

SUMMER SCHOOL 2021

Bash Scripting Part 2

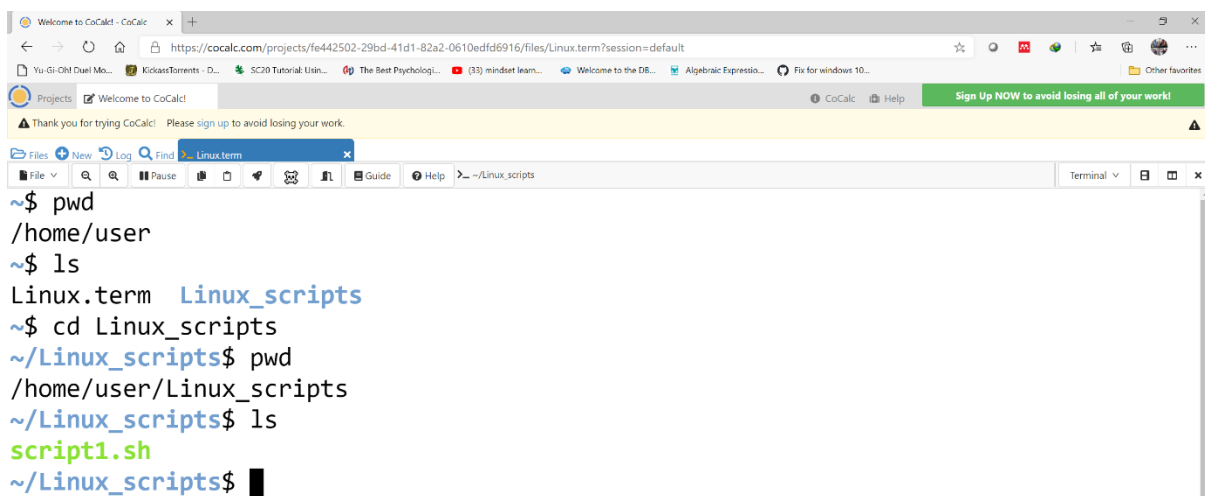
Dr Krishna K. Govender

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
read	Used to read a string/integer input from the terminal
echo	Print text to screen
var=XX	Create a variable called var that stores XX
\$var	Print the contents of the variable var

- 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
```



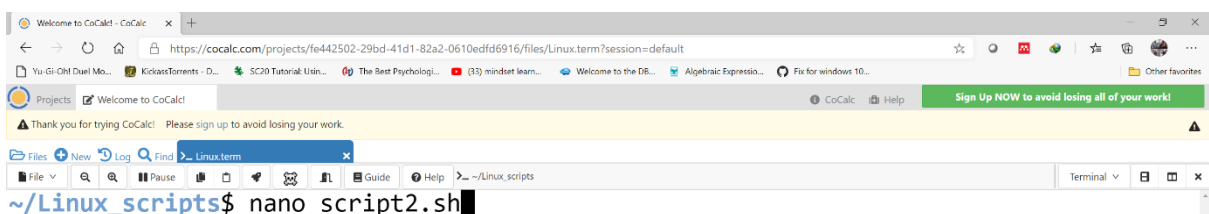
```

Welcome to CoCalc - CoCalc
https://cocalc.com/projects/fe442502-29bd-41d1-82a2-0610edfd6916/files/Linux.term?session=default
Projects Welcome to CoCalc! 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 > ~/Linux_scripts
~$ pwd
/home/user
~$ ls
Linux.term Linux_scripts
~$ cd Linux_scripts
~/Linux_scripts$ pwd
/home/user/Linux_scripts
~/Linux_scripts$ ls
script1.sh
~/Linux_scripts$ █

```

- You should find **script1.sh** inside your directory which was created in the previous lecture.
- You can write scripts that will accept input from the user who is running the script:

nano script2.sh



```

Welcome to CoCalc - CoCalc
https://cocalc.com/projects/fe442502-29bd-41d1-82a2-0610edfd6916/files/Linux.term?session=default
Projects Welcome to CoCalc! 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 > ~/Linux_scripts
~/Linux_scripts$ nano script2.sh █

```



```
GNU nano 4.8 script2.sh
#!/bin/bash

# This script will accept input from the user
# and store the input as a variable

echo 'Please enter your name'
read name
echo 'Hello '$name

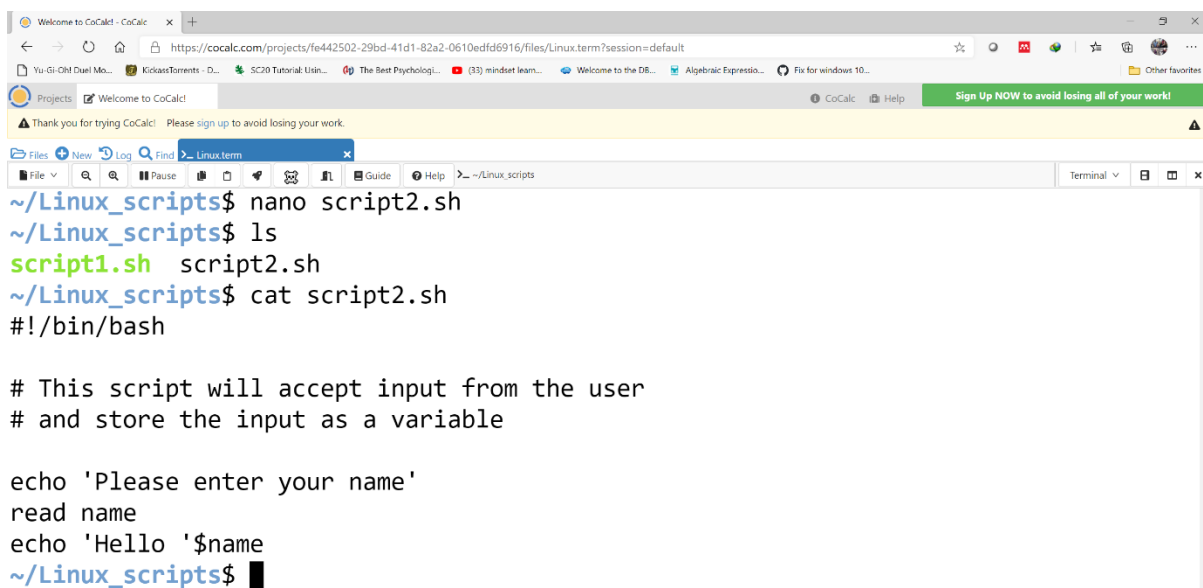
[ Read 8 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify       ^C Cur Pos
^X Exit          ^R Read File     ^\ Replace       ^U Paste Text    ^T To Spell     ^_ Go To Line
```

- When this script is run it should ask the user to enter their name. Once the name is entered it will be stored in a variable called **name**. Finally, a greeting will be printed to the person's name.
- 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 script2.sh



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

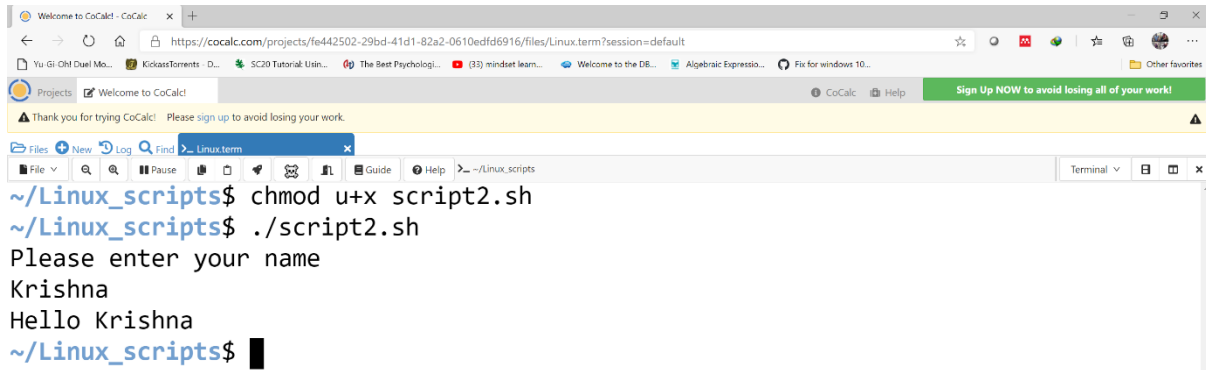
# This script will accept input from the user
# and store the input as a variable

echo 'Please enter your name'
read name
echo 'Hello '$name
~/Linux_scripts$
```

- Since we just created this script, we need to modify it before we can run it:

chmod u+x script2.sh

./script2.sh



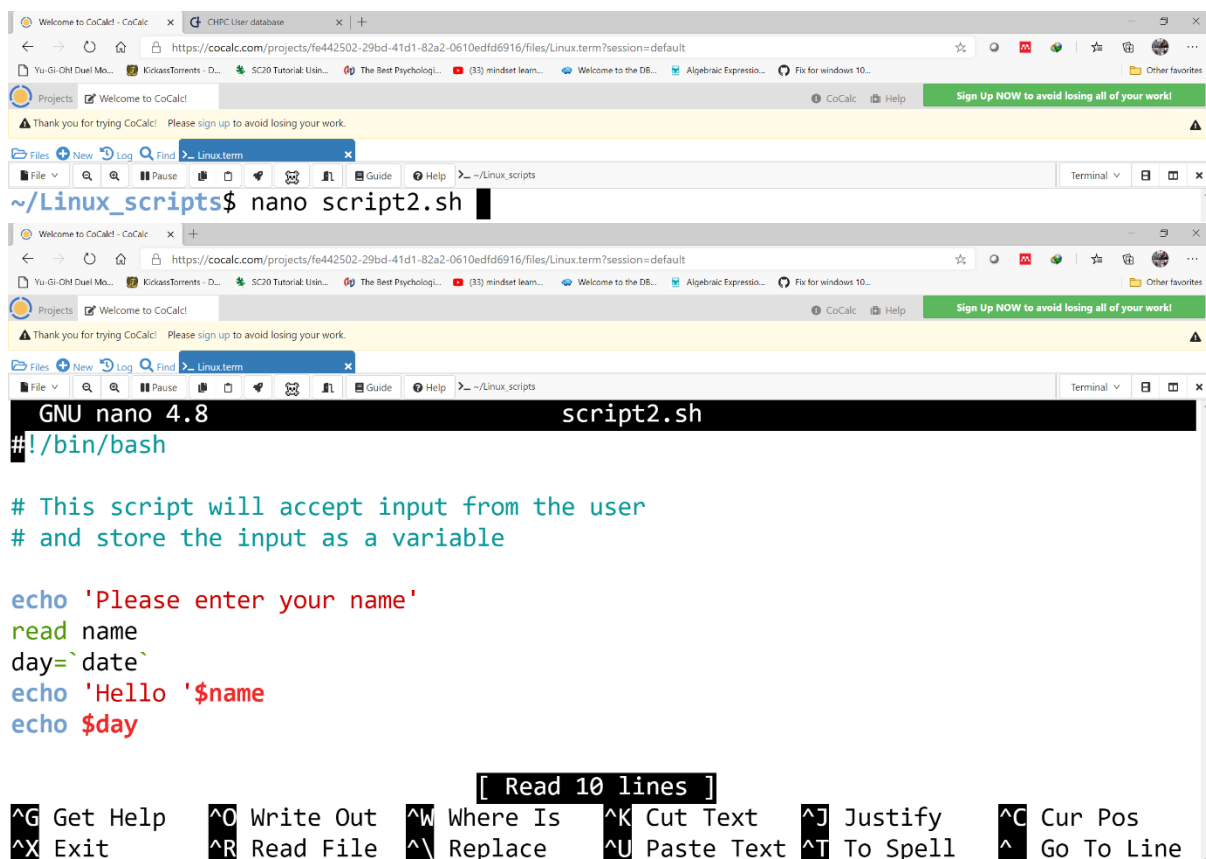
```

~/Linux_scripts$ chmod u+x script2.sh
~/Linux_scripts$ ./script2.sh
Please enter your name
Krishna
Hello Krishna
~/Linux_scripts$ █

```

- Now we will make a variable in our script that will store the date and print this to the screen:

nano script2.sh



```

GNU nano 4.8 script2.sh
#!/bin/bash

# This script will accept input from the user
# and store the input as a variable

echo 'Please enter your name'
read name
day=`date`
echo 'Hello '$name
echo $day

```

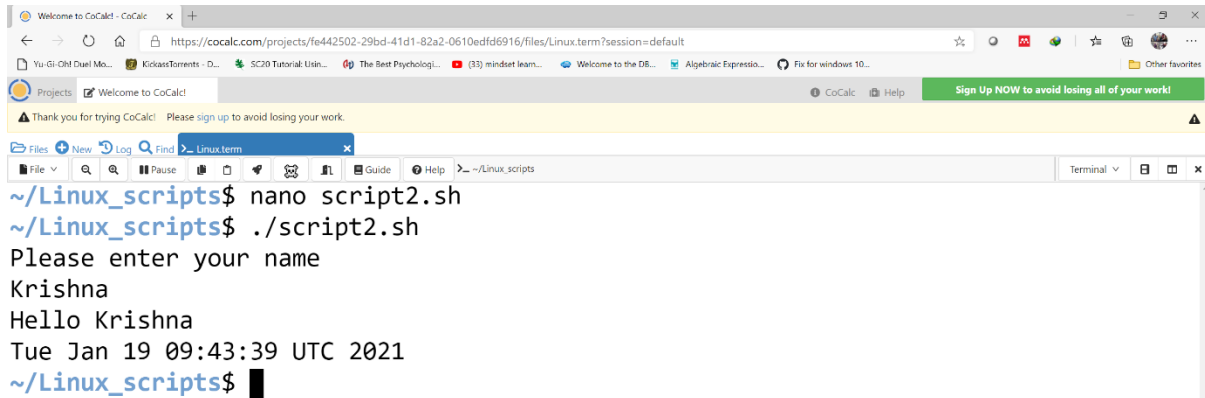
[Read 10 lines]

^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.

- Then run the script again:

`./script2.sh`

A screenshot of a web browser displaying a CoCalc terminal session. The browser's address bar shows a URL from cocalc.com. The terminal window has a title bar that says "Linux.term" and a menu bar with "File", "Edit", "View", "Log", "Find", and "Help". The terminal content shows a user at the prompt ~/Linux_scripts\$ running the command nano script2.sh. After editing, the user runs ./script2.sh. The script prompts for a name, which is entered as "Krishna". The script then prints "Hello Krishna" and the current date and time: "Tue Jan 19 09:43:39 UTC 2021". The prompt returns to ~/Linux_scripts\$.

```
~/Linux_scripts$ nano script2.sh
~/Linux_scripts$ ./script2.sh
Please enter your name
Krishna
Hello Krishna
Tue Jan 19 09:43:39 UTC 2021
~/Linux_scripts$
```

- The first thing that you should take note of is that we defined a new variable called **day** which stores the output produced when you run the command **date** on the terminal. Making use of special ticks (``) on either side of **date** ensure that only the output of the command is stored in the variable.
- Now when we run the script, we find that in addition to printing a greeting to the person's name there is also a date output.
- This brings us to the end of **Bash Scripting Part 2**.
- At this point you should try to expand on the script above by completing **Exercise 8**.