is no slash between “~” and someuser. If you use the slash, such path refers to directory called “someuser” in your home directory.

Re-visit some of the previous steps and try the commands with relative path names. Remember that the starting point of a relative path name is always your present working directory.

Also, assuming you are in a sub-directory of your home directory, write a command to list the files in your home directory using ~ and ..

10. Create a directory called **lab1** under your home directory using the command: **mkdir lab1**. Note that once a directory is created, you cannot create it again - try creating the directory again and note the error message you get. Type the command **ls -l** to display the contents of the current directory. Make note of the output. Also try the two commands: **ls -l lab1** and **ls -ld lab1** and note the differences.

If the current working directory is your home directory what will be the **shortest** command (i.e. using absolute, relative, or relative to home) to create a new directory called **junk** under your home directory? Create the directory and write down the command you used.

11. You need to copy the **passwd** file into the **lab1** directory.

The shell command to copy files is called **cp**. Look through the manual before proceeding.

Type the command:

**cp /etc/passwd ~/lab1**

This will copy the **/etc/passwd** file into the **lab1** directory.

Now copy the same file and give it a new name with the command:

**cp /etc/passwd ~/lab1/new-file**

To copy a file to the current working directory and keep its name, you can use the command:

**cp /etc/passwd .**

The period at the end is needed – **cp** command always requires a destination and there is no default.

12. Once you have a copy of the **passwd** file in **lab1**, make **lab1** the current directory. You can use the **cat** command to display the contents of the **passwd** file. Because the file is too large to fit in the terminal screen, you can only use the mouse and scroll bar to see text that's gone off the terminal screen, which is not the most efficient approach. However, you should know commands that let you see large text files one page at a time – more and less.

13. Insert your USB flash drive into a USB port on the computer. After a few moments you should see a pop-up window. This window is showing you the contents of the USB stick. If the window or title does not appear, ask your professor or lab assistant for help.

Minimize the window and look for any new icons on the desktop. There should be at least one representing your USB stick.

When new media (like a flash drive, an external hard drive, an SD card, etc) is added to a system, it must be *mounted* to the file system. That is, it has to be added to the file tree somewhere so you can access it (so you can use commands like **cd**, **ls**, etc, to access it). Your USB stick was automatically added to the file system (*automounted*) by Linux. So where the heck is your USB stick in the file system? There are conventions for where new media is added to a Linux file system, although this sometimes varies from distribution to distribution (and can be changed in configuration files). Most often portable media appears as a directory of the **/media** directory. To confirm this, you can use the “up” arrow in the file browser and check the path that's shown above the files. Note the name of your USB stick.

You can also right click on the icon on the desktop and view properties on the Volume tab, which should show you the full path to the mount point. Do both to find the path and directory/volume name of your USB stick.

14. On the command line type: **ls -l /media/nnnnn** where nnnnn is the name of your USB stick. You should see the contents of the base directory of your USB stick. Replace nnnnn with your USB stick name in all the following commands.

At the command prompt, type **mkdir /media/nnnnn/lab1** to create a new directory on your USB flash drive. Use the **ls** command to verify that the new directory has been created.

Use the appropriate copy command **cp** to copy all the files in **/home/your-learnid/lab1** directory to the **lab1** directory on your USB flash drive. Use the **ls** command to verify that all the files have been copied to the USB flash drive. Although you can usually just unplug flash drives without issue, this can sometimes lead to problems if a read or write operation is still in process - that is, if not all the data has been transferred before you unplug the USB stick. To make sure all read/write operations have completed before you remove your USB stick, you should *unmount* it (remove it from the file tree). To do so, right click the icon on the desktop and choose the Unmount Volume option. When the icon disappears, it is safe to remove the USB stick.

15. You have created two new directories in your home directory: **lab1** and **junk**. Let's do some clean up before we log out. To delete or remove an empty directory, use the **rmdir** command.

Delete the **junk** directory now and verify your results by using the **ls** command.

Try deleting the **lab1** directory as well. You should get an error message. Do you know why?

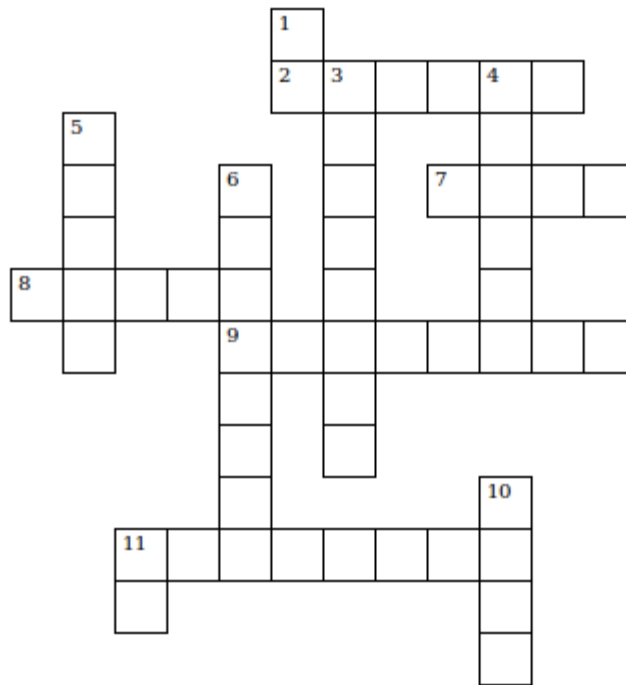
21. In order to delete the **lab1** directory, you must empty it first – you need to delete the **passwd** file. To delete (remove) a file, use the **rm** command. To remove the file called **passwd** in the **lab1** directory using relative path name, type the command **rm lab1/passwd**. Note that the previous command works only if the current working directory is your home directory, otherwise, you must type the command with the appropriate path name.

22. There are a number of directories found under the root directory that are common to all Linux systems (and most UNIX systems). Use your favourite web search engine to fill in the table below:

Directory	Used for:
/bin	
/boot	
/etc	
/home	
/root	
/tmp	
/var	

# Lab Review

After completing all steps above, complete the following crossword puzzle. Please make an effort to do this independently, as a form of self-review.



## ACROSS

- 2 File which stores user account names and their settings
- 7 UNIX system administrator
- 8 Command that lets you delete empty directories
- 9 Process of adding a storage device to a file system
- 11 Path name which starts with the current directory

## DOWN

- 1 Command that lets you copy files
- 3 Path name that starts with the root directory
- 4 Displays your user name
- 5 Command which displays system characteristics
- 6 Name of a program which runs a shell
- 10 Command that lets you see a file one screen at a time
- 11 Allows you to delete files