Lab 1 **CS5296** **Cloud Computing: Theory and Practice**

Student Name: _____

Student No. : | | | | | | | | |

Lab Section: ☐ Wednesday 18:00 - 18:50 ☐ Wednesday 21:00 - 21:50

SSH and Basic Linux Commands

Introduction

Topics to be covered in this lab include:

1. Logging in to CityU CSLAB's Ubuntu Linux server (SSH Gateway).
2. Basic Linux commands (ls, nano, pwd, cd, man, whatis, mkdir, rmdir, cp, mv, rm, and clear).

Points to note about Linux commands:

1. Unlike DOS, Linux is ***case sensitive***, therefore all commands must be typed in the appropriate case, e.g. ls is different to LS.
2. In Linux the directories in a path are separated by a ***forward slash*** /, e.g. /home/grads/cctom2.

Acknowledgement

This lab was adapted from http://glasnost.itcarlow.ie/~mcmanusa/notes/cfy/Linux%20Labs/.

Logging in to the Linux server

- Start the SSH client **PuTTY** from *Work Desk Menu*, or use the web SSH client in a web browser (https://gateway.cs.cityu.edu.hk).
- Login to the Linux server using the following details:

  Host Name: gateway.cs.cityu.edu.hk

  Username: your EID (e.g., cctom2)

  Password: your password

Your password will not be shown on the screen as you type it, not even as a row of stars (******).

After successfully logging in, the shell will always give you a prompt if it is ready to accept commands. A shell prompt normally ends in a `$` sign like this:

```
cctom2@ubt16a:~$
```

Some shell prompts use `%` or > instead, and give more information, such as:

```
ubt16a:/home/grads/cctom2>
```

NOTE: Never copy/type the shell prompt used in this lab. Please don't forget to log out (use the `exit` command) after you finish your work.

A sample Linux file system

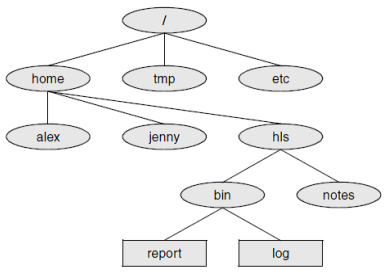
Paths:

- Root is /
- Paths separated by /

e.g.,

/home/hls/notes

/home/alex



The Linux directory structure is like a tree. The base of the Linux file system hierarchy begins at the **root**. Directories branch off the root, but everything starts at root.

More details here: https://bit.ly/2kcbpB5

In the example above, write the full path to the `report` directory: .....

ls (list - directory listing)

The **ls** command lists the contents of the current directory, across the screen in several columns.

Key in **ls**, then press the enter key.

```
cctom2@ubt16a:~$ ls
Windows  www
```

Files and directories will be listed. *Nothing appears if you have no files yet in your current directory.*

Options (arguments)

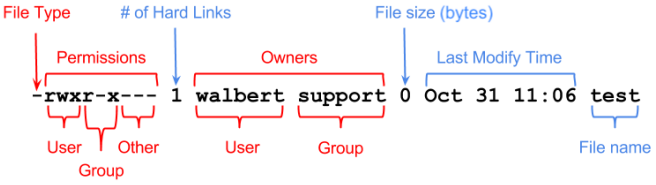
An option changes the behaviour of a command. The **ls** command can be used with several options. An example of an option that can be used with **ls** is **-l**.

Key in **`ls -l`**

Your screen should look similar to the screenshot below:

```
cctom2@ubt16a:~$ ls -l
total 8
drwx------x 1 cctom2 grads 4096 Jan  9 13:34 Windows
drwx------x 1 cctom2 grads 4096 Jan  9 13:40 www
```

Lower-case letter l.



What effect does this option have? What are **ls -l** output columns? What is the size of each file in bytes for your output? ……

Let's create a file using the **nano** text editor program.

Type **nano**

- Key in the following text:
  `Welcome to Cloud Computing course.`
- Note the various options you can use in this editor on the bottom of the screen.
- Note the option called **Write Out** and the symbol **^O** beside it. This means you need to hold down the **Ctrl** key and the letter **O** to access this option.
- After you have keyed in the text, choose the **^O Write Out** option by pressing Ctrl and O (This is the same as the Save option in a word processor/editor).
- You are prompted with the message **File Name to Write** - key in **welcome.txt** and press enter.
- Choose the **^X Exit** option by pressing Ctrl and X.

Leave the nano editor after having saved your file. Key in the command below: **`ls -l`**

```
cctom2@ubt16a:~$ ls -l
total 8
-rw-r--r-- 1 cctom2 grads   35 Jan  9 14:34 welcome.txt
drwx------x 1 cctom2 grads 4096 Jan  9 13:34 Windows
drwx------x 1 cctom2 grads 4096 Jan  9 13:40 www
```

pwd (print working directory)

The **pwd** command will show you the path to your current working directory. Unlike our ssh gateway server, some other Linux prompts may not show your working directory. So you can use **pwd** to find out where you are in the directory tree.

Type **pwd** command to view the path of the directory you are currently in.

This is your **current directory**. What is it? ……/home/ms19/yuqiaweng2……

cd (change directory)

The **cd** command changes your current directory. Use the command **cd ..** to go one level up in the directory tree. Key in this command ***repeatedly*** until you can no longer go back any further in the directory tree. You are now in what is called the ***root*** directory.

```
cctom2@ubt16a:~$ cd ..
cctom2@ubt16a:/home/grads$ cd ..
cctom2@ubt16a:/home$ cd ..
cctom2@ubt16a:/$ cd ..
cctom2@ubt16a:/$
```

Notice double dot (..) being used;

. (dot) represents the directory you are in;

.. (dot dot) represents the parent directory.

Linux uses forward slashes to separate the directory names. The root directory is indicated by the single forward slash in the above screenshot.

Do a *long* directory listing (remember the **-l** option mentioned earlier?) The screen should look similar to the one below. The forward slash (on the first line) indicates that your current directory is the "root" directory.

```
cctom2@ubt16a:/$ ls -l
total 147
drwxr-xr-x   2 root root   12288 Jan 11  2019 bin
drwxr-xr-x   4 root root    4096 Jan 11  2019 boot
drwxr-xr-x  17 root root    3860 Jan 12  2019 dev
drwxr-xr-x 173 root root   12288 Sep  4 11:00 etc
drwxr-xr-x   8 root root       0 Sep  5 20:07 home
-rw-r--r--   1 root root      44 Nov 20  2018 ubar.txt
(content removed for brevity, the same hereinafter.)
```

Note the d in column 1 (lines 3 through to 7) of the above screenshot. The d indicates a *directory*. So bin, boot, dev, etc and home, and etc are all directories. In line 8 (last line), column 1 (there is no d in the first position). This indicates that ubar.txt is a *file*.

Exercise: Change back to your home directory using the sequence in the screenshot below:

```
cctom2@ubt16a:/$ cd home
cctom2@ubt16a:/home$ cd grads
cctom2@ubt16a:/home/grads$ cd cctom2
cctom2@ubt16a:~$
```

⚠ You should change the login id in the line that reads ***cd cctom2*** to your login id. You should also change ***cd grads*** to your own group, e.g., bsft18, elft19, or grads.

Repeat the Exercise: Use the **cd** command to change to the root directory and then change back down to your home directory.

Instead of keying in ***cd ..*** several times to change to the root directory, we could have used the command **cd /**. This will change the current directory to the "root" directory (no matter which directory is your current directory.**)**

Instead of keying in ***cd home***, ***cd grads***, and ***cd cctom2*** to go back to home directory. We could have used **cd /home/grads/cctom2** command.

At any point, you can key in the following command to take you to your home directory. Note: no arguments have been supplied to the cd command.

```
cctom2@ubt16a:~$ cd
```

~ (represents your home directory)

You can also use ~ at the start of a path name so that that path starts at your home directory. For example, the command **ls ~/reportFiles** will do a directory listing of the reportFiles directory that is a subdirectory of your home directory. This will work no matter where you currently are in the directory structure. Other examples of its use are:

```
cd ~
rm ~/welcome.txt
```

Exercise:

Change to the **root directory** using a single command. What command did you use?

cd /.

Change back to your **home directory**. Where your **home directory** is, will depend on what account you are logged in as. What is the full path of your home directory?

/home/ms19/yuqiaweng2

Exercise:

1. Use **ls** to view all files in the **root** directory (/):

2. Change to the */home* directory:

cd home

3. Use **ls** to view all files in the */home* directory:

4. What command would you use to go **directly** to your home directory from any other directory?

cd /home

5. Change back to the root directory

cd .. or cd /.

man (reference manual for getting help)

To bring up help on a command, use the **man** command. For example to bring up help on the **ls** command you would key in the following:

```
cctom2@ubt16a:~$ man ls
```

Note: While you are in the help:

Pressing ***enter*** or down arrow key (↓) will allow you to scroll down through the text.

Pressing ***q*** will allow you to quit from the help.

What does the **-a** / **-l** (letter l) / **-1** (number 1) option do for the **ls** command?

What is the difference between the -g and -G options for the **ls** command?

Some commands also provide a long option like **`--help`** to display usage help, e.g.,

```
ls --help
```

Exercise:

1. View the man page for the **mv** command.

2. Display the usage help of the **mv** command.

mkdir (make directory)

The **mkdir** command will allow you to create a new directory. To create a subdirectory in your current directory, use **mkdir** command followed by the name of your new directory, e.g.

```
mkdir cloudcomputing
```

To create a directory inside a directory other than your current directory, use **mkdir** followed by a path to your new directory, e.g.

```
mkdir cloudcomputing/mydir
mkdir -p nonexistdir/dir2
```

What does each of these commands do?
i.e. where is the new directory being created?
What does the -p option do for the **mkdir** command?

Exercise:

1. Create a new directory called reportFiles, in your home directory.

……

2. Do a directory listing of your home directory.

……

3. Create a file in the directory called reportFiles called cloudcomputing.txt and write some texts to it.

……

4. Do a directory listing of the reportFiles directory.

……

5. Without changing to the **reportFiles** directory, create inside it a new directory called **backup**.

……

6. Change into the **reportFiles** directory and check for yourself that the backup directory was created by your previous command.

……

rmdir (remove directory)

The **rmdir** command will delete a directory. The directory that you wish to delete must be **empty** before it can be deleted. To delete a directory type **rmdir** followed by the name (and path if needed) of the directory to be deleted. E.g.

```
rmdir cloudcomputing/mydir
rmdir cloudcomputing
```

cp (copy)

The **cp** command allows you to copy a file from a source location to a destination location. To use it, use **cp** followed by the path to the source file, followed by the path to the destination, e.g.

```
cp file1.txt backup/file1.txt
cp backup/file1.txt .
```
Notice the . (dot) being used

The last example above, copies the file, file1.txt, from the subdirectory backup into your current working directory.

You can also use **cp** to copy a file and save the copy under a new name, e.g.

```
cp file1.txt file2.txt
cp file1.txt backup/file2.txt
```

Commonly used option:

-R, -r, --recursive: copy directories recursively

E.g., to copy directory backup and its contents to a new directory, run:

```
cp -r  backup backup2
```

To do some of these next exercises, you will need to create a few files. You can use the Nano text editor to create a few files for working with. Call them **myfile.txt** and **new.txt**. Store them in your home directory. You can put any text that you like in these files.

Exercise: Try not to move from your home directory for each of the questions below.

Create a subdirectory in your home directory and call it **backup**.

……

Copy **myfile.txt** into **backup**, keeping its original name.

……

Copy **new.txt** into **backup** and call the destination file **new.bak**

……

Copy **new.bak** from the **backup** directory to your **current directory**.

……

Create a directory called **letters** in your current working directory (home directory)

……

Copy **new.bak** from the **backup** directory to **letters** directory and call the new file (the destination file) **new2.bak**

……

mv (move)

The **mv** command allows you to move a file from one location to another. To do this, type **mv** followed by the path to the source file, followed by the path to the destination, e.g.

```
mv file1.txt backup/file1.txt
mv backup/file1.txt .
```

Notice the . (dot) being used

It can also be used to rename a file, e.g.

```
mv file1.txt file2.txt
```

The last command will rename file1.txt in your current directory to file2.txt. Unlike the **cp file1.txt file2.txt** command, you will not be left with a file called file1.txt as well as the file file2.txt.

Exercise:

1. Move the file **new.txt** into your **backup** directory.

……

2. Without changing to the **backup** directory, move the file **new.txt** from the **backup** directory into your current working directory.

……

3. Rename the file **new.bak** to **new2.txt**, using the **mv** command.

……

rm (remove)

Use **rm** to delete (remove) a file. To delete a file, type **rm** followed by the name of the file you want to delete (you can supply a path to the file if it is not in the current working directory), e.g.

```
rm file1.txt
rm backup/file1.txt
```

Commonly used options:

```
-r, -R, --recursive: remove directories and their contents recursively
-d, --dir: remove empty directories
```

Exercise:

1. Delete the file new2.txt.

……

Verify that it has been removed by issuing the **ls** command.

2. Delete the file in your backup directory called myfile.txt.

……

3. Change directory to the backup directory and then delete the file myfile.txt in your home directory.

……

4**.** Write the Linux command to delete the folder **backup** and its contents.

……

clear (clear screen)

To clear the screen of all the previous commands, type **clear**. Try this out. Alternatively, you may use Ctrl+L shortcut key.

Summary

Basic Linux commands

ls	List the contents of the current directory
nano	Linux editor
pwd	Show the full path of where you are
cd	Change directory
man	Help in Linux
mkdir	Make a directory/folder
rmdir	Delete/remove a directory
cp	Copy a file or group of files
mv	Move a file or group of files
rm	Delete a file
clear	Clear the screen