An Introduction to the Linux Command Line

UF | **Research Computing**
Information Technology
Home of High-Performance Computing and **HiPerGator**

Last updated: September 17, 2020

Getting Around in Linux

- File paths (directories or folders): `/`, `/home/<gatorlink>/`, `/blue/<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, 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):

1. Connect to HiPerGator:

   1. *Mac*: `ssh <gatorlink>@hpg.rc.ufl.edu` (where `<gatorlink>` is your GatorLink username)

      For additional help, watch the video tutorial on logging in with from a Mac

   2. *Windows*: hostname: `hpg.rc.ufl.edu`

      For additional help, watch the video tutorial on logging in with from Windows

   3. Note that on both Mac and Windows, when you are typing your password, no characters display while you type. Just keep typing, and hit Enter and you should be logged in.

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 `/blue` folder. Let's change directories there: `cd /blue/<group>/<username>` (replace group with your PI's group and username with your GatorLink)

   1. You can see your primary group by typing the `id` command:

```
[userA@login2 ~]$ id
uid=10856(userA) gid=1234(mygroup)
groups=1234(mygroup),1235(othergroup)
```

> In the output of the `id` command, you can see your **primary group** after the `gid=`, in this example, `mygroup`. Other groups are listed after that in the `groups=` section, showing that this user is also in the `othergroup`.

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

6. 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

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

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

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

      > The `molecules` folder of files is also available in this repository at `data/molecules`. This folder originated from the software-carpentry Shell Training materials.

9. Check that the copy worked: `ls`

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

11. 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`

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 `>>`.

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

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

    1. 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:

    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`

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

    1. `grep ATOM propane.pdb`

16. *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}'`

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

    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