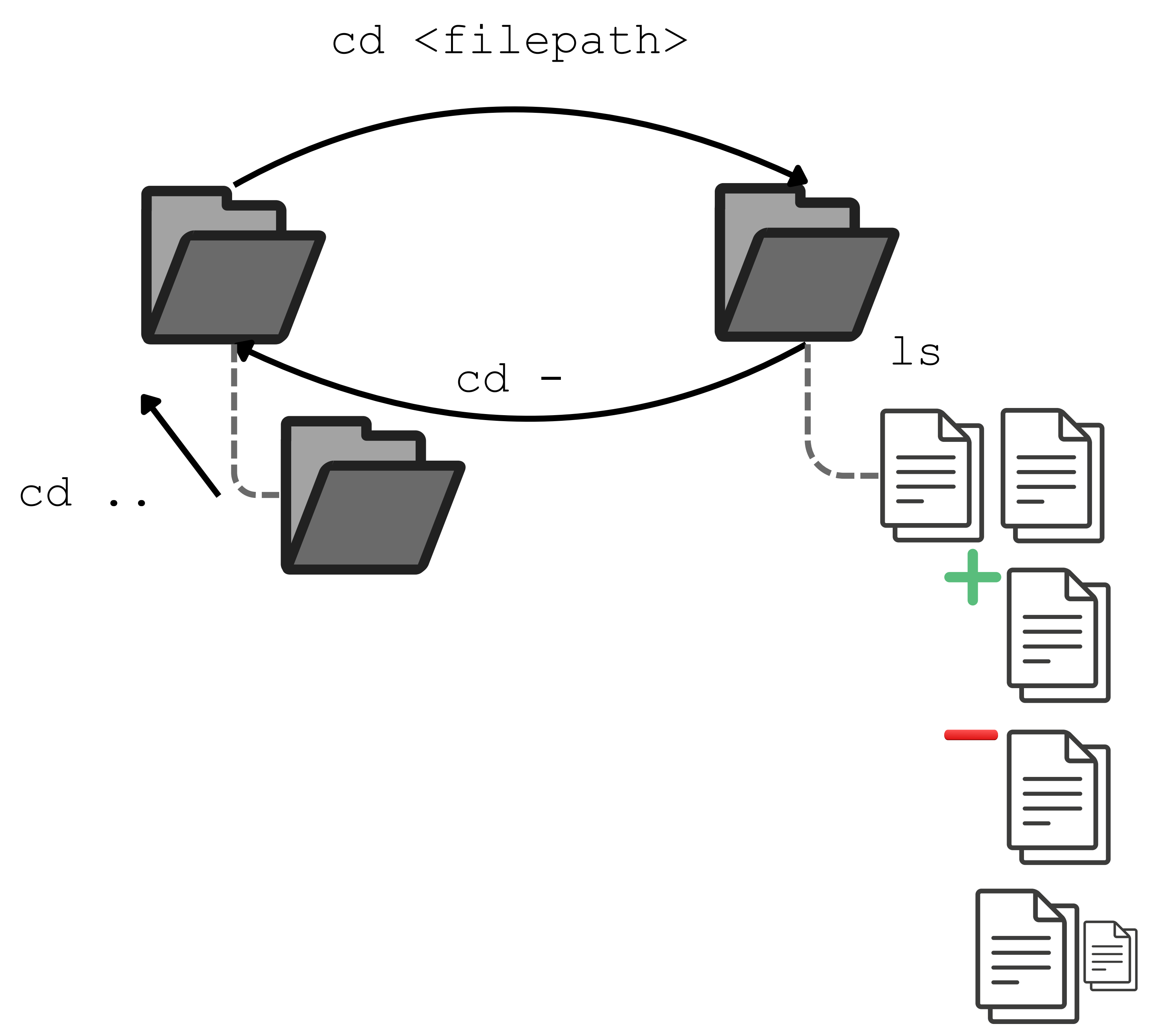
Bash in_1_page



Navigation and file manipulation

`touch <file>`   `mkdir` (📁)

`rm <file>`   `-r` (📁)

`cp <file> <new file>`   `-r` (📁)

Globbing - str matching

`ls *.txt` — list all files with .txt in their name
'*' means any string

`ls new*.txt` — list any files that start with 'new' and end with '.txt'

Also works with:

`cp, mv, cat, grep, find`

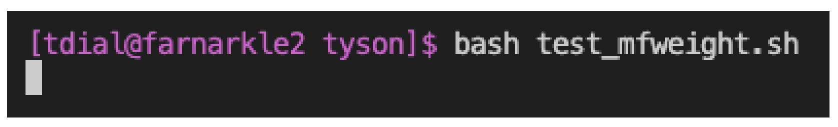
Useful Bash shortcuts

Tab ⇆

press + `ls new` = `ls new_file.txt`

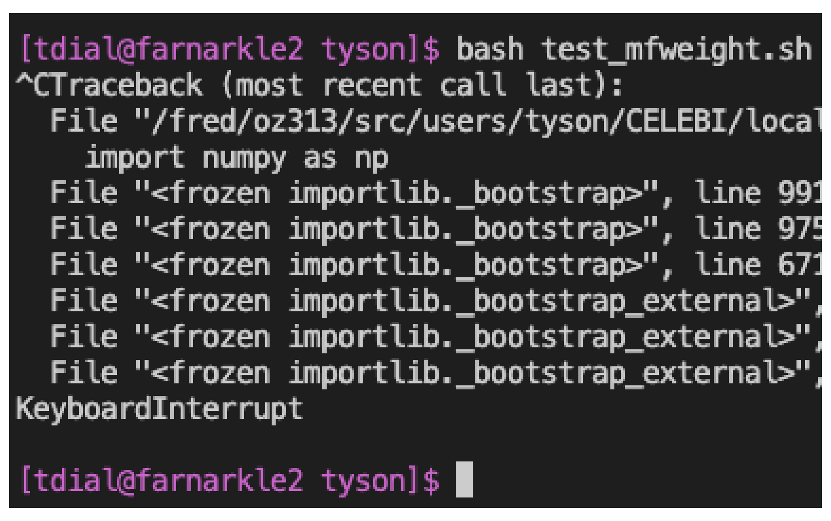
hold + `ls new` = `ls new_file*`

Running a process? KILL IT!



```
[tdial@farnarkle2 tyson]$ bash test_mfweight.sh
```

+ Ctrl + C =



```
[tdial@farnarkle2 tyson]$ bash test_mfweight.sh
^CTraceback (most recent call last):
  File "/fred/oz313/src/users/tyson/CELEBI/local
    import numpy as np
  File "<frozen importlib._bootstrap>", line 991
  File "<frozen importlib._bootstrap>", line 975
  File "<frozen importlib._bootstrap>", line 671
  File "<frozen importlib._bootstrap_external>",
  File "<frozen importlib._bootstrap_external>",
  File "<frozen importlib._bootstrap_external>",
KeyboardInterrupt

[tdial@farnarkle2 tyson]$ 
```

Most important commands for navigation and file manipulation:

`cd /fred/oz002`
`cd ../`
`cd -`
	Move to the directory `/fred/oz002`
	Move up one directory
	Go back to previous directory

`ls (-lrt) /fred/oz002`	List files in the directory `/fred/oz002` (including metadata)

`cp file file2`
`cp file /fred/oz002/.`
`cp -r folder folder2`
	Make a copy of `file` and call it `file2`
	Make copy of `file` and put it in `/fred/oz002`
	Make copy of folder and all contents (-r recursive)

`rm (-r) stuff`	Remove file (-r to remove folder and all contents)

`mv file /fred/oz002/.`
`mv file file2`
	Move file to `/fred/oz002`
	Rename `file` to `file2`

`touch file`	Create empty file
`mkdir dir`	Create empty directory/folder

`cat file`
`cat file | less`
	Display all contents of file
	Display contents of file gradually

`grep (-n) “str” file`	Find all occurrences of `str` in `file` (-n include line numbers)
`find -name “myfile.txt”`	Find file named `myfile.txt` and print out its filepath
`ln -s /fred/oz002 sym_dir`	Make directory `sym_dir` that points to `/fred/oz002`

Other Useful commands:

`pwd`	Print fullpath of current directory

`echo “hello”`
`echo $var`
	Print “`hello`” to terminal
	Print `var` variable to terminal

`clear`	clear text in terminal
`history`	list previous commands that were run in terminal
`display image.png`	Display `image.png` in GUI window (requires X11 forwarding)
`chmod <perms> file`	Change permission `<perms>` of `file`
`man <command>`	Open manual (help) of bash command
`exit`	Exit shell (ipython, ssh, sinteractive session)

SLURM in_1_page

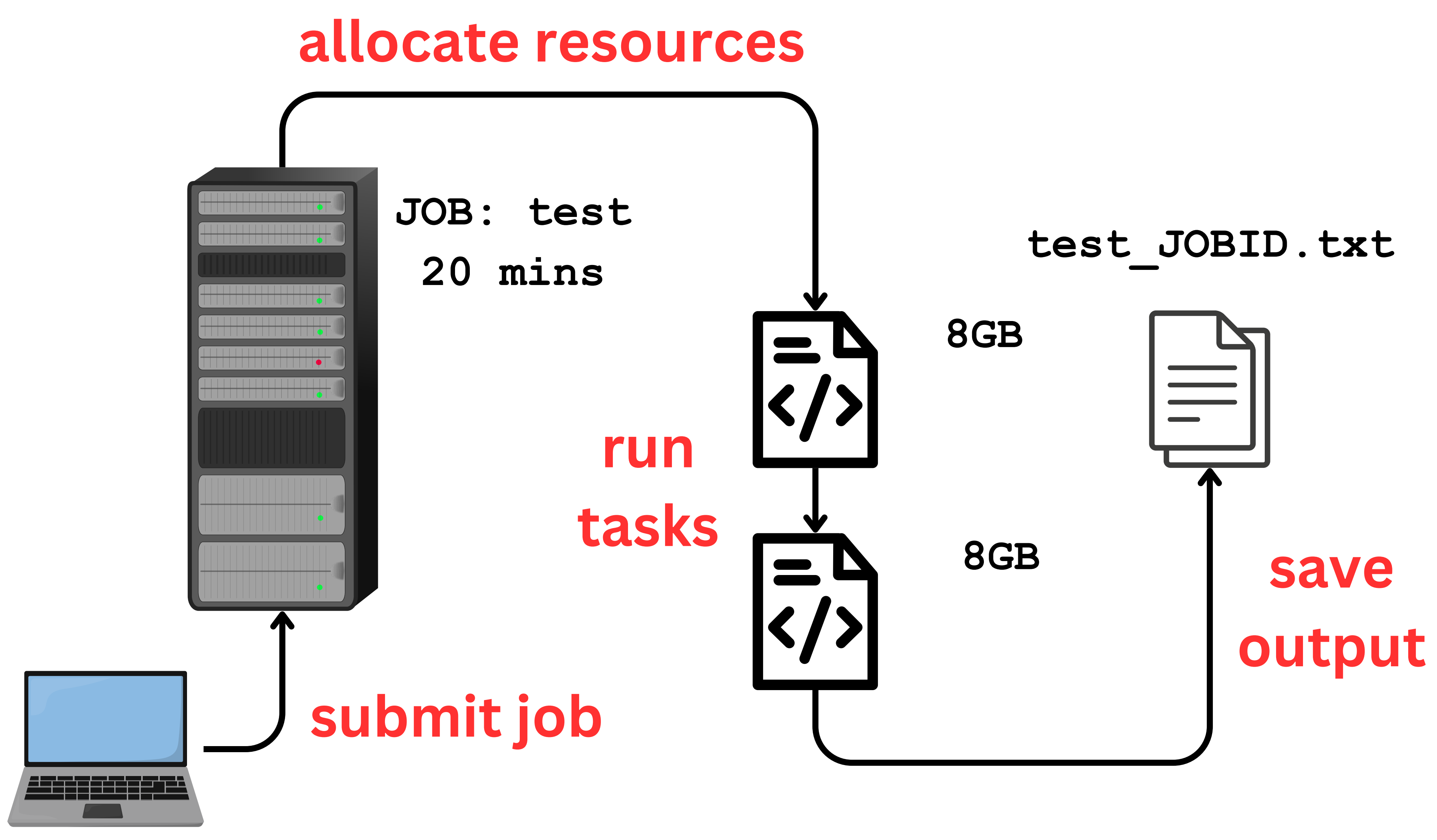
JOB

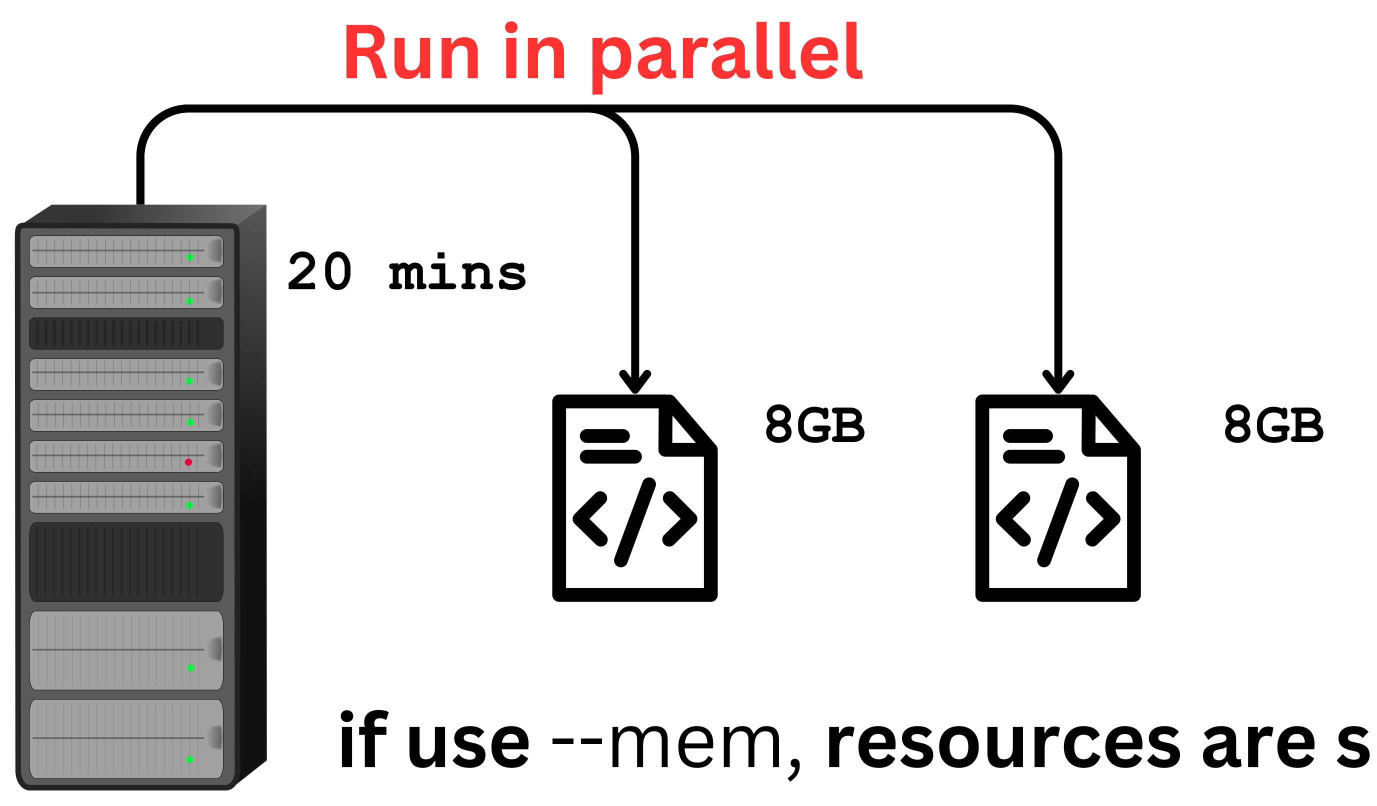
```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=test_%j.txt
#
#SBATCH --time=20:00
#SBATCH --mem=8GB

bash script.sh
bash script2.sh
```

TASKS

```
> sbatch slurm_script
```





```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=test_%j.txt
#
#SBATCH --time=20:00
#SBATCH --mem-per-cpu=8GB
#SBATCH --ntasks=2

srun --opt script.sh
```

```
> sbatch slurm_script
```

if use --mem, **resources are shared between tasks**

specify resources for srun **task**

Commands

```
[user]$ sbatch slurm_script
submitted batch job 500
```

Start slurm job with job id -j

```
[user]$ squeue -j 500
```

Check status of job
(-u <user> **will list all the users jobs)**

```
[user]$ scancel -j 500
```

Cancel job
(-u <user> **will cancel all the users jobs)**

SLURM --Options

Option	Description
--job_name=""	Name of Job
--output=""	All text outputs during runtime are saved to this file
--time=dd:hh:mm:ss	Requested time to run job
--mem=8GB	Requsted memory, units can be K, M, G and T
--ntasks=4	Run 4 parallel instances of each a task or script
--gpus=2	Requested number of gpus
--mem-per-cpu=4GB	Requested memory for each cpu
--cpus-per-task=4	Requested number of cpus per task
--mem-per-gpu=4GB	Requested memory for each gpu
--gpus-per-task=2	Requested number of gpus per task
--ntasks-per-gpu=4	Requested number of tasks per gpu (Not compatible with –gpus-per-task)
--mail-user=user@swin.edu.au	Sends emails on the progress of the job
--wrap=""	Run the specified string as a bash script using sbatch
--tmp=8GB	Specify a minimum amount of temporary disk space per node
--array=0-10	Run an array of jobs (multiple parallel jobs), 0-10 specifies the unique array-job-id

Many more Slurm options are available,
see slurm.schedmd.com/sbatch.html for a comprehensive list.