

# AN INTRODUCTION TO WORKING IN THE LINUX COMMAND LINE



August 15, 2018

## SOME SOFTWARE YOU WILL NEED

(Note: these are examples of free software, not endorsements):

| IN ORDER TO                   | FROM MAC OR LINUX          | FROM WINDOWS                    |
|-------------------------------|----------------------------|---------------------------------|
| Connect to HiPerGator (ssh)   | Terminal                   | MobaXterm                       |
| Move files to/from HiPerGator | FileZilla, CyberDuck (Mac) | MobaXterm, FileZilla, CyberDuck |
| Edit text files               | BBedit (Mac)               | MobaXterm, Notepad++            |

## GETTING AROUND IN LINUX

- File paths (directories or folders): `/`, `/home/<gatorlink>/`, `/ufrc/<group>/<gatorlink>/`
- `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** (Note: some of the data and examples are taken from software-carpentry.org):

1. Connect to HiPerGator:
  - a. **Mac:** `ssh <gatorlink>@hpg.rc.ufl.edu`
  - b. **Windows:** `hostname: hpg.rc.ufl.edu`
  - c. 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. At Research Computing, we ask that users keep most of their data in the `/ufrc` folder. Let's `cd` there: `cd /ufrc/group/username` (replace `group` with your PI's group and `username` with your GatorLink)
5. Let's make a directory to put some data in: `mkdir cli_demo`
6. 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
7. Change into `cli_demo` directory: `cd cli_demo` or `cd cl<tab>`
8. Copy some demo data here (.):  

```
cp -r /ufrc/data/training/LinuxCLI/molecules .
```

  - a. Note the `-r` to recursively copy, since `cp` won't copy directories by default
  - b. Also note the `."` at the end to copy the `molecules` directory to your current location.
9. Check that the copy worked: `ls`
10. `cd` in to the `molecules` directory: `cd molecules`
11. Look at these files with `more`, `cat`, `head`, `tail`:
  - a. `more propane.pdb` and `cat propane.pdb`
  - b. `head propane.pdb` or `head -n2 propane.pdb`
  - c. `tail propane.pdb` or `tail -n2 propane.pdb`
12. Redirects: You can redirect the output of a command to a file with the `>` character. **Caution:** This erases the file first. You can append to a file with `>>`
  - a. `wc -l *.pdb > lengths.txt`

b. Let's see what this file looks like: `cat lengths.txt`

13. Sorting: We might want the lengths sorted: `sort -n lengths.txt`

a. What happens without the `-n`?

14. Pipes: We can connect commands together by piping the output from one command into the input for the next command:

a. `wc -l *.pdb | sort -n > lengths.txt`

b. Or if we only want to know the shortest file:

```
wc -l *.pdb | sort -n | head -n1
```

15. grep: We can search for text using `grep`:

a. `grep ATOM propane.pdb`

16. awk: `awk` can do a lot, but one thing it's good at is pulling out columns from files:

a. `grep ATOM propane.pdb | awk '{print $3}'`

17. `uniq`: `uniq` is a command to find unique entries in a *sorted* list:

a. `grep ATOM propane.pdb | awk '{print $3}' | sort | uniq`

18. Loops: One of the great things about the command line is the ability to automate repetitive tasks. Let's say we want to verify that all our molecules are hydrocarbons (made up of only C and H atoms):

```
for molecule in *.pdb
> do
> echo $molecule
> grep ATOM $molecule | awk '{print $3}' | sort | uniq
> done
```

19. Deleting files: Let's get rid of the `lengths.txt` file: `rm lengths.txt`

a. That file is now gone!! There is no undo, no recycle bin or trash can. As soon as you type the command and hit return, the file is gone!

b. Be careful, but don't keep everything either!