UEEN3113 / 3413 Server Configuration and Management

Lab 1

Part 1: Installation

Tasks:

1. Create a virtual machine in VMware workstation.
2. Install Ubuntu server to the newly created virtual machine.
3. Explore **apt** and **apt-*** command.

Part 2: Familiarise with Common Commands

pwd	cp	cat	free	kill	tar	lshw
ls	mv	more	df	chmod	gzip	head
cd	rm	less	du	chown	history	tail
mkdir	touch	grep	top	mount	clear	2>
rmdir	vi	find	ps	umount	lsblk	>&

Path name

The default system root directory is **/**. (Not to confuse with the root user directory, which is **/root**). By default, user's home directory will be located at /**home**. For example, an user with user name "john", will have his home directory located at **/home/john**.

File or directory can be specified in **full path name** or **relative path name** (location relative to current working directory).

For example, there are two sub directories in /home/john

- doc1, contains the file intro.txt
- doc2, contains the file backup_intro.txt

Full path names for all files are:

- /home/john/doc1/intro.txt
- /home/john/doc2/backup_intro.txt

Assume that the current working directory is /home/john/doc1, the relative path name of backup_intro.txt is **../doc2/backup_intro.txt**.

Command Usage

1. **cd**

Syntax: **cd** *directory_name*

NOTE:
When terminal is launched, the default working directory is the user's home directory. You can always check the current working directory with the **pwd** command.

Examples:
To change into doc1 directory in john's home directory: **cd doc1**
To change into doc2 directory from doc1, using full path name: **cd /home/john/doc2** or **cd ~/doc2**
To change into doc2 directory from doc1, using relative path name: **cd ../doc2**

2. **mkdir**

Syntax: **mkdir** *directory_name(s)*

Examples:
mkdir doc1 doc2
mkdir –p lab1/doc

3. **cp**

Syntax: **cp** *file(s) destination*
Examples:
cp intro.txt ../doc2 (copy the file to doc2 directory)
cp content.txt ../doc2/backup_content.txt (copy the file to doc2 directory and with new file name)

4. **rm**

Syntax: **rm** *file(s)*
By default, it does not remove directories. Use the -r option to remove directories and their contents.
Examples:
rm intro.txt (remove a file)
rm –r doc2 (remove all files in docs recursively and also remove the doc2 directory)

5. **mv**

Syntax: **mv** *file destination*
Examples:
mv intro.txt intro_old.txt (rename the file, in current directory)
mv intro.txt ../doc2/backup_intro.txt (move the file to doc2 and rename the file)

NOTE: Check the command option by using the **--help** option or **man** page. Example: ls –help

Creating file

touch

Syntax: **touch** *file(s)*
Example: **touch** intro.txt content.txt

echo with redirection

Syntax: **echo** content to be written **>** file_name
Example: **echo** "Created using echo..." **>** file_a

cat with redirection

Syntax: **cat >** file_name
Content line 1
Contnet line 2
[CTRL + D]

Example:

```
cat > simple_script
echo "Hello, welcome..."
echo "This is a text file that contains a few commands."
echo "Created using cat"
echo Today is $(date).
echo Enjoy the learning.
[CTRL + D]
```

Use **cat** to combine files.
Example: **cat** file_a file_b **>** file_c

vi editor

Syntax: **vi** file_name
Example: **vi** file_a
The command will open file_a or create it if it doesn't exist.

Cursor movement:
Using arrow keys *OR* commands as listed below.

nG	Moves to line n in the file.
G	Moves to the end of file.
w	Forward one word.
b	Back one word.
0 (zero)	Moves to the beginning of the current line.
$	Moves to the end of the current line.
+	Moves to the first character in the next line.
CTRL-d	Down 1/2 screen.
CTRL-u	Up 1/2 screen.
Page Down	Forward one screen.
Page Up	Back one screen.
H	Moves to the top of the screen.
M	Moves to the middle of the screen.
L	Moves to the bottom of the screen.

Editing commands:

x	Delete the character at the cursor.
dd	Delete the current line.(Can use for cut-and-paste, the deleted line is in the buffer)
dw	Delete the word starting at the cursor.
d$	Delete from the cursor to the end of the line.
d0	Delete from the cursor to the start of the line.
yy	Copy the line into buffer.
p	Paste text from buffer.
u	Undo
.	Repeat the most recent change.

General:

i	Change to insert mode to edit the file. (Default mode is command mode)
ESC	Change to command mode to use the commands above.
:w	Save changes.
:x or :wq	Save changes and exit.
:q	Quit without saving.

Command line editing

Use arrow keys (↑ and ↓) to step through previous commands in the history list and select a command line.

Useful keystrokes:

Ctrl+A / [HOME] : Go to the beginning of the current line.
Ctrl+E / [END] : Go to the end of the current line.
Ctrl+C : Delete the entire line.
Alt+F : Go forward one word.
Alt+B : Go backward one word.
Alt+U : Change the current word to uppercase.
Alt+L : Change the current word to lowercase.
Alt+C : Change the current word to an initial capital.

Output of a command can be passed to another command through **pipe** (**|**).
Syntax: command_1 | command_2 | command_3
Example:

```
ls /usr/share | sort | less
```

To sort the output in alphabetical descending order, press the up arrow key to get the previous command, use the appropriate keystrokes and change the command to

```
ls /usr/share | sort -r | less
```

To view the command history list (stored in *.bash_history*), use the **history** command.
Each command in the list is preceded by a number, which can be used to run the command. For example, to run a command numbered 24, type **!24** and press [ENTER].

To clear the history in the current shell, run history command with the **-c** option: **history -c**
To overwrite the history file, run the history command with the **-w** option: **history -w**

Commands and Operators

The **&&** Operator
Syntax: command1 && command2
Execute command2 **ONLY IF** command1 executes successfully.

Examples:
ls && pwd
lsss && pwd

The **||** Operator
Syntax: command1 || command2
This will execute command2 **IF** command1 fails

Examples:
ls || pwd
lsss || pwd

Command Expansion

To write command efficiently.
Suppose that we are going to create files namely **part-1, part-2, part-3, part-5, part-6, part-9**
Using the **touch** command, it will be written as:

touch part-1 part-2 part3 part-5 part-6 part-9

An efficiently way to write the command to achieve the same effect:

touch part-**{1,2, 3,5,6,9}**