

# An Introduction to Working in the Linux Command Line

June 8, 2016

**Note:** Do not attempt to copy/paste out of this document. Commands will likely not work!

**Some software you will need** (Note: these are examples of free software, not endorsements):

<u>In order to</u>	<u>From Mac or Linux</u>	<u>From Windows</u>
Connect to submit (ssh)	Terminal	PuTTY, MobaXterm
Move files to/from submit	FileZilla, CyberDuck (Mac)	FileZilla, CyberDuck
Edit text files	TextWrangler (Mac)	Notepad++

## Getting around in Linux:

- File paths (directories or folders): `/`, `/home/magitz/`, `/ufrc/<groupname>/magitz/`
- `pwd`, `cd`, `ls` (Where am I, change directory, list directory)
- `cp`, `mv`, `rm` (copy, move, delete)
- `more`, `less`, `head`, `tail`, `cat` (examine files)
- `nano`, `vim` (text editors in Linux)

## Making things easier:

- Tab completion- type part of a path and hit tab-key, shell will auto-complete for you
- `history`: redo something that you did before without retyping (use `↑`)
- `man`: getting help, also `-h` or `--help` flag (e.g.: `man ls`)

## Learning by doing:

1. Connect to HiPerGator 2.0: `ssh <user>@hpg2.rc.ufl.edu`
  - a. Type your password and hit return (no characters display while you type).
2. Where are you when you login? `pwd`
3. What files are there? `ls`
4. Make a directory: `mkdir test_script`
5. Now what's there? `ls -l`
  - a. Linux commands usually have flags to change how they work
  - b. `man`, `-h` or `--help` often give you help
6. Change into `test_script` directory: `cd test_script` or `cd te<tab>`
7. Copy a sample file here (`.`): `cp /ufrc/data/training/LinuxCLI/simple.sbatch .`
8. Check that the copy worked: `ls`
9. Delete that file: `rm simple.sbatch`
10. That file is now **GONE!** Not in your recycle bin or trashcan, but gone!
11. Luckily we were working with a copy, let's copy it again: up-arrow, up-arrow, return
12. Look at the file: `less simple.sbatch` (type `q` to exit when done)

13. For now, ignore lines that start with a # or #SBATCH, what does this script do?

- a. Can we run this script here? Why not?
- b. Start an interactive development job :

```
srun -p hpg2-dev -t 30:00 --pty -u bash -i
```

- c. Get back to where you want to be: `cd test_script`
- d. Now we can run our script: `./simple.sbatch`
- e. Did it do what you expected?
- f. Logout of test node: `logout`

14. Use the scheduler to run this job:

- a. Can't run from home, so...
- b. Change to scratch file space: `cd /ufrc/<groupname>/<username>`
- c. Submit the job: `sbatch ~/test_script/simple.sbatch` (~ means your home)
- d. The scheduler pays attention to the #SBATCH lines to schedule and manage your job

15. Look at the result file: `less example.out`

16. Time to get some data!

- a. Find a file on your computer, or a web site, with some text. Copy and paste that into a new file in your text editor (E.g. TextWrangler, NotePad++). We're just looking for some text to process. It doesn't matter what it is!
- b. Save the file as: **some\_text.txt** (Check that line breaks are Unix)
- c. Using your text editor or FileZilla, upload this file to your space in /ufrc
  - i. SFTP (port 22) to host: **sftp.rc.ufl.edu**
  - ii. Make a directory in `/ufrc/<groupname>/<username>/` called: `word_cloud`

<b>Do not use hpg2.rc.ufl.edu</b>
---------------------------------------

17. Go back to your ssh client and navigate to

```
/ufrc/<groupname>/<username>/word_cloud
```

18. Use `more`, `head` and `tail` to look at `some_text.txt`

19. Copy the example `cloud.sbatch` script to your directory:

- a. `cp /ufrc/data/training/LinuxCLI/cloud.sbatch .`

20. Edit this script to have your e-mail: `nano cloud.sbatch`

21. Submit the cloud script: `sbatch cloud.sbatch`

22. Check your e-mail

23. Use FileZilla to download your results.

24. Open in a web browser.