BASIC LINUX COMMANDS

Here is a list of commands and softwares we will use during the CROCO training week.

1: Basic Linux commands

Linux command	Description	Linux command example
cd	Change director with a specific path	`cd lustre/croco; cd ..;`
clear	Clear the screen	`clear`
cp	Copy file(s)	`cp path1/file1 path2/file2`
diff	Compare the content of files	`diff file1 file2`
exit	Log out of Linux	`exit`
grep	Find a string of text in a file	`grep "NBPROCS" file1`
head	Display the heading of a file	`head file1`
history	List commands used the terminal session	`history`
ln	Create a symbolic link	`ln -sf /mnt/lustre lustre`
ls	List contents of a directory	`ls path1/directory1`
meld	Compare the content of files	`meld file1 file2 file3`
mkdir	Create a directory	`mkdir directory`
mv	Move file(s) or rename file(s)	`mv path1/file1 path2/file2`
rm	Delete file(s)	`rm file1`
rmdir	Remove an empty diectory	`rmdir directory`
tail	Display the end of a file	`tail file1`
tar	Store, list, or extract file in an archive	`tar file1`
scp	Copy file(s) from/to a distant machine	
ssh	Connect to a distant machine	`ssh login@lengau.chpc.ac.za`
vi	Edit file(s) with simple text editor	`vi file1`

2: Useful softwares and their execution commands

→ Linux text editor: nedit (or gedit, emacs…)
  ➥ Execute **`nedit file1`**

→ To pre- and post- process CROCO files: Matlab (*https://mathworks.com*)
  ➥ Execute **`matlab -nodesktop`** (or the alias **`mat`**)

→ To display the header of a NetCDF file: NetCDF library (*https://www.unidata.ucar.edu/software/netcdf*)
  ➥ Execute **`ncdump -h my_NetCDF_file.nc`**

→ To inspect a NetCDF file: ncview (*https://cirrus.ucsd.edu/ncview*)
  ➥ Execute **`ncview my_NetCDF_file.nc`**

→ To manipulate NetCDF files: NCO tools (*https://nco.sourceforge.net*)
  ➥ Concatenate files: **`ncrcat CROCO_avg*.nc My_CROCO_file.nc`**
  ➥ Extract a variable: **`ncks -v myvar My_CROCO_file.nc My_CROCO_myvar.nc`**

AN INTRODUCTION TO THE VI EDITOR

It's easy to invoke **vi**.
➥ At the command line, you type `vi <filename>` to create a new file, or to edit an existing one.

```
[login@cnode0220 ~]$ vi filename.txt
[login@cnode0220 ~]$
```

NODES

1: vi modes: Command and Insert

→ The **vi** editor has **two modes**: `Command` and `Insert`.

➥ When you first open a file with **vi**, you are in `Command mode`. Command mode means you can use keyboard keys to navigate (using the up ⮝ and down ⮟ arrows or the horizontal arrows ⮜⮞), delete, copy, paste, and do a number of other tasks — except entering text.

➥ To enter `Insert mode`, press **i**. In Insert mode, you can enter text, use the Enter key to go to a new line, use the arrow keys (up ⮝, down ⮟, left ⮜, and right ⮞ arrows) to navigate text, and use vi as a free-form text editor. To return to Command mode, press the Esc key once.

→ In **vi** 's `Command mode`, almost every letter on the keyboard has a function.
➥ To save a file, you must first be in Command mode. Press Esc to enter `Command mode`, and then type `:wq` to write and quit the file. The other, quicker option is to use the keyboard shortcut `ZZ` to write and quit. In **vi**, write means save, and quit means exit. If you've made mistakes along the way while editing and want to back out (abandon) all non-saved changes, enter `Command mode` by pressing Esc and typing `:q!` This command quits without saving any changes and exits **vi**.

✋ Always make a copy of an existing file prior to editing with **vi** or any editor. This is especially critical when editing system and configuration files.

2: Some vi shortcuts

→ The best way to learn **vi** is to create a new file and try it out for yourself. Feel free to use the common keyboard shortcut list below to help you learn **vi**'s extensive vocabulary. This list of shortcuts is by no means exhaustive, but they will enable you to edit files and learn **vi** in a short amount of time.

`i` : Switch to Insert mode.
Esc : Switch back to Command mode.

`:w` : Save and continue editing.
`:wq` or `ZZ` : Save and quit/exit **vi**.
`:q!` : Quit vi and do not save changes.
`/bgtdf` : Seach the "bgtdf" string
`n` for next appearance, `esc` to quit search
`yy` : Yank (copy) a line of text.
`p` : Paste a line of yanked text below the current line
`o` : Open a new line under the current line.
`O` : Open a new line above the current line.
`A` : Append to the end of the line.
`a` : Append after the cursor's current position.
`I` : Insert text at the beginning of the current line.

`b` : Go to the beginning of the word.
`e` : Go to the end of the word.
`x` : Delete a single character.
`dd` : Delete an entire line.
`Xdd` : Delete X number of lines.
`Xyy` : Yank X number of lines.
`G` : Go to the last line in a file.
`XG` : Go to line X in a file.
`gg` : Go to the first line in a file.
`:num` : Display the current line's number.
`h` : Move left one character.
`j` : Move down one line.
`k` : Move up one line.
`l` : Move right one character.

LENGAU CLUSTER AT CHPC

The main system at the CHPC for high performance computing is a cluster supercomputer called **Lengau** — *cheetah* in seTswana. This petascale system consists of Dell servers, powered by Intel processors, using FDR InfiniBand by Mellanox, and is managed by the Bright Cluster Manager.

1: The Cluster

→ The CHPC's Dell Linux cluster has been up and running since 2014.

➥ The system is a homogeneous cluster, comprising Intel 5th generation CPUs. As of March 2017, it has 1368 compute nodes, each with 24 cores and 128 GiB of memory (360 nodes have only 64 GiB), and five large memory "fat" nodes with 56 cores and 1TiB each, all interconnected by FDR 56 Gb/s InfiniBand and accessing 4 PB of shared storage via the Lustre file system.

→ The cluster has both NFS and Lustre filesystems over Infiniband:

Mount Point	File System	Size	Quota	Backup
/home	NFS	80Tb	15GB	NO
/mnt/lustre/users	lustre	4PB	None ⚠	NO

2: The installed softwares/libraries

→ CHPC uses the GNU modules utility, which manipulates your environment, to provide access to the supported software in /apps/.

➥ We will load the following modules directly in your `.bashrc`, to use the associated softwares/libraries: Matlab, ncview, Intel Fortran Compilers, NCO tools, meld command, etc…

Command	Description
module purge	Remove all loaded modules
module avail	List of available modules
module list	List currently loaded modules
module help	Give information of a particular module file's operations

3: Job Scheduler

→ The CHPC cluster uses **PBSPro** as its job scheduler. With the exception of interactive jobs, all jobs are submitted to a batch queuing system and are executed only when the requested resources become available. All batch jobs are queued according to priority. A user's priority is not static: the CHPC uses the "Fairshare" facility of PBSPro to modify priority based on activity. This is done to ensure that the finite resources of the CHPC cluster are shared fairly among all users.

PBS Pro commands	Description	PBS command example
qsub [script file]	Job submission	`qsub run_croco.pbs`
qstat -u login	Job status (for user)	`qstat -u sillig` (or the alias `qs`)
qstat -f [job_id]	Extended job status	`qstat -f 10098976`
qdel-x [job_id]	Job deletion	`qdel -x 10098976`
qstat-Q	List of usable queues	

→ The available queues with their nominal parameters are given in the following table:

Queue Name	Max. cores	Min. cores	Max. jobs		Max. time	Notes	Access
	per job		in queue	running	hrs		
serial	23	1	24	10	48	For single-node non-parallel jobs.	
seriallong	12	1	24	10	144	For very long sub 1-node jobs.	
smp	24	24	20	10	96	For single-node parallel jobs.	
normal	**240**	**25**	**20**	**10**	**48**	**The standard queue for parallel jobs**	
large	2400	264	10	5	96	For large parallel runs	*Restricted*
xlarge	6000	2424	2	1	96	For extra-large parallel runs	*Restricted*
express	2400	25	N/A	100 total nodes	96	For paid commercial use only	*Restricted*
bigmem	280	28	4	1	48	For the large memory (1TiB RAM) nodes.	*Restricted*
vis	12	1	1	1	3	Visualisation node	
test	24	1	1	1	3	Normal nodes, for testing only	
gpu_1	10	1		2	12	Up to 10 cpus, 1 GPU	
gpu_2	20	1		2	12	Up to 20 cpus, 2 GPUs	
gpu_3	36	1		2	12	Up to 36 cpus, 3 GPUs	
gpu_4	40	1		2	12	Up to 40 cpus, 4 GPUs	
gpu_long	20	1		1	24	Up to 20 cpus, 1 or 2 GPUs	*Restricted*

+ A temporary queue (named `R6060705`) with 3 nodes dedicated to our CROCO Summer's school.

4: Supplementary material

→ Some useful contents are available on CHPC website:

- Video demo of interactive sessions: https://www.youtube.com/watch?v=FZFwgSVE0HY
- Video first job script: https://www.youtube.com/watch?v=cuxj0oWvQzA