Lab 5: **The Linux Labs – Standard Linux Commands**

CSI4103 – Web Application Software Design

Faculty of Engineering – University of Ottawa

Objective:

Understand and practice how to use linux command line instructions. This is useful when working on server backends through a terminal window (e.g. using telnet or an SSH connection). This part of the lab continues on with a review of basic linux commands that are needed when navigating and using the file system from a linux command line

Instructions:

Linux files are organized within a hierarchical directory structure. We will work with two entities that exist within a directory structure, files and directories.

Each file and each directory is accessed via a path. That path can be expressed in two ways, as an absolute path and as a relative path. When working with files, Linux does not know where the files are located; you need to tell the operating system, the EXACT LOCATION of the file using either an absolute or relative path.

Most of the Linux commands will follow the following standard structure:

- `command –options –other_options argument(arguments…)`

  Example: `ls -la --color=never /etc/`
  where `ls` is the command, `-la` are two options, `--color=never` is also an option and `/etc` is an argument

To explore the directory structure, the following commands will be used:

- `cp` – copy files
- `mv` – move or rename files
- `rm` – remove or delete files
- `cat` – concatenate files. Used to display file contents
- `passwd` – to change a password
- `find` – to find files
- `man/info` - to display help
- `sleep` – to pause or wait
- `touch` – to change the timedate stamp of a file. Used to create an empty file.
- `clear` – to clear the screen

touch and cp

The **touch** command updates different timedate stamps. It is also frequently used to create empty files.

The **cp** command makes a copy of an existing set of files or directories (remember that directories are just a type of file) into another area of the file system. The syntax for the cp command is:

- **`cp [-options]` *`source destination`***
- Where source is often a directory and file specification and destination is often just a directory

The mkdir command

1. *touch clock*
2. *ls –l clock*

   Record the timedate stamp: _____

3. **sleep 61*

   What does this command do? _____

4. *touch clock*
5. *ls –l clock*

   Record the timedate stamp: _____

Copying files to a directory

a. **cd**

b. **touch f1 f2 f3**

c. **ls**

What does this show? _____

d. **mkdir lab**

e. **ls**

What does this show? _____

f. **cp f1 f2 f3 lab**

g. **ls lab**

What does this show? _____

h. **mkdir coffee**

i. **cd coffee**

j. **touch cream sugar**

k. **cd ..**

l. **cp coffee/cream coffee/sugar lab**

m. **ls –l lab**

Have the files been copied? _____

2. **<u>Copying directories to a directory (-r option) recursive</u>**

a. **mkdir dir1 dir2 dir3**

Record the command used to verify the directories have been created: ___

b. **cp dir1 dir2 dir3 lab**

Record one of the messages displayed: _____

c. **ls lab**

Have the directories been copied? _____

d. **cp –r dir1 dir2 dir3 lab**

   Record one of the messages displayed: _____

e. **ls lab**

   Have the directories been copied? _____

3. Copying directories to a directory (-r & --parents options)

   a. **mkdir parent/child**

      Record one of the messages displayed: _____

   b. **mkdir –p parent/child**

4. **cp –r --parents parent/child lab?????**

   List the contents of the lab directory:

   _____

   _____

5. Now try the *verbose* option of the copy command:

   `cp –v source destination`

   What is different? _____

   _____

mv and rm

The **mv** command moves or renames files. Recall that directories are just a type of file. The syntax for the **mv** command is:

- `mv` *`source destination`*

1. **Renaming files**
    a. **cd ~/lab**
    b. **ls**
    c. **mv f1 m1**
    d. **ls**

    Has the file been renamed? _____

2. **Moving files**
    a. **touch red green blue**
    b. **mkdir colours**
    c. **mv red green blue**

    Record the error message: _____

    d. **mv red green blue colours**
    e. **ls**
    f. **ls colours**

    What have you observed?_____

    _____

3. Try exercises 1 and 2 again, this time using mv with the **–i** option

    What does this option do? _____

4. Try exercises 1 and 2 again, this time using mv with the **–u** option

    What does this option do? _____

5. Try exercises 1 and 2 again, this time using mv with the **–b** option

    What does this option do? _____

6.
    a. **mkdir toddlers children sandbox**
    b. **mv toddlers children sandbox**
    c. **ls**

    Are there toddlers and children in the sandbox? _____

7. Try Exercise 6 again with the **–v** option

What does this option do? _____

The **rm** or remove file command allows you to delete the *content* of any directory. It is both a dangerous and useful command because of it flexibility. Unlike DOS or Windows, a file deleted in Linux is gone. (Is this always true? What do you think can prevent this from happening? When people have a problem, they usually invent a solution.) The syntax for the **rm** command is:

- `rm file_specification`

1. **Removing files**
    a. **cd**
    b. **cd lab/sandbox**
    c. **touch child1 child2 child3**
    d. **ls**

    Are there children in the sandbox? _____

    e. **rm child1 child2**

    Are there children in the sandbox? _____

    How do you know? _____

    f. **rm child3**

    Are there children in the sandbox? _____

    What command did you enter to confirm your answer? _____

    g. **cd ..**
    h. **rmdir sandbox**
    i. **cd**
    j. **rmdir lab**

    Record the error message: _____

    k. **rm –r lab**

    Has the directory been deleted? _____

    How do you know? _____

cat

Cat is a utility which will concatenate files and print on the standard output. It's power actually lies in the fact that is uses standard input and standard output which can be redirected and with pipes. It is often used to display small files to the screen (which is standard output). The syntax of the cat command is:

- `cat [options] [file]`

`clear`

- The clear command simply clears the terminal window of output. Try it! Remember there are still ways to review the previous output. You can use SHIFT-PG UP / SHIFT-PG DOWN or if you are in a graphical environment, you would normally have a scroll bar available.

Viewing files with cat.

a. **cat /etc/hosts**
b. **cat /etc/fstab**
c. **cat /etc/inittab**
d. **cat /var/log/messages**
e. **cat /var/log/messages | more**
f. **cat /var/log/messages | less**

passwd

The `passwd` command will change your password if you enter it without a username argument. If you are root or superuser, you can change any user's password.

If you are root, you can also use any password without regard to any password policies which may otherwise restrict certain password choices.

If you are a regular user trying to change your own password, the default password policies should prevent you from choosing trivial passwords.

1. **passwd**
    a. Record the message here: _____
2. Enter the following password: **sky**
    a. Record the message here: _____
3. **passwd**
    a. Enter your current UNIX password:
    b. Try to change your password to linux
    c. Were you successful? Explain: _____
    _____
4. **passwd**
    a. Try to change your password to 56?1Lin
    b. Were you successful? Explain: _____
    _____
5. You can now close the terminal window used for this exercise and open a new one to continue

Executing commands that are in the search path

1. **`echo $PATH`**
    - Record the path:

      _____

    - What is this?

      _____
2. **`whereis ls`**
    - Record the output:

      _____

    - What is the purpose of the whereis command?

      _____
3. **`which ls`**
    - Record the output?

      _____

    - What is the purpose of the which command?

      _____
4. Describe the circumstances you would use these commands?

   _____
5. What are the differences between the **`whereis`** and **`which`** commands?

   _____

   _____
6. **`ls`**
    - Did the command work? Why? _____
7. **`whereis runlevel`**
    - Record the output: _____
8. **`runlevel`**
    - Record the output: _____
    - Did the command work? Why? _____
9. **`/sbin/runlevel`**
    - Record the output: _____
    - Did the command work? Why? _____
    -

Executing commands that are not in the search path.

- **`echo $PATH`**
    - Record the output: _____
- **`PATH=`**
- **`echo $PATH`**
    - Record the output: _____
- **`ls`**
    - Did the command work? Why? _____
- **`/bin/ls`**
    - Did the command work? Why? _____

      _____

10. **`exit`**
    - To log out. This will "fix" your path.
11. Log back in (or su – if appropriate)

- **`echo $PATH`**
    - Record the output? _____
- **`mkdir bin`**
- **`cp /bin/pwd  mypwd`**
    - What did the command do? _____
- **`mypwd`**
    - Did the command work? Why? _____
- **`./mypwd`**
    - Did the command work? Why? _____
- **`cp /bin/pwd  bin/mybinpwd`**
- **`mybinpwd`**
    - Did the command work? Why? _____
- _____
- **`rm  mypwd  bin/mybinpwd`**
- **`whoami`**
    - What is your current login id? _____
    - Record your current prompt

- **`id`**
  - Record the output: _____
- **`echo $PATH`**
  - Record the output: _____
- **`su -`**
  - Enter the root password
- **`whoami`**
  - Record the output: _____
- **`id`**
  - Record the output: _____
- **`echo $PATH`**
  - Record the output: _____
- **`runlevel`**
  - Did the command work? _____
- **`exit`**
- **`whoami`**
- **`id`**
  - What is your current login id? _____

Review

This exercise assumes that the commands listed below are executed in a **regular** user's home directory. If you want to keep things neat, create a new user just for the purpose of performing this review. Keep notes so that you can answer the questions at the end of the exercise.

Not all commands will be successful! Make SURE that you have entered everything correctly!

```
mkdir   ~/labex

cd  labex

mkdir  ./orchard

touch  apple  orange

mv  orange  orchard/lemon

cat  orange

touch  lettuce  tomato  cucumber

cp tomato  lettuce  garden

mkdir  jardin  forest

mv lettuce  cucumber  jardin

rmdir  garden

cd  ..

touch  test

cd  orchard

cd  labex/forest

mv  tomato  forest
```

1. How many **directories** are **created** during the review exercise? _____
    a. List them: _____

    _____
2. How many **directories** are **deleted** during the review exercise? _____
    a. List them: _____

    _____

3. How many **regular files** remain in the directory `labex`?
   Do not include subdirectories. _____

   a. List them: _____

   _____

4. How many **regular files** remain in the directory **orchard**?
   Do not include subdirectories. _____

   a. List them: _____

   _____

5. How many **tomato** files remain in the **labex** directory and all its **subdirectories**? _____

   a. List the directories: _____

   _____

6. How many **unsuccessful** attempts creating (using **touch** or **cp**) or removing **regular files**? _____

   a. List the errors: _____

   _____

7. How many unsuccessful attempts creating and removing **directories**? _____

   a. List the errors _____

   _____

8. What is the **current** directory at the end of the exercise?

   _____

1. How do you know? What command did you enter to find out?

   _____