

# An Introduction to the Linux Command Line



## Non-HiPerGator version


*Last updated: September 17, 2020*

This version of the handout is for users who do not have a HiPerGator account. It uses the free [repl.it](https://repl.it) service to run the examples. These instructions *should* work in most Bash environments, so you do not need to use repl.it, it is offered as a suggestion.

## Getting Around in Linux

- File paths (directories or folders): `/`, `/home/runner/`, `/home/runner/<crazy_name>/`
- `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, the shell will auto-complete for you
- **history**: redo something that you did before without retyping (use  arrow key )
- **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](https://software-carpentry.org)):

1. Connect to a repl.it Bash server:

1. Navigate to <https://repl.it/>
2. Click on the `<> Start coding` button
3. In the Create new repl Language selection window, type "Bash", select that, and click Creat repl.
4. When the repl opens, for the most part we will focus on the command prompt on the right-hand side. You can resize that pane to get some more room to work.
5. Copy and paste (*you will probably need to use right-click to paste*) this command to get a copy of these notes and the data for the exercise: `git clone https://github.com/UFResearchComputing/Linux_training.git`

2. Where are you when you login? `pwd`

3. What files are there? `ls`

4. Let's make a directory to put some data in: `mkdir cli_demo`

5. Now what's there? `ls -l`

1. Linux commands usually have flags to change how they work
2. `man`, `-h` or `--help` often give you help

6. Change into `cli_demo` directory: `cd cli_demo` or `cd cl<tab>`

7. Copy some demo data here (`.`):

```
cp -r ../Linux_training/data/molecules .
```

1. Note the `-r` to recursively copy, since `cp` won't copy directories by default
2. Also note the `"."` at the end to copy the molecules directory to your current location.

You could use the copy in the repository you cloned, but this keeps the directions in parallel with the directions for users on HiPerGator and practices the `mkdir` and `cp` commands. This folder originated from the [software-carpentry Shell Training materials](#).

8. Check that the copy worked: `ls`

9. Change directories into the molecules directory: `cd molecules`

10. Look at these files with `more`, `cat`, `head`, `tail`:

1. `more propane.pdb` and `cat propane.pdb`
2. `head propane.pdb` or `head -n2 propane.pdb`
3. `tail propane.pdb` or `tail -n2 propane.pdb`

11. **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 `>>`.

1. `wc -l *.pdb > lengths.txt`
2. Let's see what this file looks like: `cat lengths.txt`

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

1. What happens without the `-n`?

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

1. `wc -l *.pdb | sort -n > lengths.txt`
2. Or if we only want to know the shortest file: `wc -l *.pdb | sort -n | head -n1`

14. **grep:** We can search for text using `grep`:

1. `grep ATOM propane.pdb`

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

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

16. **uniq:** `uniq` is a command to find unique entries in a sorted list: `grep ATOM propane.pdb | awk '{print $3}' | sort | uniq`

17. **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
```

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

1. 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!
2. Be careful, but don't keep everything either!

## Additional exercises

- Which molecule has the most H atoms?
- Make a directory in your cli\_demo folder and copy the methane.pdb file there (preferably without moving from the molecules directory)
- From the molecules directory, get the head of the methan.pdb file in the directory you created above.
- Change directories to the directory you made above, rename the methane.pdb file to my\_methane.pdb.
- Edit the my\_methane.pdb file and put your name as the AUTHOR.
- How many ATOMS does methane have?
- Use an SFTP program to download the molecules folder to your computer