WESTGRID

compute canada
regional partner

UBC

UBC100

Advanced Research Computing

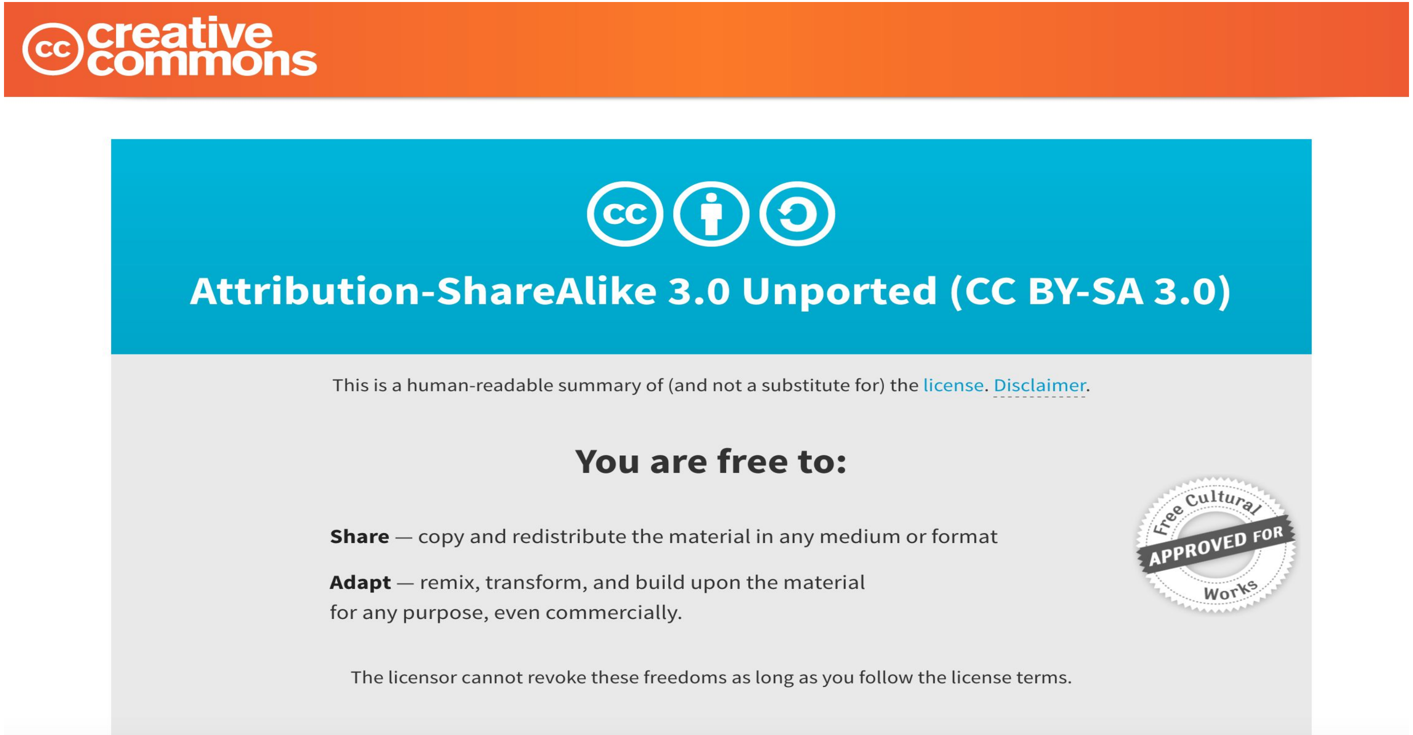
Education Outreach and Training

Tutorials

Introduction to Linux: Command Line Basics

September 26th, 2017 (10:00AM-12:00PM PST)
Phillip A Richmond

Copyright Information



Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material
for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

The material is open source, and in this presentation no previous external work was utilized.

Welcome!

- Welcome to the Introduction to Linux Command Line
- In this tutorial we will explore the command line interface for the WestGrid High Performance Compute resource (available for free to academic Canadian researchers)
- If you can, follow along with me. But if I move too fast (and I will for some people), just listen and take notes.
- This presentation will be recorded and the slides will remain available indefinitely
- DO THE PROBLEM SET AT THE END!!!

Interactive Experience

We hope this is an interactive experience for all of you.

Questions/Problems can be posted to the etherpad:

https://etherpad.openstack.org/p/EOT_Tutorial_IntroToLinuxCommandLine

We have 4 TAs to assist in answering questions and solving problems, at the end of the session I can address unresolved questions

Your own cheat sheet

Copy paste commands from the github gist:

Github Gist

(https://gist.github.com/Phillip-a-richmond/a22f4e967c1fd56235f77fbe1c7936f8)

Each command is broken down as follows:

```
# What it does (name_of_command)
## Basic/advanced usage
### template example
Actual Command Line
```

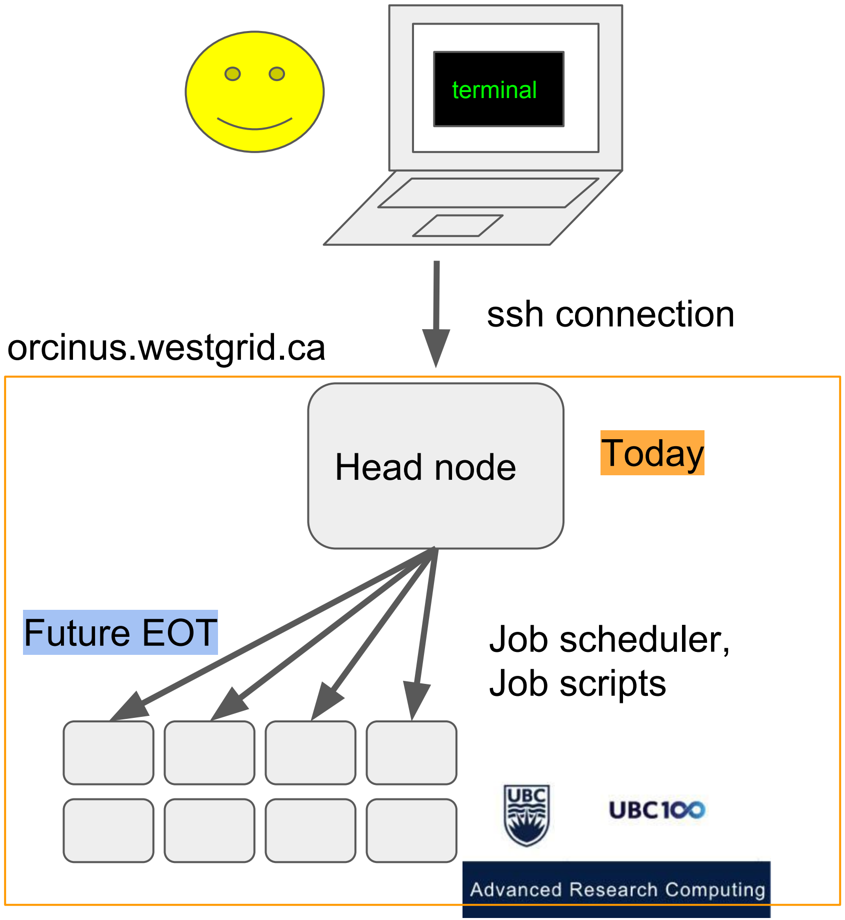
Session Learning Goals

- Basics of computing with high performance computers
- Learn about the filesystem hierarchy
- Master basic linux commands for navigating filesystem
  - cd, ls, pwd, mkdir
- Utilize basic linux commands for file handling
  - mv, cp, rm, grep, more, less, head, tail, cat
- Use the manual command (man) to get more details on linux command line options
- Download and decompress files from the internet
  - wget, gunzip, gzip
- Edit files using nano
  - nano

The things learned today will be quickly forgotten without practice and repetition. There are also other learning modules for linux that I highly recommend for beginners (see Website)

Computing via servers

- Compute servers allow us to use immense compute power, far beyond the capabilities of our own desktop
- User interacts with their own desktop
- Through a terminal, they can communicate with the head node
- The head node communicates with the execution nodes through the job scheduler



WESTGRID

Let’s get started! Login to Orcinus

You should have already attempted this by now, but as a reminder:

1. Open up a terminal (PC: MobaXterm, Putty | Mac/Linux: Terminal)
2. Login to Orcinus

Command (login):

```
$ ssh <username>@orcinus.westgrid.ca
$ ssh  richmonp@orcinus.westgrid.ca
```

NOTE: Whenever you see me represent something with the <>, I want you to replace it with what applies to you. Also, whenever there is a “$”, I am showing you a command. Commands will be highlighted, with the format in yellow, and the actual example in green

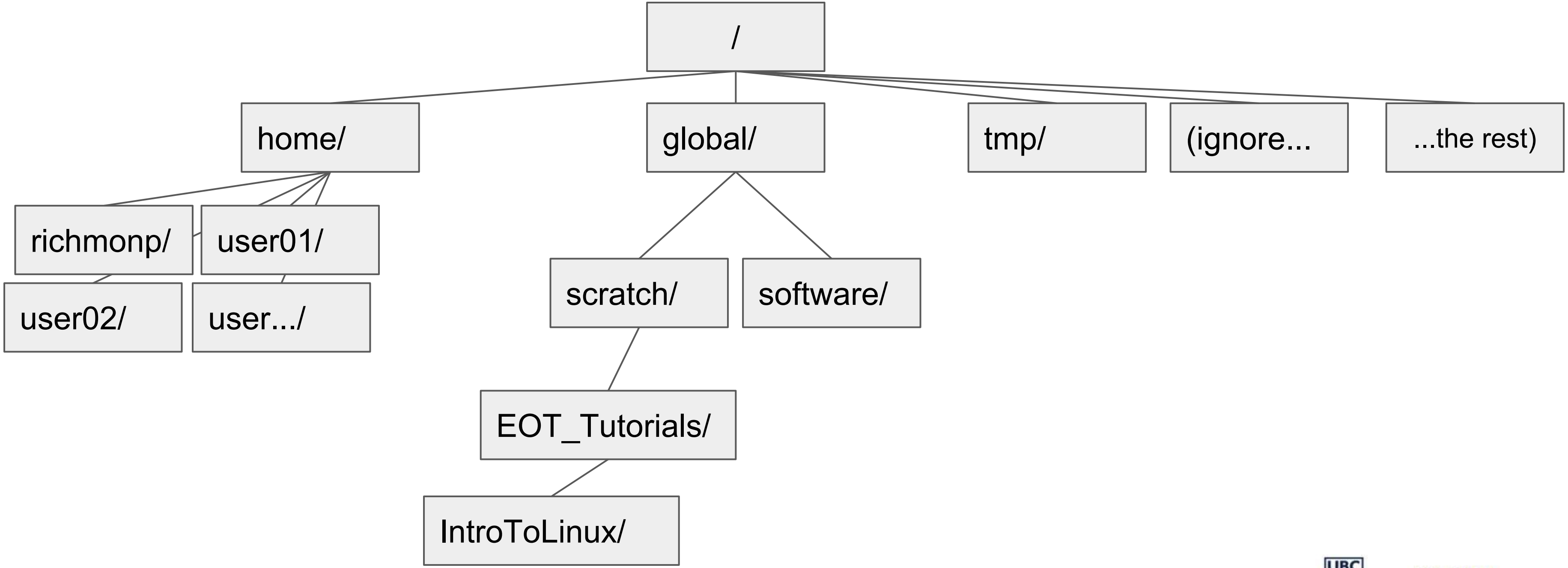
WESTGRID

UBC

UBC100

Advanced Research Computing

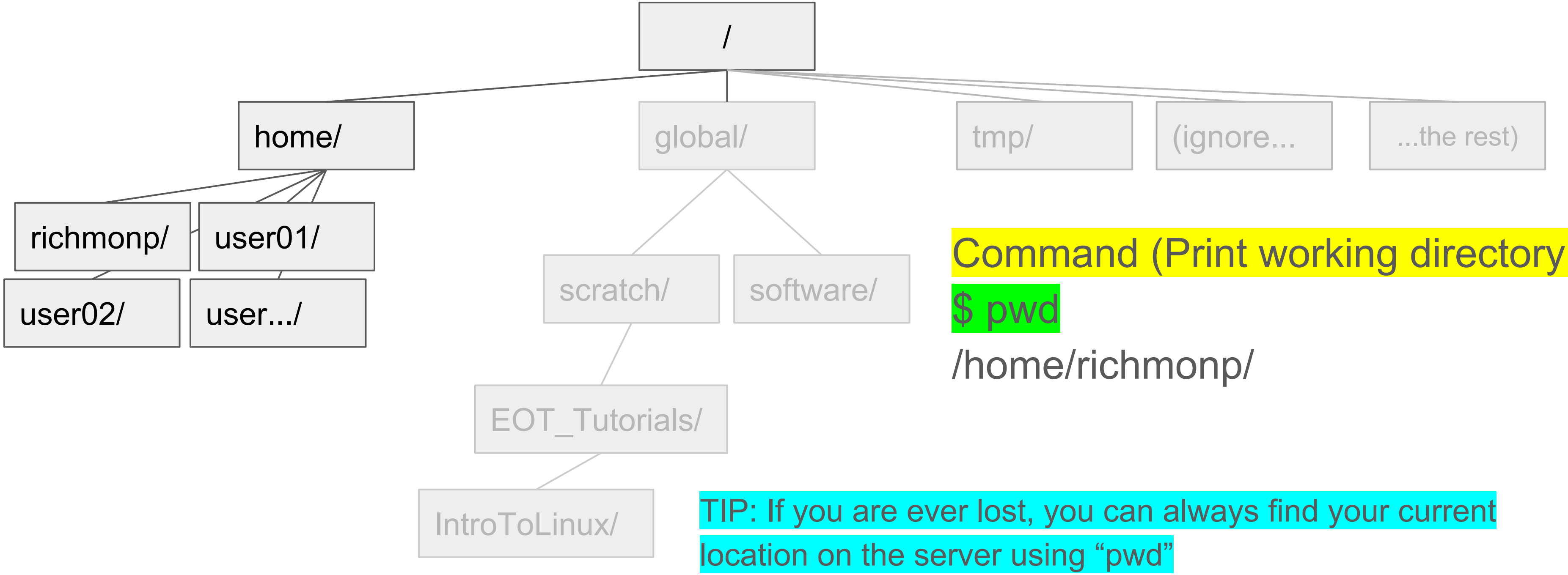
Orcinus Filesystem Organization



WESTGRID

UBC UBC100
Advanced Research Computing

First logging in: Your home directory

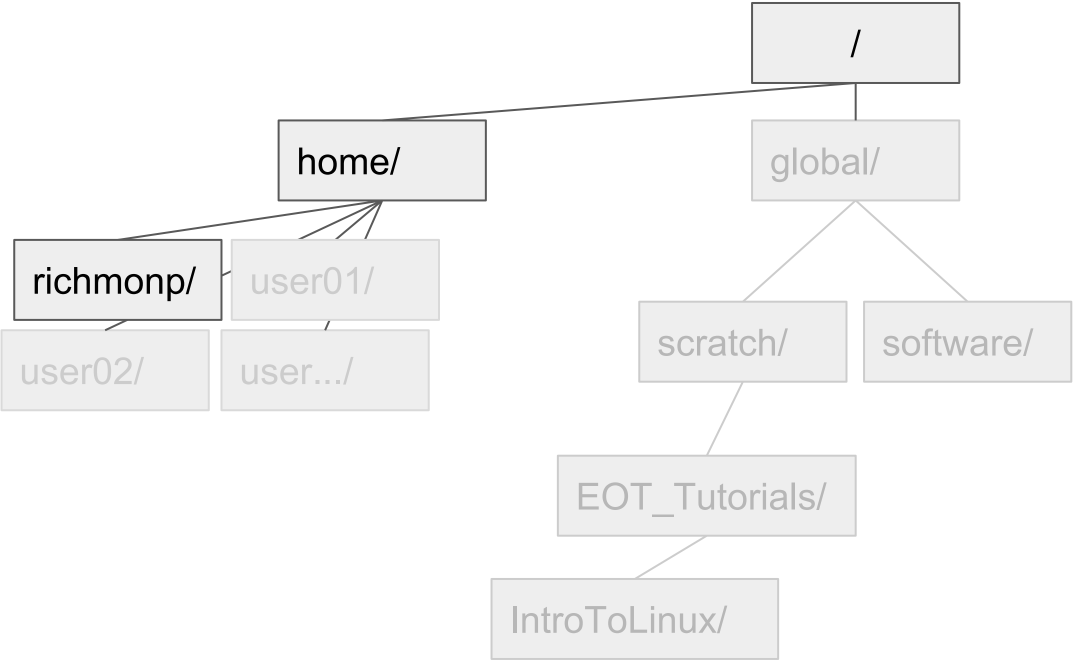


Command (Print working directory):

```
$ pwd
/home/richmonp/
```

TIP: If you are ever lost, you can always find your current location on the server using “pwd”

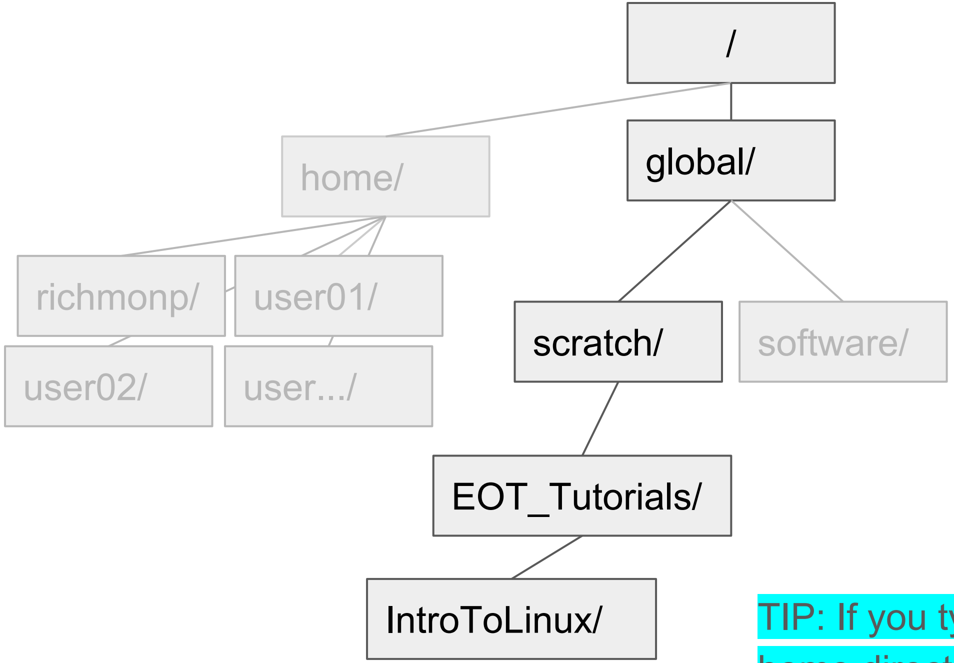
Full Filepath!



/home/richmonp/ is the “Full Filepath” for my home directory.

A full filepath means you can connect all the way to the “root” directory “/”.

Change Directory To /global/scratch/EOT_Tutorials/



Command (change directory):

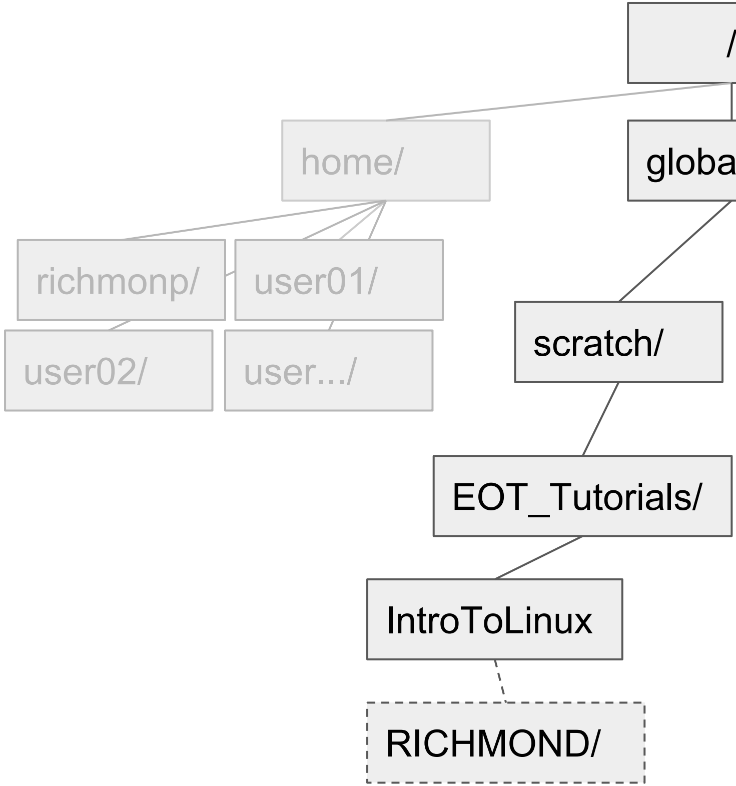
```
$ cd <filepath>
$ cd /global/scratch/EOT_Tutorials/IntroToLinux
```

Command (List):

```
$ ls <filepath>
$ ls /global/scratch/EOT_Tutorials/IntroToLinux
```

TIP: If you type “cd” without a filepath, it’ll take you back to your home directory

Make yourself a “Workshop” directory inside of IntroToLinux, title it: <LASTNAME>/



/

home/

richmonp/

user01/

user02/

user.../

global/

scratch/

EOT_Tutorials/

IntroToLinux

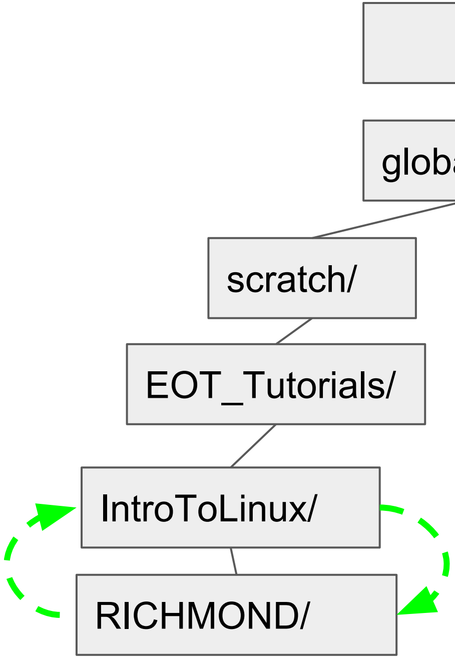
RICHMOND/

Command (make directory):

```
$ mkdir <filepath>
$ mkdir /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND
```

TIP: Anytime you create a file or directory, or change directories, do an “ls” command to double check yourself.

Let’s move around using relative filepaths



Your current directory can be referred to as ./

The directory behind you can be referred to as ../

This way, if you are in IntroToLinux/, and want to change to RICHMOND, then you can type:

```
$ cd ./RICHMOND
```

If you want to go back a directory, then you can type cd ../

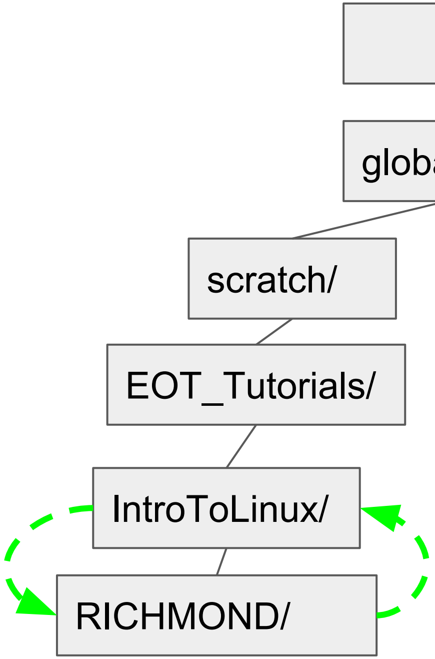
```
$ cd ../
```

WESTGRID

UBC UBC100

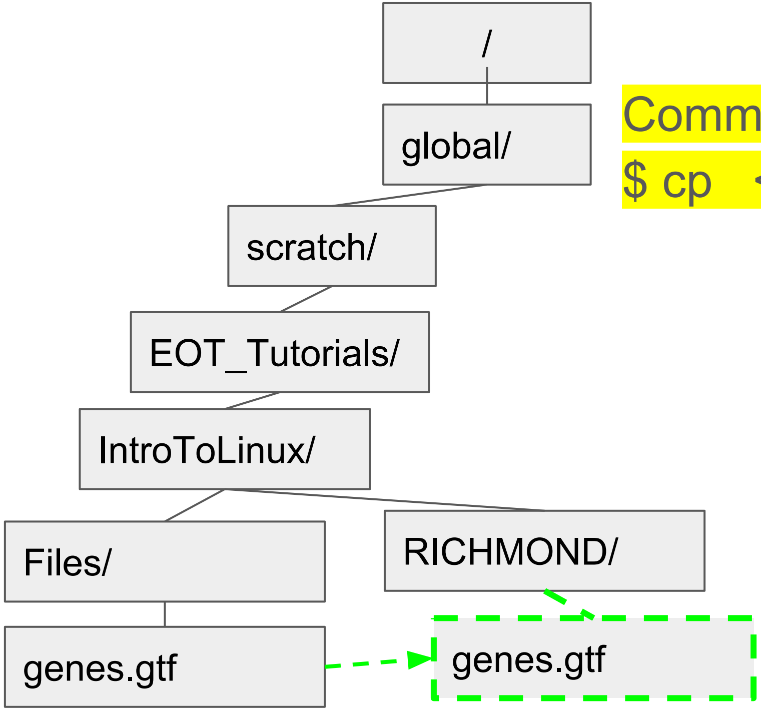
Advanced Research Computing

Interactive Navigating Example



1) Change directory to your IntroToLinux working directory
   $ cd <filepath>
   $ cd /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/
2) Check your location using “pwd”
   $ pwd
3) Move backwards one directory
   $ cd ../
4) Check your current location using “pwd”
   $ pwd
5) Check current list of directories using “ls”
   $ ls
6) Change directories back into your session 1 workshop directory
   $ cd ./RICHMOND
7) Check your current location using “pwd”
   $ pwd

Copy a file into your workshop directory



Command (copy):

```
$ cp   </filepath/for/file/to/copy>   </destination/filepath>
```

TIP: The command line will not allow you to have spaces in file names!

```
$ cp /global/scratch/EOT_Tutorials/IntroToLinux/Files/genes.gtf
/global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/
```

Long listing of a file

Let’s check to make sure we have the whole file copied over. You can get details on a file using the “ls” command with different options or parameters.

The options we’ll add will be “-l and -h”

```
$ ls -l -h <filename>
$ ls -l -h /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

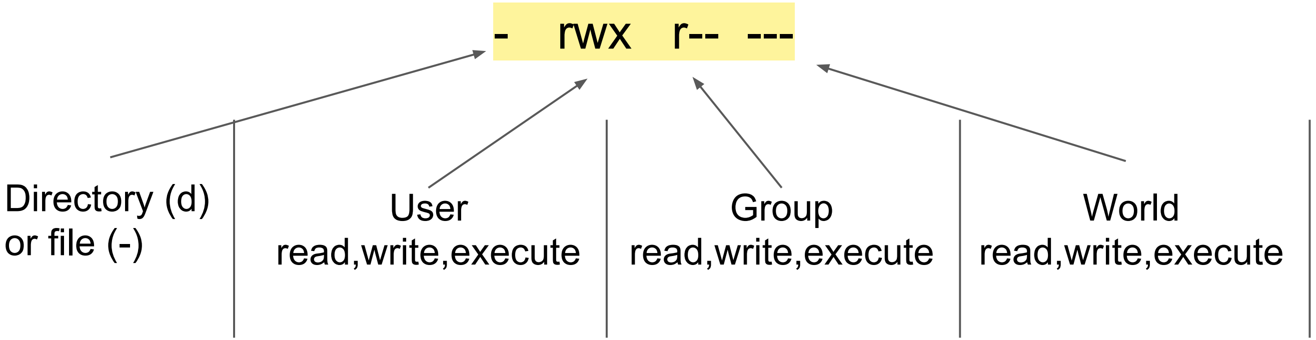
-rwxr-----	1	richmonp	richmonp	263M	May 4 13:37	genes.gtf
permissions	?	group of file owner	File owner	File size	Date of creation	File name

TIP: Look at this file size often. It will let you know when a file is completely empty

Permissions

Permissions will dictate who can read, write and execute. The permissions line is broken down like this:

r = read, w = write, and x = execute, - = none



- rwx r-- ---

Directory (d) or file (-)

User read,write,execute

Group read,write,execute

World read,write,execute

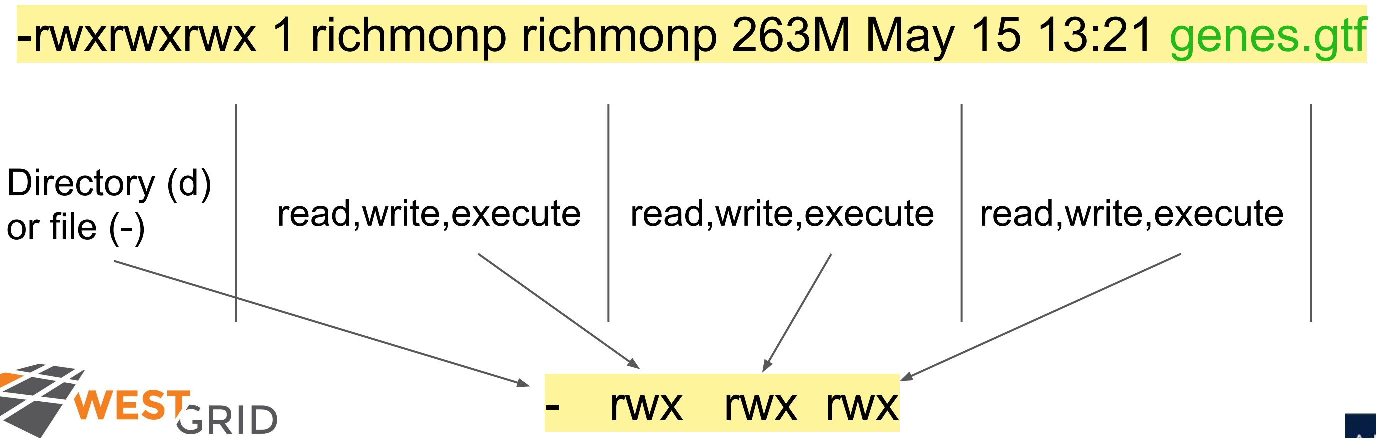
Permissions, Changing them

To change permissions, we will use the chmod command

Where u = user, g = group, and o = world.
And r = read, w = write, and x = execute

TIP:add -R to change permissions on all contents inside a directory

```
$ chmod ugo=rwx <filename>
$ chmod ugo=rwx /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
$ ls -lhtr /global/scratch/EOT_Tutorials/RICHMOND/genes.gtf
```



```
-rwxrwxrwx 1 richmonp richmonp 263M May 15 13:21 genes.gtf
```

Directory (d)
or file (-)

read,write,execute

read,write,execute

read,write,execute

```
-   rwx   rwx  rwx
```

Read the contents of the file genes.gtf

These are some useful commands for exploring the contents of a file:

More - Opens up a human readable file:

```
$ more <filepath>
$ more /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

Less - Opens up a human readable file:

```
$ less <filepath>
$ less /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

TIP: You can escape this viewer by typing “q” or “control+Z”

Head, tail, cat

Head - prints the first few lines of a file to standard out

```
$ head <filepath>
$ head /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

Tail - prints the last few lines of a file to standard out

```
$ tail <filepath>
$ tail /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

Cat - Concatenates the full file to the standard out

```
$ cat <filepath>
$ cat /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

Standard out, or printing to the “screen”

Some commands will print to standard out, which you can think of as printing to the screen.

However, we can capture what would be printed to the standard out using the “>” character. Let’s take the head command from before:

```
$ head /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

Now if we add a “>” character, then instead of printing to the screen, it’ll get captured in the filename we follow “>” with.

```
$ head /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf   >
/global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/SomeLines.gtf
```

We can do this with any command that prints to the standard out, including the next few examples.

Remove (rm)

When a file is no longer useful, or is the result of just testing a command, we can delete it...FOREVER!

The rm command is a ruthless command that will delete any file you give it, without a way to retrieve the file after you enter the command.

```
$ rm <filename>
$ rm /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/SomeLines.gtf
```

Grab Regular Expression (grep)

The grep command allows us to search a file for a regular expression. Since that's beyond the scope of this lesson, we can simply use it to search for patterns or strings, much like a Command+F

```
$ grep   <“pattern”>   <filename>
$ grep “PAX6” /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf
```

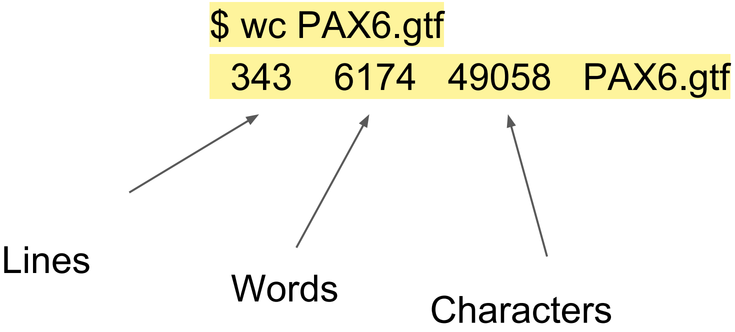
Now if we add a “>” character, then instead of printing to the screen, it'll get captured in the filename we follow “>” with.

```
$ grep “PAX6” /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/genes.gtf   >
/global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/PAX6.gtf
```

Check File Contents with Word Count (wc)

Another useful trick for checking the contents of a file is to see how many lines it has.

```
$ wc <filename>
$ wc /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/PAX6.gtf
```



```
$ wc PAX6.gtf
  343    6174   49058   PAX6.gtf
```

Lines

Words

Characters

Every command has options, explained in man

Any linux command has a manual page associated with it.

To access the manual, you use the Manual command (man)

```
$ man <command>
$ man wc
```

What option would we use to only have the wc command print lines?

Getting files from the web (wget)

Often times you will need to download data files from the internet. Anything from reference genome materials, to raw data.

To do this, we can use the web get command (wget)

```
$ wget <url>
$ wget ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/tab_delimited/variant_summary.txt.gz
```

Unzipping compressed files

The files we download from the internet will often be compressed, as labelled by the file extension (in the case of our clinvar file, .gz).

There are many different forms of compression/decompression: tar, gzip/gunzip, zip/unzip. Here, we see that we have a .gz file, so we can use gunzip.

```
$ gunzip <filename>
$ gunzip  /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/variant_summary.txt.gz
```

If we want to zip it back up:

```
$ gzip <filename>
$ gzip /global/scratch/EOT_Tutorials/IntroToLinux/RICHMOND/variant_summary.txt
```

Text editors

There are three main text editors to choose from, and each has a learning curve.

Emacs is the most popular editor, nano is the easiest editor, and vi is the most powerful but it has a steep learning curve (this is just from my experience).

Emacs has both a GUI (graphical user interface) and a terminal version. The GUI is self explanatory and behaves like basic text editors. The terminal version is controlled by commands that can be found on the cheat sheet:

http://www.rgrjr.com/emacs/emacs_cheat.html

In today's session I'll be using nano. My preferred editor is vim, and if you're on MobaXTerm I recommend emacs.

Edit a new file using Nano

Edit a new file called README.txt

```
$ nano <filename>
$ nano README.txt
```

[Save and exit]: Ctrl+o then Enter, Ctrl+x

Type in your name and the date, and then save (save: type Ctrl+o, hit enter, then quit with Ctrl+x).

FLASH DEBUGGING

```
$ ls /global/scratch/EoT_Tutorials/IntroToLinux/
ls: cannot access
/global/scratch/EOT_Tutorials/IntroToLinux/: No such file
or directory
```

```
$ cp PAX6.gtf
cp: missing destination file operand after `PAX6.gtf'
Try `cp --help' for more information.
```

```
$ mroe genes.gtf
-bash: mroe: command not found
```

```
$ chmod rwx=ugo PAX6.gtf
chmod: invalid mode: `rwx=ugo'
Try `chmod --help' for more information.
```

```
$ grep PAX6.gtf
… (scratches head, why is nothing happening?)
```

Fix: Use proper capitalization, use tab-complete.

```
$ ls  /global/scratch/EOT_Tutorials/IntroToLinux/
```

Fix: You forgot the destination silly.

```
$ cp  PAX6.gtf  PAX6_hg19.gtf
```

Fix: Watch your spelling, type with 2 hands carefully.

```
$ more genes.gtf
```

Fix: Order matters for permissions: ugo=rwx Not rwx=ugo

```
$ chmod ugo=rwx PAX6.gtf
```

Fix: You forgot to specify what you’re searching for.

```
$ grep “CDS” PAX6.gtf
```

End of Lecture, what to do next

- Take a quick break
- Ask a question
- Start on the Problem_Set
  - Review the commands in the Gist
  - Use your linux/unix cheat sheet
  - Work with others!

Learning Linux is like learning a language. Don't practice it, and you will lose it.

- Additional Linux Resources:
  - http://www.ee.surrey.ac.uk/Teaching/Unix/
- Additional Genomics Resources:
  - https://phillip-a-richmond.github.io/Introduction-to-Genomic-Analysis/

WESTGRID

UBC UBC100

Advanced Research Computing

Acknowledgements

- Phil Richmond (Teacher)
  - PhD Student Wasserman Lab, enjoys teaching
- Assorted TAs
  - Da real MVPs: Elizabeth, Oriol, Rashedul, Robin
- WestGrid https://www.westgrid.ca/ (Jana Makar)

WESTGRID

UBC UBC100

Advanced Research Computing