

First job script

Dr Krishna K. Govender

Summary of commands used in this demonstration along with their meanings

Command	Meaning
ssh	Secure Shell login to remote servers
pwd	Present working directory
nano	Linux text editor
vim	Linux text editor
pushd	Push into a directory
popd	Pop back to the directory you originally came from
echo	Print text that appears in quotes to screen
date	Prints out current date and time
sleep	Runs sleep for a certain amount of time

- As with most of the demonstrations thus far you first need to log into the cluster, by typing **ssh username@lengau.chpc.ac.za** (where username is your cluster username).
- Go to your **lustre** directory (**/mnt/lustre/users/username/**) as this is the place where you should be running simulations from
- Make a directory called **demo** and then another directory inside **demo** called **first**:

```
06/07/2020 09:43:41 /home/mobaxterm ssh krishna@lengau.chpc.ac.za
krishna@lengau.chpc.ac.za's password:
Last login: Mon Jul 6 09:33:59 2020 from 146.64.235.118

Welcome to LENGAU

#####
#
# In order to receive notifications via email from the CHPC all users should #
# be subscribed to the CHPC user distribution list. If you are not part of the #
# distribution list you can subscribe at the following link: #
# https://lists.chpc.ac.za/mailman/listinfo/chpc-users #
#
#####
[krishna@login2 ~]$ pwd
/home/krishna
[krishna@login2 ~]$ cd /mnt/lustre/users/krishna/
[krishna@login2 krishna]$ mkdir demo
[krishna@login2 krishna]$ cd demo
[krishna@login2 demo]$ mkdir first
[krishna@login2 demo]$ cd first/
[krishna@login2 first]$ pwd
/mnt/lustre/users/krishna/demo/first
[krishna@login2 first]$
```

- Any script we create from here onwards should be contained in the directory `/mnt/lustre/users/username/demo/first`, so should you get logged out of the cluster at any time just be sure to go back to this directory before doing anything else.
- Since a job script is just text the first thing that you need is to familiarize yourself with the text editors that are available on the cluster. Two popular text editors that are on the cluster are **nano** and **vim**. For simplicity the text editor that I shall use throughout my demonstrations will be **nano**. You are welcome to make use of **vim** should you be comfortable with the editor.
- Open a new file called **submit.pbs** with the following command:

nano submit.pbs

```
[krishna@login2 first]$ nano submit.pbs
```

- You should see a new screen open:

- You are no longer in the Linux command prompt. You are now in the text editor **nano** as you can see from the top of the screen. You will also notice several options provided at the bottom of the screen where ^ represents the **CTRL** key on your keyboard. We

will use one of these options later in this demonstration once we complete our submission script.

- Now enter the following:

#!/bin/bash

#PBS -P YOUR_RESEARCH_PROGRAMME

#PBS -N demo

#PBS -l select=1:ncpus=1

#PBS -l walltime=0:20:00

#PBS -q serial

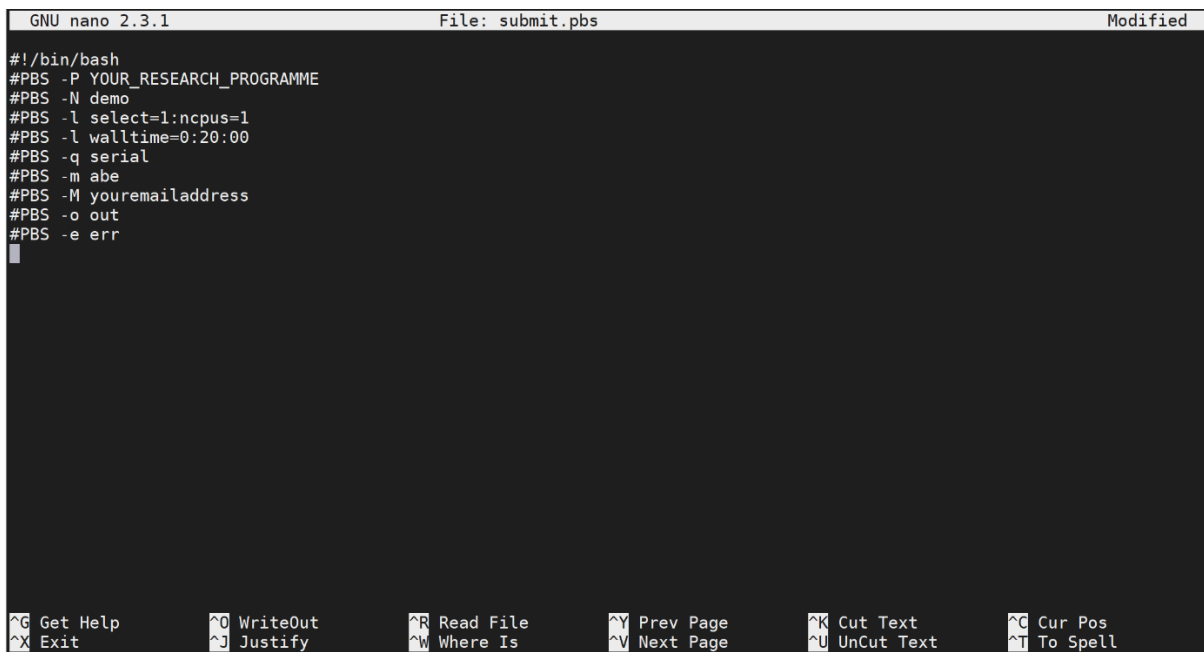
#PBS -m abe

#PBS -M youremailaddress

#PBS -o out

#PBS -e err

- I will explain what each line means once we have completed creating the file
- Once you enter the above you should have something that looks like:

A screenshot of a GNU nano 2.3.1 text editor window. The title bar at the top shows "GNU nano 2.3.1", "File: submit.pbs", and "Modified". The editor area contains the following text:

```
#!/bin/bash
#PBS -P YOUR_RESEARCH_PROGRAMME
#PBS -N demo
#PBS -l select=1:ncpus=1
#PBS -l walltime=0:20:00
#PBS -q serial
#PBS -m abe
#PBS -M youremailaddress
#PBS -o out
#PBS -e err
```

A cursor is visible at the end of the last line. The bottom status bar displays various keyboard shortcuts: ^G Get Help, ^X Exit, ^O WriteOut, ^J Justify, ^R Read File, ^W Where Is, ^Y Prev Page, ^V Next Page, ^K Cut Text, ^U UnCut Text, ^C Cur Pos, and ^T To Spell.

- When you have all the text you need to use **CTRL+x** and you will see something that looks like:

```
GNU nano 2.3.1 File: submit.pbs Modified
#!/bin/bash
#PBS -P YOUR_RESEARCH_PROGRAMME
#PBS -N demo
#PBS -l select=1:ncpus=1
#PBS -l walltime=0:20:00
#PBS -q serial
#PBS -m abe
#PBS -M youremailaddress
#PBS -o out
#PBS -e err

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No ^C Cancel
```

- Then press **Y** and you should find something like:

```
GNU nano 2.3.1 File: submit.pbs Modified
#!/bin/bash
#PBS -P YOUR_RESEARCH_PROGRAMME
#PBS -N demo
#PBS -l select=1:ncpus=1
#PBS -l walltime=0:20:00
#PBS -q serial
#PBS -m abe
#PBS -M youremailaddress
#PBS -o out
#PBS -e err

File Name to Write: submit.pbs
^G Get Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend
```

- Finally, you need to just press **Enter** and you will be taken back to the terminal:

```
[krishna@login2 first]$ nano submit.pbs
[krishna@login2 first]$
```

- To summarize, when you have a document opened in **nano** and you have added text to this document the sequence of keyboard keys used to close **nano** and save your document is: **CTRL+x > y > Enter**
- Now let's look at the commands we entered in the **submit.pbs** file above:

#!/bin/bash

This is used for to let Linux know that this is a bash script, so when it is executed the commands that follow will be the same as those which can be run on the terminal

#PBS -P YOUR_RESEARCH_PROGRAMME

This is a command that tells the PBS scheduler which research programme you belong to. Instead of YOUR_RESEARCH_PROGRAMME you need to put in the appropriate name. At the CHPC research programmes are defined as follows, ASTRXXXX, CBBIXXXX, CHEMXXXX, CSCIXXXX, EARTHXXXX, HEALXXXX, INDYXXXX, MATSXXXX, MECHXXXX, PHYSXXXX, which represent Astrophysics, Bioinformatics, Chemistry, Computer Science, Earth Science, Health Science, Industry, Material Science, Mechanical Engineering and Physics, respectively. The XXXX is merely a 4-digit number that is assigned to your research programme. If you are not sure what your research programme is you can login to (<https://users.chpc.ac.za>) and you will find the programme there.

#PBS -N demo

This is just the name of the job that will be displayed by the scheduler when you check the status of your job. Here I have given the job the name **demo**.

#PBS -l select=1:ncpus=1

This allows you to choose the number of nodes on which to run your job as well as the number of CPUs per node. In this case we just have 1 node and 1 CPU, but the cluster has up to 24 CPUs per node, so ncpus can be anything from 1 to 24.

#PBS -l walltime=0:20:00

Since jobs cannot run forever on the cluster, you need to specify a walltime. This is the time that the job will run on a compute node before terminating. Bear in mind that once

the walltime is reached the job will stop weather it has run to completion or not. For this example, we are using a walltime of 1 hour.

#PBS -q serial

This specifies the queue on which the job should be submitted. There are various different queues available on LENGAU and the appropriate queue needs to be based on your resource requirements, such as number of nodes and number of CPUs per node. The different queues available on the cluster can be found on our wiki page ([http://wiki.chpc.ac.za/quick:start?s\[\]=queues](http://wiki.chpc.ac.za/quick:start?s[]=queues)).

#PBS -m abe

This tells PBS to email you when the job a (aborts), b (begins) and e (ends).

#PBS -M youremailaddress

Here you should provide your email address, which will be emailed when one of the above-mentioned conditions has been met.

#PBS -o out

#PBS -e err

This is the output and error files that are written by PBS. These would typically provide you with additional information pertaining to why a job may have exited abruptly and help with determining what needs to be done in order to get your simulation running.

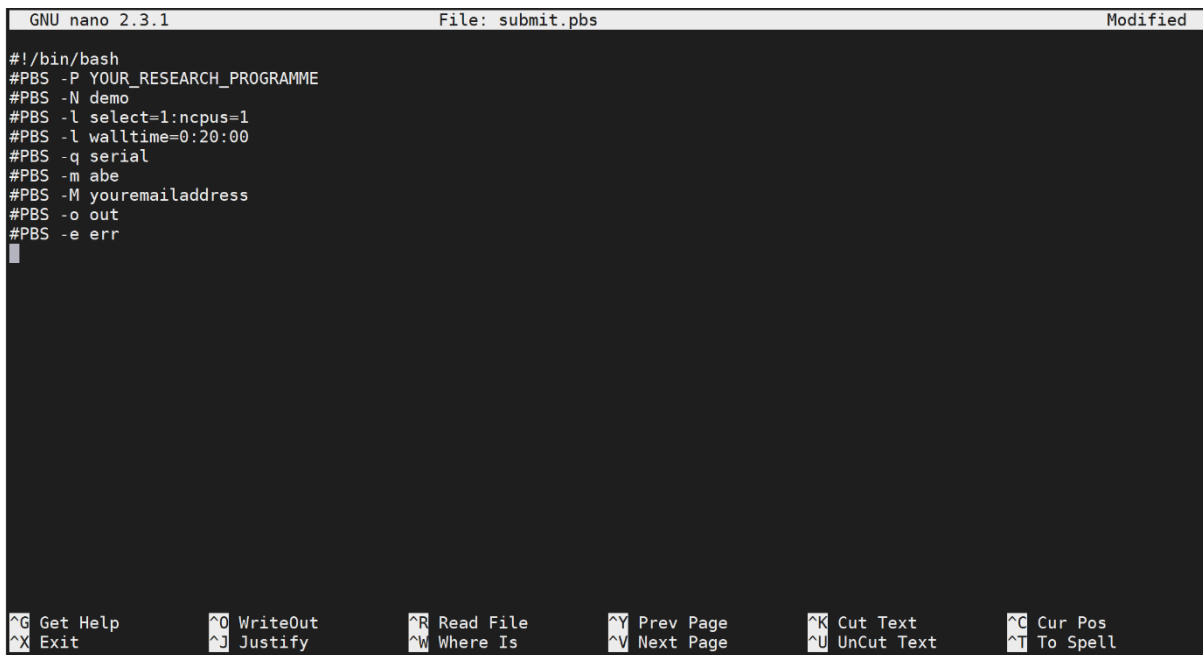
- The script provided above should be used as a templet for any job submissions that you wish to do on the cluster.
- In its current form there is no actual simulation that will be run if this script is submitted. So, we need to add some additional information to the **submit.pbs** file in order to have it produce some output when it is submitted to the cluster.
- Ensure you are in the correct location and open the file with nano once more:

pwd

nano submit.pbs

```
[krishna@login2 first]$ pwd
/mnt/lustre/users/krishna/demo/first
[krishna@login2 first]$ nano submit.pbs
```

- You should see the text that we typed above:



```
GNU nano 2.3.1 File: submit.pbs Modified
#!/bin/bash
#PBS -P YOUR_RESEARCH_PROGRAMME
#PBS -N demo
#PBS -l select=1:ncpus=1
#PBS -l walltime=0:20:00
#PBS -q serial
#PBS -m abe
#PBS -M youremailaddress
#PBS -o out
#PBS -e err
```

- Now put in the following text after the line where you specified the **err** file:

```
pushd /mnt/lustre/users/username/demo/first
echo "This is the date and time when this script first starts"
date
sleep 5m
echo "This is the date and time after 5 minutes"
date
sleep 5m
echo "This is the date and time after 10 minutes"
date
echo "This is the end of the script"
popd
```

- **pushd** will ensure that you are in the correct directory once your job lands on a compute node.

echo will print the text as it appears between the quotes.

date will provide the current date and time.

sleep 5m will run the sleep command for 5 minutes

popd will just take you back to the directory where you first started when you landed on the compute node.

- Your **submit.pbs** file should look like this:

```

GNU nano 2.3.1                               File: submit.pbs
#!/bin/bash
#PBS -P YOUR_RESEARCH_PROGRAMME
#PBS -N demo
#PBS -l select=1:ncpus=1
#PBS -l walltime=0:20:00
#PBS -q serial
#PBS -m abe
#PBS -M youremailaddress
#PBS -o out
#PBS -e err

pushd /mnt/lustre/users/krishna/demo/first
echo "This is the date and time when this script first starts"
date
sleep 5m
echo "This is the date and time after 5 minutes"
date
sleep 5m
echo "This is the date and time after 10 minutes"
date
echo "This is the end of the script"
popd

```

[Read 22 lines]

^G Get Help	^O WriteOut	^R Read File	^Y Prev Page	^K Cut Text	^C Cur Pos
^X Exit	^J Justify	^W Where Is	^V Next Page	^U UnCut Text	^T To Spell

- Once your file has all the information you can save and exit **nano** using: **CTRL+x > y**
> **Enter**
- Now that we have our first script, we will look at how to submit and monitor your job in the next section (Monitoring: qstat, ssh, htop).