Essential Linux and Slurm

T. Chantavat and P. Boonaom

The Linux Terminal

- **echo $0** : is used to display the filename of the script that is currently being executed.
- **whoami** : shows the currently logged-in user.
- **hostname** : is used ti obtain the DNS (Domain Name System) name and set the system's hostname or NIS (Network Information System) domain name.

From: https://itsfoss.com/basic-terminal-tips-ubuntu/

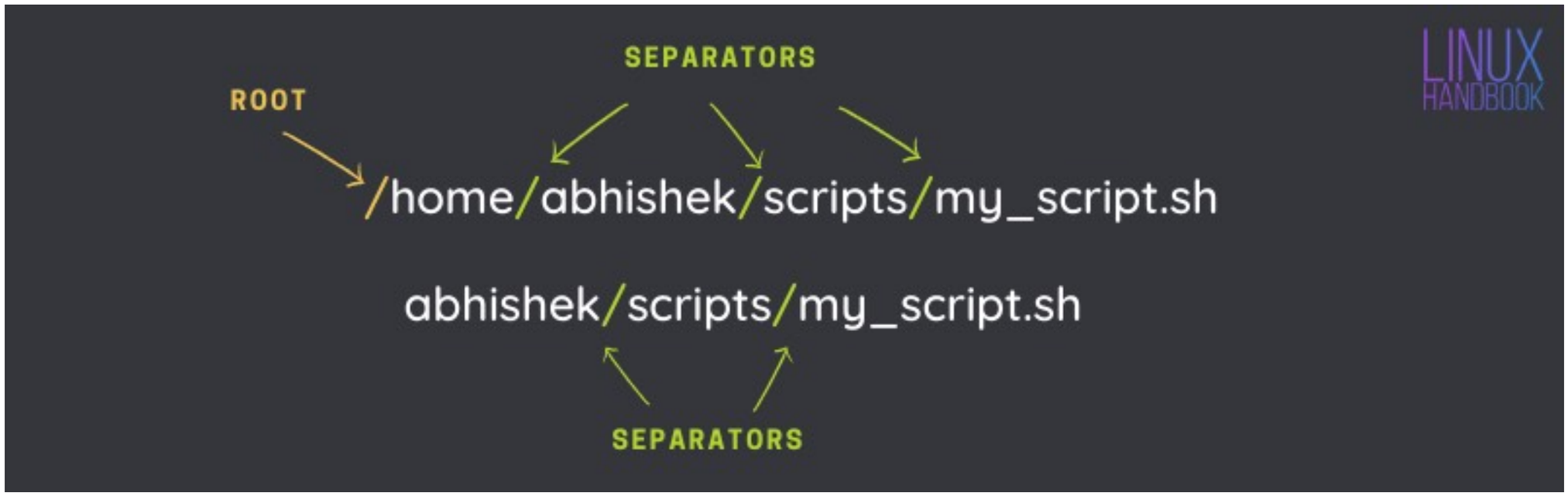
Commands

```
[username@scc1 ~]$ command --option argument
```

- **Command**: Command/program that does one thing
- **Options**: Change the way a command does that one thing
  - Short form: Single-dash and one letter e.g. **`ls -a`**
  - Long form: Double-dash and a word e.g. **`ls --all`**
- **Argument**: Provides the input/output that the command interacts with.

From: https://itsfoss.com/basic-terminal-tips-ubuntu/

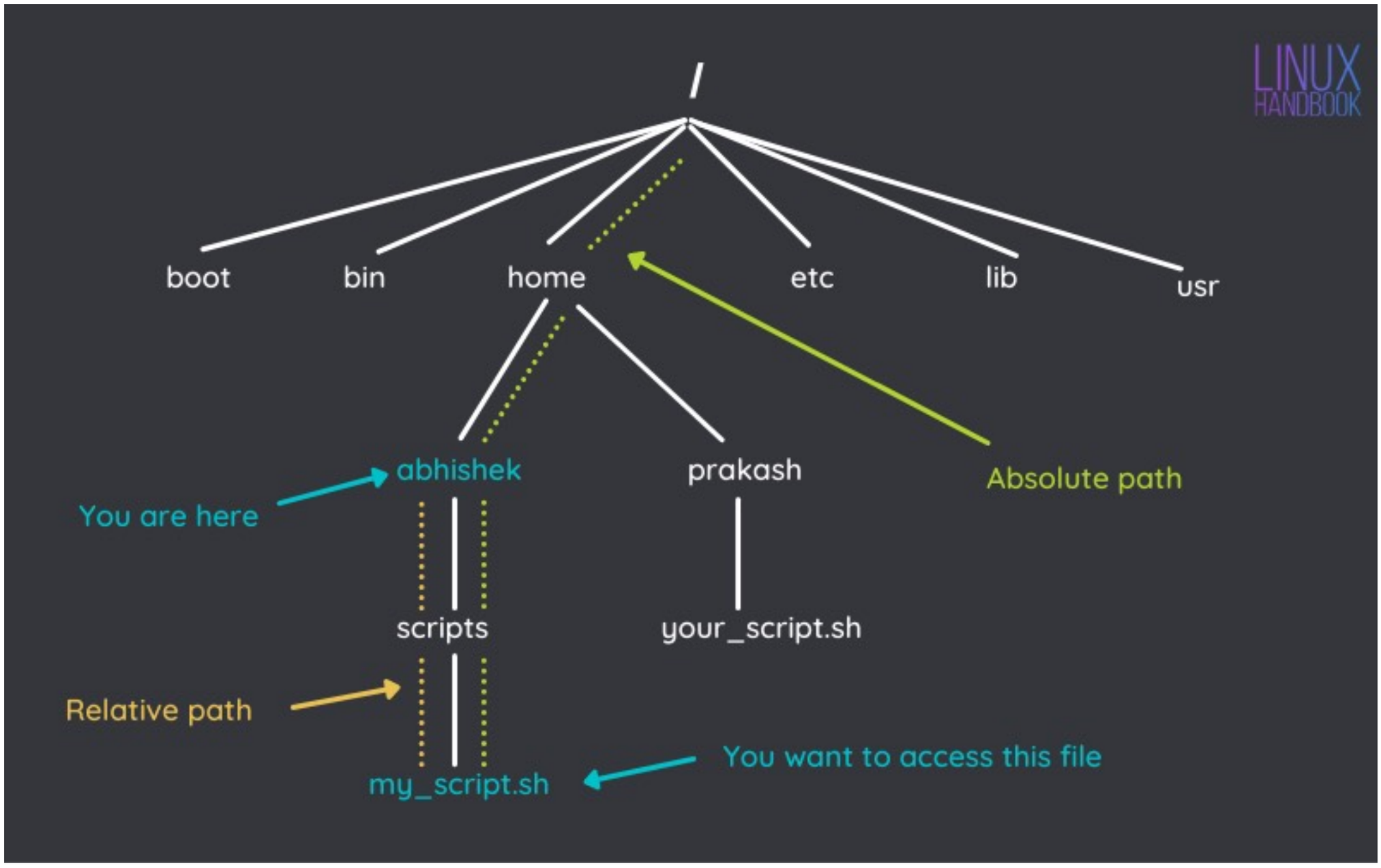
PATH



- ~ : is a shorthand for home directory

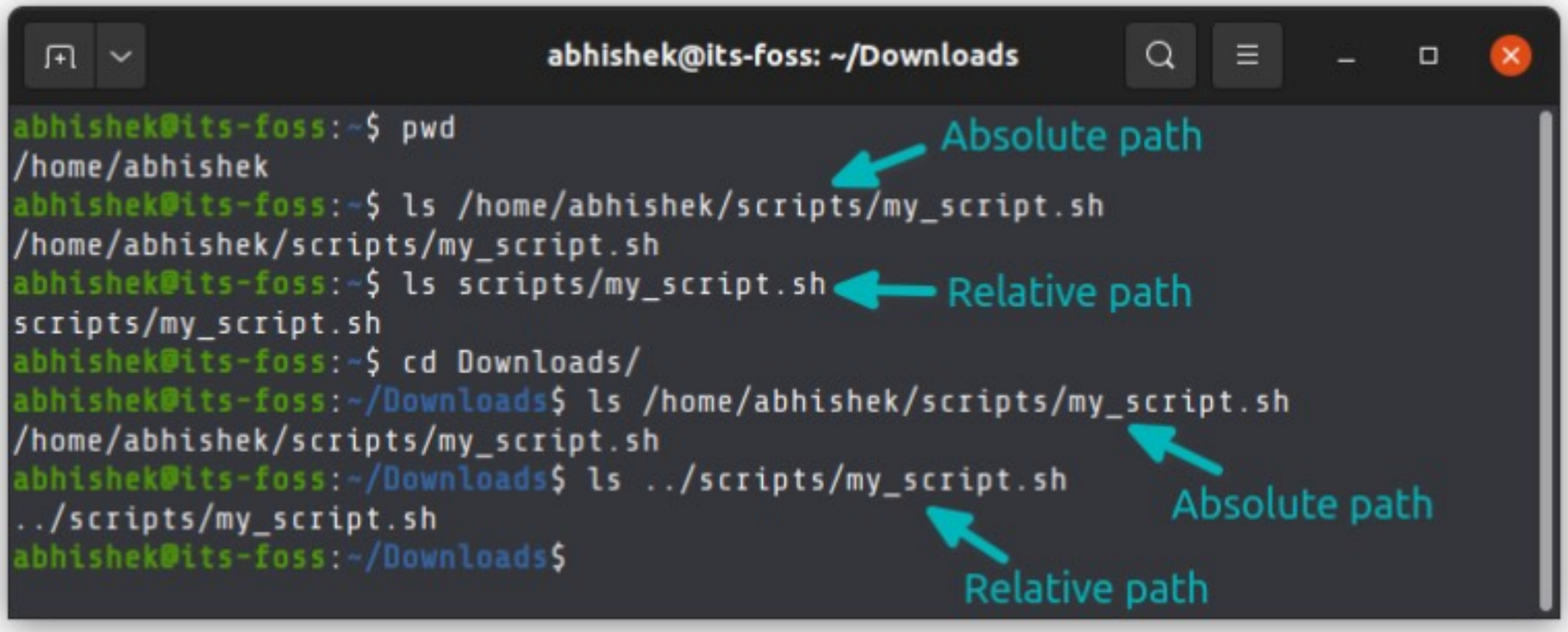
From: https://itsfoss.com/basic-terminal-tips-ubuntu/

PATH



From: https://itsfoss.com/basic-terminal-tips-ubuntu/

PATH



- . : current directory

- .. : parent directory

From: https://itsfoss.com/basic-terminal-tips-ubuntu/

man [command]

- Display the help information for the specified command.

Examples : man ls

apropos [keyword]

- Search the database for strings in commands.

Examples : apropos directory

ls [options] [names]

- List directory contents

Examples : ls -l, ls -a

cp [options] file1 file2

cp [options] file directory

- Copy files and directories. **Beware!!!** `cp` can delete your file permanently by replacing an existing file.

Examples : cp test1.txt test2.txt, cp -rf dir1 dir2

mv [options] source target

- Rename or move file(s) or directories. **Beware!!!** `mv` can also delete your file permanently by replacing an existing file.

Examples : mv file1 file2, mv myfile.txt /usr/backup

rm [options] [file | directory]

- Remove (delete) file(s) and/or diectories. **Beware!!!** `rm` can also delete your file permanently by replacing an existing file.

Examples : rm file1, rm -rf mydir/

ln [options] existing newname

- Creat a shortcut

Examples : ln -s /mnt/exthdd external_hdd

cd [directory]

- Change directory

Examples : cd /home, cd ../../../Downloads

pwd

- Display the pathname for the current directory

Examples : pwd -P (show Physical path)

mkdir [options] directories

- Create a new directory

Examples : mkdir test

ln [options] existing newname

- Creat a shortcut

Examples : ln -s /mnt/exthdd external_hdd

cd [directory]

- Change directory

Examples : cd /home, cd ../../../Downloads

cd (go to /home directory), cd - (go to previous directory)

cat [options] [files]

- Display file’s contents to the standard output device

Examples : cat data.txt

less [directory] [files]

- View the contents of a file one page at a time

Examples : less data.txt

head [options] [files]

- Display the first n lines of a file (the default is 10)

Examples : head -20 data.txt

tail [directory] [files]

- Display the last n lines of a file (the default is 10)

Examples : tail -15 data.txt

touch [options] files

- Create an empty file with the specific name

Examples :touch newfile

clear

- Clear a command line screen/window for a fresh start

Examples : clear

chmod [options] [mode] [files]

- Change a file's permission

Examples : chmod -R 777 forbidden_file

ls -l

rwx
read write execute

Execute let's you execute executables

4 2 1 4 2 1 4 2 1

\- rwx rwx rwx

user group other

7 7 7

Permissions (mode)	Number
rwx	7 (4+2+1)
rw-	6 (4+2)
r-x	5 (4+1)
r--	4
-wx	3 (2+1)
-w-	2
--x	1
---	0

chown (change ownership) - ex chown root file1

chmod (change mode [permissions]) ex chmod 540 file1

From: https://www.freekb.net/Article?id=1275

find [pathnames] [conditions]

- Search files and directories

Examples : find /home/John -name ‘*.gz’

which [options] [commands]

- Display the path to the command specified

Examples : which python3

grep [options] pattern [files]

- Search files or output for a particular pattern

Examples : grep -inR ‘Cosmology’

echo [options] [string]

- Display the output

Examples : echo $PATH

gzip, gunzip [options] [files]

- Compress and extract files

Examples : gzip my_file

tar [options] [tarfile] [other-files]

- Manipulate tape archives

Examples : tar -xzvf hello.tar.gz, tar -czvf bye.tar.gz bye1 bye2

df [options] [name]

- Display used and available disk space

Examples : df -h

du [options] [file]

- Show how much space each file takes up

Examples : du -h

ssh [options] hostname [command]

- Remotely log in to another Linux machine, over the network. Leave an ssh session by typing **exit**

Examples : ssh pboonaom@lanta.nstda.or.th

scp [options] hostname path

- Remotely copy file to another Linux machine, over the network.

Examples : scp pboonaom@lanta.nstda.or.th:/home/file.txt /home/Downloads/

File editor

- **gedit**
  - Notepad-like editor with some programming features (e.g., syntax highlighting).
- **nano**
  - Lightweight editor.
- **emacs**
  - Swiss-army knife, has modes for all major languages, and can be customized. Formerly steep learning curve has been reduced with introduction of menu and tool bars.
- **vim**
  - A better version of ‘vi’ (an early full-screen editor). Very fast, efficient. Steep learning curve. Popular among systems programmers.

Basic Slurm In Use

PB (Pongsapat Boonaom)

Slurm workload manager

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. Slurm requires no kernel modifications for its operation and is relatively self-contained. As a cluster workload manager, Slurm has three key functions. First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work. Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes. Finally, it arbitrates contention for resources by managing a queue of pending work.

sinfo

- reports the state of partitions and nodes managed by Slurm. It has a wide variety of filtering, sorting, and formatting options.

```
[pongsapat@pollux ~]$ sinfo
HOSTNAMES PARTITION      AVAIL CPUS(A/I/O/T)  CPU_LOAD  ALLOCMEM  FREE_MEM  GRES    STATE   TIMELIMIT
pollux2   chalawan_gpu   up    16/12/0/28     4.05      32768     244310    gpu:4   mix     infinite
pollux3   chalawan_gpu   up    16/12/0/28     4.03      32768     244181    gpu:4   mix     infinite
pollux1   chalawan_gpu   up    0/24/0/24      0.02      0         125889    gpu:4   idle    infinite
castor12  chalawan_cpu*  up    0/0/28/28      27.72     0         1605      (null)  down*   infinite
castor2   chalawan_cpu*  up    8/8/0/16       8.02      40000     42369     (null)  mix     infinite
castor5   chalawan_cpu*  up    12/16/0/28     12.04     24576     113540    (null)  mix     infinite
castor6   chalawan_cpu*  up    13/15/0/28     12.98     29576     100103    (null)  mix     infinite
castor11  chalawan_cpu*  up    16/12/0/28     15.98     44576     62906     (null)  mix     infinite
castor14  chalawan_cpu*  up    27/1/0/28      27.04     64152     89946     (null)  mix     infinite
castor15  chalawan_cpu*  up    27/1/0/28      27.04     64152     42342     (null)  mix     infinite
castor17  chalawan_cpu*  up    24/4/0/28      20.13     49152     99556     (null)  mix     infinite
castor19  chalawan_cpu*  up    12/16/0/28     12.00     24576     114186    (null)  mix     infinite
castor4   chalawan_cpu*  up    28/0/0/28      28.03     57344     110602    (null)  alloc   infinite
castor7   chalawan_cpu*  up    28/0/0/28      28.05     69152     54243     (null)  alloc   infinite
castor8   chalawan_cpu*  up    28/0/0/28      28.01     57344     109701    (null)  alloc   infinite
castor9   chalawan_cpu*  up    28/0/0/28      28.01     57344     110750    (null)  alloc   infinite
castor13  chalawan_cpu*  up    28/0/0/28      12.33     120000    117815    (null)  alloc   infinite
castor16  chalawan_cpu*  up    28/0/0/28      28.01     57344     111991    (null)  alloc   infinite
castor18  chalawan_cpu*  up    28/0/0/28      28.01     57344     110731    (null)  alloc   infinite
castor3   chalawan_cpu*  up    0/16/0/16      0.01      0         56576     (null)  idle    infinite
castor10  chalawan_cpu*  up    0/0/28/28      0.02      0         116591    (null)  down    infinite
viser03   viser          up    0/0/32/32      0.01      0         1049      (null)  down    infinite
```

sbatch [options] executable [arguments]

- is used to submit a job script for later execution. The script will typically contain one or more srun commands to launch parallel tasks.

scancel [job id]

- is used to cancel a pending or running job or job step. It can also be used to send an arbitrary signal to all processes associated with a running job or job step.

squeue

- reports the state of jobs or job steps. It has a wide variety of filtering, sorting, and formatting options. By default, it reports the running jobs in priority order and then the pending jobs in priority order.

```
[pongsapat@pollux ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            519288 chalawan_ 5-1H-2_V nuttacha PD       0:00      1 (Resources)
            506081 chalawan_    c3-sa  michael  R 15-10:04:56      1 castor8
            513307 chalawan_  q11N10M  worraki  R 2-11:01:25      1 castor11
            518186 chalawan_ q09_5N10  worraki  R 1-02:59:30      1 castor6
            518185 chalawan_ q08N10M_  worraki  R 1-03:07:40      1 castor5
            518184 chalawan_ q06_5N10  worraki  R 1-03:07:51      1 castor19
            518187 chalawan_ q12N10M_  worraki  R 1-02:40:41      1 castor14
            518188 chalawan_ q13_5N10  worraki  R 1-02:40:41      1 castor14
```

```
[pongsapat@pollux ~]$ squeue -u pongsapat
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            518205 chalawan_  CosmoMC pongsapa  R    4:34:49      1 pollux3
            518197 chalawan_  CosmoMC pongsapa  R   12:43:13      1 pollux2
```

Examples of batch scripts

```
#!/bin/bash

#SBATCH -J task1                    # Job name
#SBATCH -t 00:01:00                 # Run time (hh:mm:ss)

echo "Hello World!"
```

```
#!/bin/bash

#SBATCH -J R3_0.002       # Job name
#SBATCH -p chalawan_cpu # Partition
#SBATCH -n 4 -c 4             # Number of task

module purge
module load gnu8
module load hwloc
module load openmpi3
source data/clik_14.0/bin/clik_profile.sh

mpirun -np 4 -c 4  ./cosmomc inifiles/R3_0.002/plc-bao.ini
```

Frequently used sbatch options

Options	Descriptions
-J , —job-name=<name>	name of job
-N , —nodes=<N>	number of nodes on which to run (N = min[-max])
-n<count>	number of tasks to run
-c , —cpus-per-	number of cpus required per task
-e , —error=<err>	file for batch script’s standard error
-o , —output=<output>	file for batch script’s standard output
-p , —	partition requested
-t , —time=<minutes>	time limit
—mem=<MB>	minimum amount of real memory
—gres=<list>	required generic resources