Munster Technological University

Computer Science Dept.

COMP6042 Operating Systems in Practice

Spring 2024

Lab 6

Date: Week-starting March 18**,** 2024 - your COMP1-group scheduled Lab-class.

- Attendance at your Lab class is strongly recommended.
- You may be asked show your lecturer your working questions.
- You will need to make a ***vokoscreen recording*** for ***Q1*** and ***Q3*** to show your work.
- *For answering descriptive questions, , we recommend **not to use cut & paste**; use **your own words**.*

- Login to your ***Linux Ubuntu V22.04.x LTS*** virtual machine.
- ***Download this*** pdf-document (in a folder created for this module and Lab class).
- ***Create & Open*** your solutions-file, ***answer*** the questions, ***save*** the file.
- Use the Snipping tool program, to copy extracts from your terminal sessions into your document, to help answer questions.
- This is a practice Lab. No Canvas Submission

Document as much as possible in your report. Take snipping of the terminal showing the sequence of commands you type…

Before you start the questions:

- Create a directory (i.e. folder) in your home directory called **Lab_6**.
- Therefore, open a terminal and then type:

```
cd ~
pwd
mkdir Lab_6
ls -l          ← Make sure you can see the directory Lab_6 in the list.
```

- Finally, close the terminal.

Question 1 - Make a ***brief vokoscreen recording*** which displays the ***content*** of your ***Script1*** file followed immediately by your ***Script1*** file ***executing – No video editing permitted***

*Create and run a **simple bash script** program. Make the file a program file by changing the permissions. Describe the function and the use of '**$'** in the program.*

1. Open a terminal.
2. Change to your home directory by typing: **cd ~**
3. Change to the Lab_6 folder by typing: **cd Lab_6**
4. Check that you are in the correct directory by typing: **pwd**
5. Type **nano Script1** and then enter the following data for the ***14 line*** file

   [**caution**: take care that you type the 14 lines correctly; the script program is ***case sensitive***, so use small 'e' for echo… Also, in lines 6, 7 & 11, the '=' symbol must not have a space before or after.]

   [**reminder**: a program starts in column 1. Do not use the ***tab key***, use ***space-bar***.]

```
#!/bin/bash
echo  This is a sample program to illustrate echo
echo  "You can place the string between double quotes"
echo  $HOME
echo  $PATH
MYVAR1=0324         # Hash causes the remaining line to be a comment
MYVAR2=112          # You can define your own variables such as MYVAR1
echo  Display the number $MYVAR1 and $MYVAR2
echo  Todays date is $(date)
echo  The Linux kernel version is $(uname  -r)
IFS=
echo  The first 9 lines of the cpuinfo file are printed
echo  $(head   -9   /proc/cpuinfo)
echo
```

6. Save the file and exit nano.
7. Display the ***file permissions*** of the file - *r w - r w - r - -* by typing: **ls –l**.
8. Change the ***file permissions*** of the file to make it a program by typing: **chmod u+x Script1**
9. Display the ***file permissions*** of the file, which should now be by typing: **- r w x r w - r - -** **ls –l**

   Note: the change to *rwx* for the ***user***.

10. Run the script as a program by typing: **./Script1**
11. In your own words, describe the different uses for the '**$'** symbol, as employed in Script1 above. [~10 lines]
12. Close the terminal.

Question 2 - **No vokoscreen recording** for this question

*Create 8 empty **files** and change their **permissions**. Refer to your slides on Canvas (Cptr. 6).*

Take snipping of the terminal showing the sequence of commands you type

1. Open a terminal.
2. Change to your home directory by typing: **cd ~**
3. Change to the Lab_6 folder by typing: **cd Lab_6**
4. Check you are in the correct directory by typing: **pwd**
5. Create 8 empty files called *test1, test2, test3, test4, test5, test6, test7* and *test8* by typing **touch test1**, then type **touch test2**, **touch test3** and so on.
6. Look at the permissions of the 8 files by typing: **ls –l test***
   (**Note**: * is the wildcard).
7. Each file will have the permissions: - r w - r w - r - -.
   u g o

   The first 3 are **u,** for ***user,*** permissions; the next 3 are **g,** for ***group,*** permissions; the last 3 are **o,** for ***other,*** permissions (i.e. everyone else).

File test1 → Change permission using ***chmod*** and options ***u/g/o***

1. Change permissions of test1 to - r w x r w - r - - , by typing:
   ls -l test1
   chmod u+x test1
2. Check permissions by typing: **ls –l test1**

File test2 → Change permission using ***chmod*** and options ***u/g/o***

1. Change permissions of file2 to - r w - - - x r - -, by typing:
   ls -l test2
   chmod g-r test2
   chmod g-w test2
   chmod g+x test2
2. Check permissions by typing: **ls –l test2**

File test3 → Change permission using ***chmod*** and options ***u/g/o***

1. Change permissions of file3 to - r w - r - - r w x, by typing:
   ls -l test3
   chmod g-w test3
   chmod o+w test3
   chmod o+x test3
2. Check permissions by typing: **ls –l test3**

File test4 → Change permission using ***chmod*** and options ***u/g/o***

1. Change permissions of file4 to - - w - - w - r - -, type *your own* ***chmod*** commands which use the u/g/o options.
2. Verify your answer by typing: **ls –l test4**

File test5 → Change permission using ***chmod*** and options ***u/g/o***

1. Change permissions of file5 to - r - x r w x r - x, type *your own* ***chmod*** commands which use the u/g/o options.
2. Verify your answer by typing: **ls –l test5**

File test6 → Change permission using ***chmod*** with ***octal*** format

1. Change permissions of file6 to - r w x r w x - - -, by typing **chmod 770 test6**
2. Verify your answer by typing: **ls –l test6**

File test7 → Change permission using ***chmod*** with ***octal*** format

1. Change permissions of file7 to - - w x - w x r - -, type *your own* ***chmod*** command which uses octal format.
2. Verify your answer by typing: **ls –l test7**

File test8 → Change permission using ***chmod*** with ***octal*** format

1. Change permissions of file8 to - r w - r - - r - x, type *your own* ***chmod*** command which uses octal format.
2. Verify your answer by typing: **ls –l test8**

- Close the terminal.

Question 3 - Make a ***vokoscreen recording*** which displays the ***content*** of your ***Script2*** file followed *immediately* by your ***Script2*** file ***executing – No video editing permitted***

*Write a **shell script** program to display variables on a HTML webpage using **firefox**. Run this program. Finally, modify the program adding some of your own ideas.*

1. Open a terminal.
2. Change to your home directory by typing: **cd ~**
3. Change to the Lab_6 folder by typing: **cd Lab_6**
4. Check you are in the correct directory by typing: **pwd**
5. Type **nano Script2** and create a script program which displays user variables as follows:

```
#!/bin/bash
# Program to output HTML to firefox.
# First we redirect all the output displayed by echo to the file called 'hold.html'.
# Then we send this .html file to firefox for display.
TITLE="Displaying HTML on the firefox browser "
MyNumber=2793
echo "
<HTML>
    <HEAD>
        <TITLE>$TITLE</TITLE>
    </HEAD>
    <BODY>
        <H1>A NEW HEADER</H1>
        <P>My display will have 3 lines,<br>
            Line2 shows MyNumber =$MyNumber<br>
            Line3 shows the last line.<br></P>
    </BODY>
</HTML>" > hold.html
firefox hold.html
```

6. Type **ls –l**
7. Change the permissions of **Script2** to **rwxrw-r--** by typing: **chmod u+x Script2**
8. Run the script as a program by typing: **./Script2**
9. In your own words, describe how this program works; details required. [~10 line]
10. Copy the program to a new file by typing: **cp Script2 Script3**
11. Now type **nano Script3**, and edit this file.
12. Add your own ideas to show that you can use elements of HTML.
13. Edit the program so that it displays some useful information
    [***hint***: look at the program in Q1 which prints information).
14. Save **Script3**
15. Run **Script3**.

End Lab 06