

SUMMER SCHOOL 2021

Bash Scripting Part 3

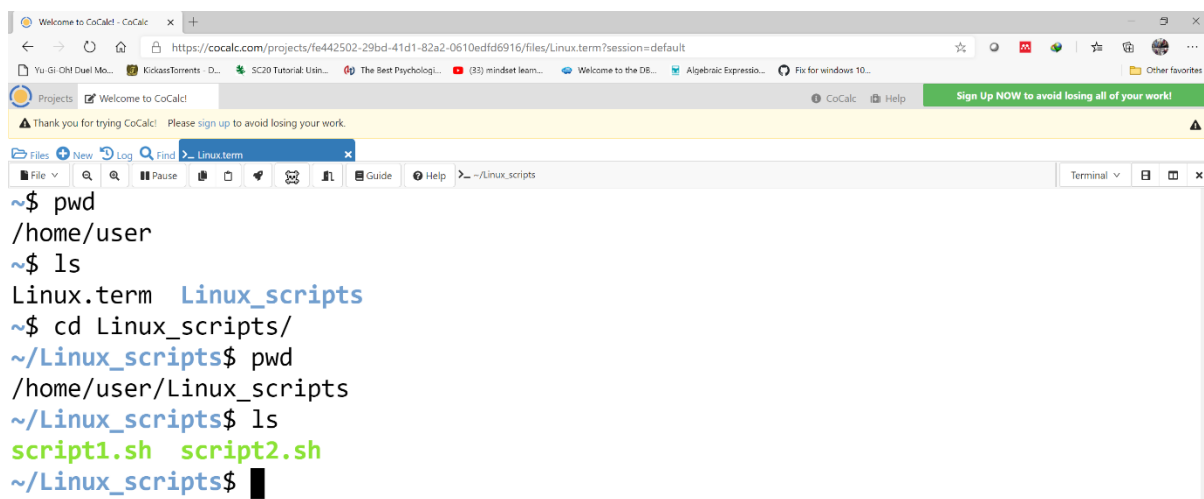
Dr Krishna K. Govender

Summary of commands used in this lecture

Command	Meaning
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
#!/bin/bash	Shebang
echo	Print text to screen
\$var	Print the contents of the variable var
 	Takes the output of one command and passes it as an input to another
wc	Wordcount
for var in XX; do ...; done	Perform an iterative task

- 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.
- If you are in your home directory first make sure you change directory to the Linux_scripts before continuing:

```
pwd
ls
cd Linux_scripts
pwd
ls
```



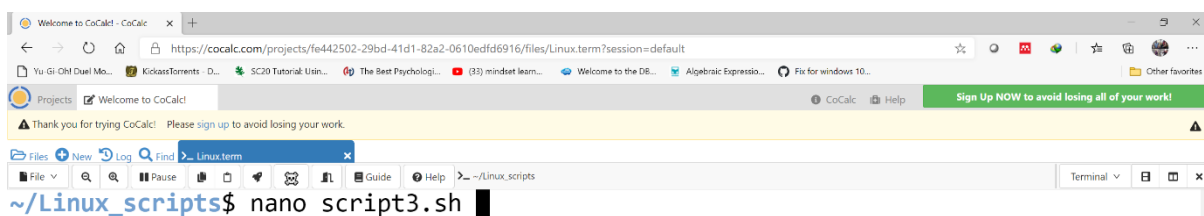
The screenshot shows a web browser window with a CoCalc terminal. The terminal output is as follows:

```
~$ pwd
/home/user
~$ ls
Linux.term  Linux_scripts
~$ cd Linux_scripts/
~/Linux_scripts$ pwd
/home/user/Linux_scripts
~/Linux_scripts$ ls
script1.sh  script2.sh
~/Linux_scripts$
```

- You should see the scripts from the previous two lectures.
- There are several predefined variables contained within Linux and RANDOM is one of them.

We would like to generate a set of 100 random numbers using this variable:

nano script3.sh



The screenshot shows the same CoCalc terminal window. The command to create a new script has been entered:

```
~/Linux_scripts$ nano script3.sh
```



The screenshot shows a web browser window with the CoCalc interface. The terminal window is open, showing the GNU nano 4.8 editor editing a file named script3.sh. The script content is as follows:

```
#!/bin/bash

# Script used to generate a set of 100 random numbers
for numbers in {1..100}
do
    echo $RANDOM
done
```

Below the terminal window, a list of nano editor shortcuts is displayed:

^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

- Even though the variable that I make use of to iterate from 1 to 100 is called numbers, this variable is never used in the loop.

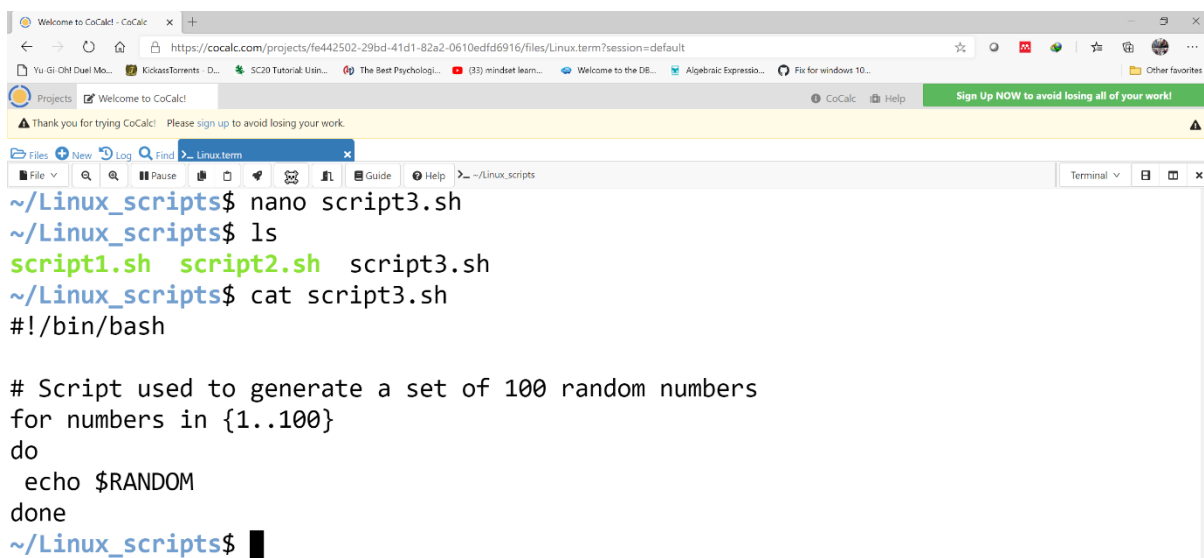
It is merely there as a place holder and the number we wish to obtain at every iteration is a RANDOM number.

- 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 script3.sh



The screenshot shows the same CoCalc web interface, but the terminal window now displays the following commands and their output:

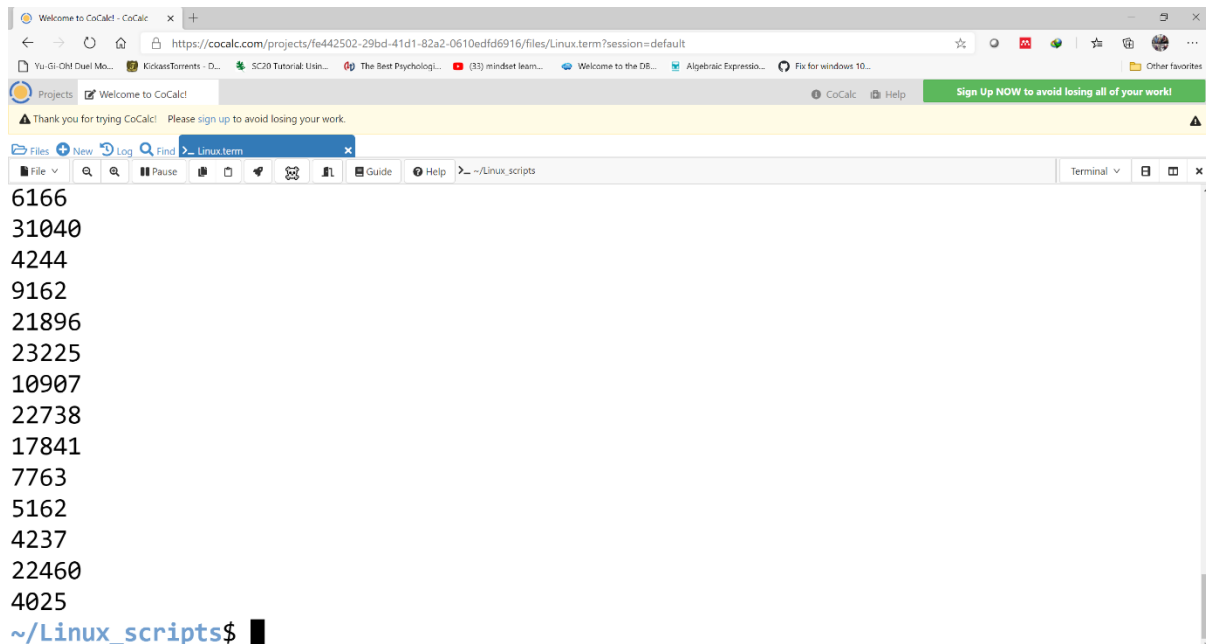
```
~/Linux_scripts$ nano script3.sh
~/Linux_scripts$ ls
script1.sh  script2.sh  script3.sh
~/Linux_scripts$ cat script3.sh
#!/bin/bash

# Script used to generate a set of 100 random numbers
for numbers in {1..100}
do
    echo $RANDOM
done
~/Linux_scripts$
```

- As this is a newly created script, we need to modify it before we can run it:

chmod u+x script3.sh

./script3.sh



The screenshot shows a web browser window with the CoCalc interface. The terminal window is open, displaying the output of the script3.sh script. The output consists of 100 random numbers printed on separate lines. The numbers are: 6166, 31040, 4244, 9162, 21896, 23225, 10907, 22738, 17841, 7763, 5162, 4237, 22460, 4025. The prompt is ~/Linux_scripts\$.

- You should get a bunch of random numbers print to the screen.
- If we want to check that there are in fact only 100 random numbers being printed to the screen we can do this on the terminal:

./script3.sh | wc -l



The screenshot shows the same CoCalc terminal window. The user has entered the command ./script3.sh | wc -l. The output is 100, indicating that 100 lines of output were generated by the script. The prompt is ~/Linux_scripts\$.

- Now instead of getting the random numbers showing on the screen those lines are passed to wordcount and only the total number of lines are printed out verifying that we have 100 random numbers being generated.

- The wordcount can also be done directly in **script3.sh** so when you execute the script all you will see is the number 100 showing you the count for number of random numbers generated.

I am not going to be doing this, but I leave it up to you to see if you can achieve the same result as in the image above without using `| wc -l` on the terminal, but in the script instead.

- This brings us to the end of **Bash Scripting Part 3**.
- At this point you should try to expand on the script above by completing **Exercise 9**.