

Dear students.

This document contains additional work that you should do. Please do not ignore this as it draws your attention to specific topics and aspects of the module's contents and will be to your advantage in the examination. There are some things you are expected to do, some extra reading and some videos you can watch to help enhance your knowledge.

There seems to be confusion over the use of the "-" character and how Linux would interpret it. Consider -p, --help and -help. Each of these will be interpreted differently. The "-p" would execute the p-option for the command (command -p). If you have command --help you will get help for the command in question. A command like "command -help" will be interpreted as a request to execute the options "-h", "-e" "-l" "-p" with the command.

Sometimes a command is typed at the prompt, but Linux responds with a message saying that it does not understand the command even though we know the command is valid. This happens if you are not in the directory where the command in question resides, but in another directory. Including the path to such a command in the PATH variable will solve the problem. Not all commands are included in the PATH variable. When you type a command the shell will perform a series of checks in a particular order to determine the command to execute. Importantly this is done in the order given below:

1. Aliases
2. Shell reserved words
3. Function
4. Built-in command
5. File system command

In case these checks do not determine the command to be executed successfully, Linux informs you that the command is not recognized.

The input you type at the BASH prompt may be automatically completed by pressing the TAB key. Please be aware this is true for the following:

Commands  
Alias  
Function  
Variable  
Username  
hostname

At times it could be convenient to write more than one command on the same line especially when more than one command is required to complete a task. Assume you need to:

- format a document named *bulk.txt* using the *troff* command with the classical *me* macro package.
- pipe the document to the printer once formatting is complete.
- provide a means to determine how long formatting took.

This is all one task but it requires more than one command to do the sub tasks. You may then write something like:

```
date ; troff -me bulk.txt | lpr ; date
```

Read more regarding troff here: <https://www.computerhope.com/unix/utroff.htm>.

The BASH shell prompt can be customized using special characters. Some of these are listed below (There are more). Refer to your textbook to see what can be done with the BASH prompt. In addition please read here: <https://ask.fedoraproject.org/en/question/7000/how-to-set-colors-for-bash-shell-primary-prompt/>. Below are a few popular examples.

- |    |    |  |
|----|----|--|
| a) | \d | displays day name, month and day number for the current month. |
| b) | \u | displays current user name.                                    |
| c) | \! | current command history number.                                |
| d) | \w | full path to working directory.                                |

An *umask* command produced a result of *0002*. Do the following exercise: Write the default permissions in decimal for a new file just created. Do the same for a new directory just created. An *umask* command is valid for 1 session, meaning until you start Linux again. If you want an *umask* to be permanently enforced, put an *umask* command in the *.bashrc* file in your home directory. Some more detail is here: <http://www.informit.com/articles/article.aspx?p=686161&seqNum=7>.

The *vi* text editor uses a series of commands. Some of these are confusing in the sense that a lower case “i” for instance has a different meaning than an uppercase “I”. The editor is used regularly so be sure that you know these commands by heart as it is a very useful “on the go” skill.

The *nice* command is used to assign a priority in terms of CPU time to process. The way it works can be somewhat confusing since a smaller number give more CPU time. What you can do with *nice* is also dependent on your status i.e. user or owner or root etc. Refer to your book. Assigning CPU time to a process is a very important administrator skill. Please make sure that you are not confused. You can watch this <https://www.youtube.com/watch?v=dKuthDLIYSs> and this [https://www.youtube.com/watch?v=91rzP\\_2e2LA](https://www.youtube.com/watch?v=91rzP_2e2LA).

There are a number of shell script test expressions. As an exercise put them in a single table so that you can memorize them. This is very important “ready at all times” knowledge.

3.1. There are two centralized database types:

- a) LDAP
- b) Winbind

LDAP is the lightweight directory access protocol. It is used to provide directory services like phone books, addresses etc. It is an open standard that is configured in many types of computing environments. Winbind allows authentication of users against a Microsoft Active Directory server. You can watch this: <https://www.youtube.com/watch?v=di5hZZ8Ty8g>.

Below are a number of popular usermod commands used with some options. There are others but these are frequently used. Please refer to your textbook to learn about other options.

- a) usermod -l change the login name for the account
- b) usermod -L Lock the account
- c) usermod -m move/copy the content of the home directory to a new location
- d) usermod -o remove restriction that UID's must be unique

You can watch this video, even though the distribution here was not Fedora. It still applies.

<https://www.youtube.com/watch?v=Dw6Hv8ava9Q>

Consider the command below and analyse it, then write down all the facts that you can.

```
setfacl -m g:reception:rx /tmp/memo.txt
```

setfacl manipulates permissions for Access Control Lists(ACL's) and the -m option means the permissions are modified while the g option means the permissions modified is for a group called *reception*. rx means the group has read and execute permissions for the file /tmp/memo.txt Be sure that you can analyze a command like this in this way.

Do you know the steps followed to install a new hard drive? Refer to your book and make sure that you are familiar with it. In brief it comes down to:

Install the new medium (hard drive usb, etc) and then partition the new medium. Create the file systems on the new medium and mount the file systems.

The SWAP area is important in terms of the operation of the operating system. Do you know how to temporarily turn a swap partition on? Do you know how to make it available permanently, and do you know how to turn it off. Refer to your book and watch this

<https://www.youtube.com/watch?v=ewNsswGd75E>.

How can you find out what file system types that are currently loaded in your kernel? Use your bbook and find the answer to that question.