

SUMMER SCHOOL 2021

Bash Scripting Part 1

Dr Krishna K. Govender

Summary of commands used in this lecture

Command	Meaning
mkdir	Make a new directory/folder
ls	List items in current directory
nano	Command line text editor in Linux
Ctrl+X > Y > Enter	Sequence used when saving changes to a file created/edited in nano
cat	Concatenate a file/s
chmod	Modify file permissions

- As in the previous lectures all commands provided in **BOLD** and center aligned from here onwards are commands that can be typed/executed directly in a terminal.

- With all material that you have covered up to now we have only been running commands within the terminal.

Although this is a good approach to use when running simple commands or if you wish to test something, it is not very efficient.

For instance, if you declare a variable in the terminal, the moment you exit the terminal and open a new one that variable disappears.

- It is often a good idea to place your Linux commands into a script, which will allow you to may use of the commands present in the script whenever you want.

- Scripts/Scripting also allows you to expand the information contained therein, so what might have started off as a very simple script can later evolve to be something more complex which you can then run to process data from your simulations, for example.

- The writing of scripts is best done in a text editor.

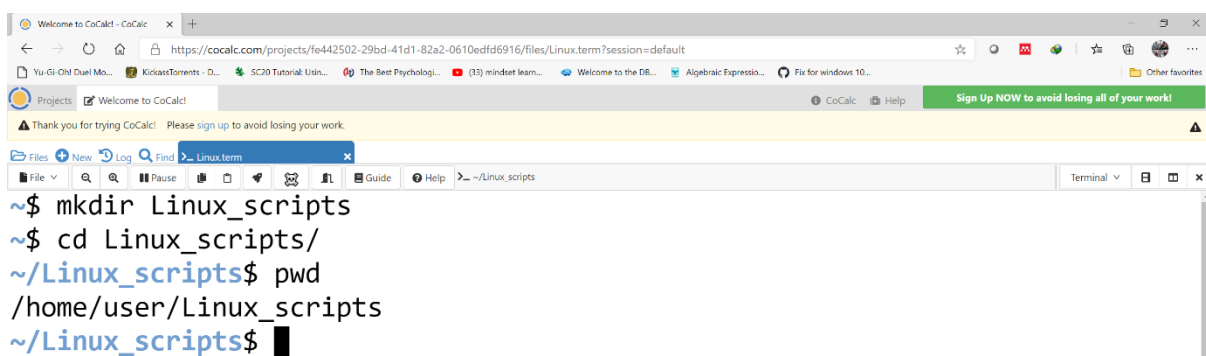
For the scripting part of this course I will make use of **nano** as my text editor, but you are welcome to use whatever you feel most comfortable with.

- First, we will create a folder where we are going to store all the scripts that we write from this point onwards:

mkdir Linux_scripts

cd Linux_scripts

pwd



The screenshot shows a web browser window with a CoCalc terminal. The terminal output is as follows:

```
~$ mkdir Linux_scripts
~$ cd Linux_scripts/
~/Linux_scripts$ pwd
/home/user/Linux_scripts
~/Linux_scripts$
```

- Now we will make our first script called **script1.sh**:

nano script1.sh

The screenshot shows a CoCalc terminal window. The user is in the directory `~/Linux_scripts` and has just opened `script1.sh` with the `nano` editor. The terminal shows the following content:

```

GNU nano 4.8 script1.sh
#!/bin/bash

# This is your first script
# It will just print something to the screen
echo 'Hello'

```

At the bottom of the terminal, there is a help menu for nano:

```

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line

```

- Use **Ctrl+X > Y > Enter** to exit **nano**, save the changes and end up back on the terminal.

Check that the file exists and that it has the information we require:

ls

cat script1.sh

The screenshot shows the same CoCalc terminal window after exiting nano. The user has run the `ls` command, which lists `script1.sh`, and then the `cat script1.sh` command, which displays the contents of the file:

```

~/Linux_scripts$ nano script1.sh
~/Linux_scripts$ ls
script1.sh
~/Linux_scripts$ cat script1.sh
#!/bin/bash

# This is your first script
# It will just print something to the screen
echo 'Hello'
~/Linux_scripts$

```

- We named our file **script1.sh**.

The first part of the script could have been called anything you like, but the file extension must be **.sh**.

This extension is used for any bash script, just as any file ending with **.py** is used for scripts that are written for python.

- The first line of a bash script is called the shebang (**#!/bin/bash**) and it points to the location of your program/binary that will be used to execute the script.

- All lines that appear after the shebang which start with **#** are used for comments and these are ignored by bash.

It is good to have comments in your scripts to allow you to know what different parts of the script does as well as to assist others that may make use of scripts later.

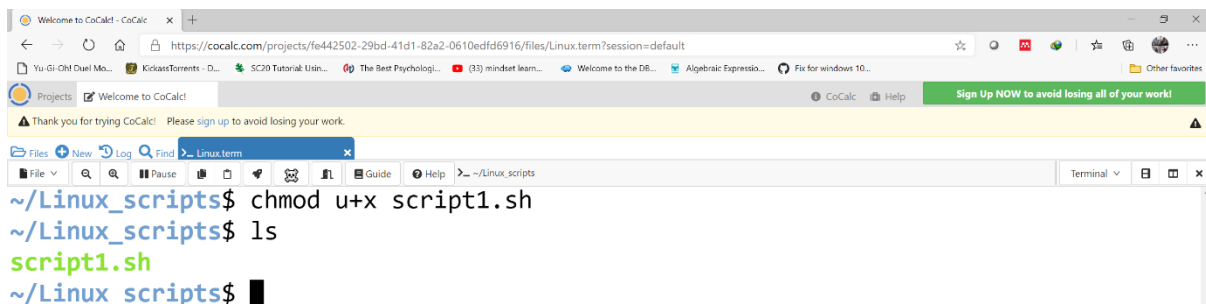
- We also have an **echo** statement present in our script and as we know from previous lectures this should print something (“Hello”) to the screen.

- Now that the script is written let us try to run it.

The first time you need to run a bash script you have to make sure that it is executable:

chmod u+x script1.sh

ls



```

Welcome to CoCalc - CoCalc
https://cocalc.com/projects/fe442502-29bd-41d1-82a2-0610edfd6916/files/Linux.term?session=default
Projects Welcome to CoCalc CoCalc Help Sign Up NOW to avoid losing all of your work!
Thank you for trying CoCalc! Please sign up to avoid losing your work.
Files New Log Find Linux.term
File v Search Pause Copy Paste Help Guide Help ~/Linux_scripts
~/Linux_scripts$ chmod u+x script1.sh
~/Linux_scripts$ ls
script1.sh
~/Linux_scripts$

```

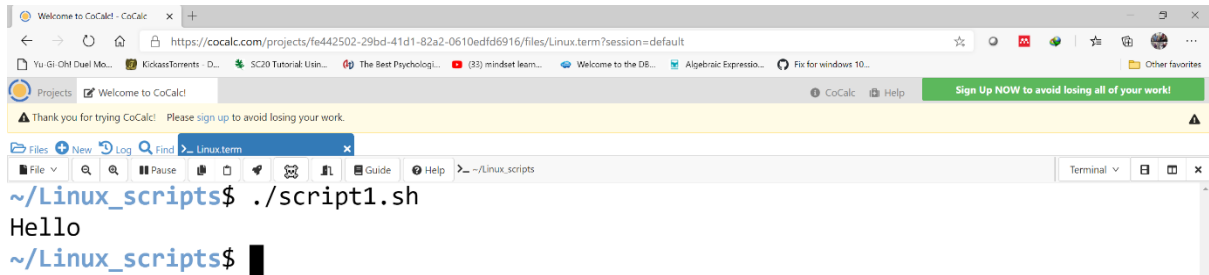
- We modify the file and make it executable for the user (**chmod u+x script1.sh**).

Once this command has been performed on a file it does not need to be done again as the file will now remain executable.

You will notice that the filename changes to green in colour after being modified.

- With the file being executable let us see what happens when we run **script1.sh**:

./script1.sh



The screenshot shows a web browser window with the CoCalc interface. The address bar shows a URL to a CoCalc project. Below the browser window, there is a terminal window titled 'Linux.term'. The terminal shows the command `~/Linux_scripts$ ./script1.sh` being executed, and the output `Hello` is displayed. The terminal prompt is `~/Linux_scripts$`.

- The only thing that shows up is the word “**Hello**”.
- None of the information provided in **script1.sh** before the **echo** statement shows on the screen as these were comments which are only visible when one opens the script in a text editor.
- This brings us to the end of **Bash Scripting Part 1**.
 - At this point you should try to expand on the script above by completing **Exercise 7**.