

23 Tue

A Bash

Learnt the following

- a) translate `tr`: when appending with `echo foo >> bar`, the final character is a newline (`\n`). Therefore the next thing we append appears on the next line. If we don't want this behavior, we could `tr` out the new-line character through

```
echo "foo" | tr "\n" "(whatever)" >> bar
```

which writes `foo(whatever)` to the file `bar` as usual, but whatever we append next will go to the right of `foo(whatever)` instead of to the next line.

- b) basic calculator `bc` for floats:
`echo 'scale=2;2/100' | bc` returns `.02`
- c) `sed` delimiter: turns out that any character following `s` is the delimiter, so for example `sed -i 's+hello+world+' file.txt` is the same as `sed -i 's/hello/world/' file.txt`.

B SLURM

Went through the following documentation:

- a) HPC basics
- b) Storage Documentation
- c) Data Storage → Overview of File System (*this one has a nice table*)
- d) Batch System Concepts

24 Wed

A Meeting with Julie

- a) Don't get caught up in the details of the loop. Focus on being able to navigate through it; remember what each part does and what files are called to do what (and where they are). *Finish the first assignment by next Wednesday.*
- b) Nicholas is currently working with XF for PAEA ARA-hpol (part B), so I will probably be working on the GA for ARA-hpol (part A)? Confirm with Amy on Friday.
- c) Should I start looking into AraSim and learn about Birefringence? Confirm with Amy on Friday.

B SLURM

Went over the following

- a) Batch System Concepts
- b) Batch Execution Environment
- c) Job Scripts
seems like we don't really use parallel computing introduced in this section? We just submit jobs through a job array
- d) Job Submission (mostly focused on **Job Arrays**). In particular, note that for example

```
sbatch --array=1-10%4 test.sh
```

is going to submit an array of 10 jobs, but only 4 of these will be run at the same time (so 1-4, then 5-8, and finally 9 and 10).

Some options:

- a) **A** is account (**PAS1960**); **N** is number of nodes; **t** is wall time.
- b) **n** is the number of tasks. Much better explanations available at <https://stackoverflow.com/questions/65603381/slurm-nodes-tasks-cores-and-cpus> and <https://stackoverflow.com/questions/39186698/what-does-the-ntasks-or-n-tasks-does-in-slurm>

C problems

- a) still a bit unfamiliar with the SLURM directives such as **n**. Should review the section **Job Scripts** in the future.
- b) still unfamiliar with parallel computing (multiple nodes and/or multiple cores). But since we always use only one node per job, I don't have to worry about **srun**, **sbcas**t and all those parallel computing commands (at least not now?)

25 Thu

A Meeting with Julie

- a) received Julie's candidacy paper. Plan to start reading by next meeting (Wed).
- b) Restarted homework 1 on google drive. Half-way to finishing (?)
- c) **TODO:** Look into part E tomorrow. Seems like things are not exactly as described by the dissertation appendix.

26 Fri

A Meeting with Julie

- a) Pretty much done with homework 1 part E, but I should look into it in more detail once I am more familiar with the entire loop.
- b) Going through the question 3's for the remaining parts. Should be able to finish on time as planned.

27 Sat

A Meeting with Julie

a) Mostly done with Part B1

28 Sun

A Meeting with Julie

a) Did Part C

30 Tue

A Meeting with Julie

- a) Did Part D1

B Bash

- a) comment block:

```
:<<'END'  
    Hello World!  
    This is another line.  
END
```

- b) doing arithmetic, note that many operators need to be escaped. For example:

```
expr $A \* $B  
returns  $A \cdot B$ .
```