Getting to Know the Command Line
What is Linux?

Joel Frahm, CU Boulder Research Computing

Joel.Frahm@colorado.edu

www.rc.colorado.edu

Slides: https://github.com/ResearchComputing/Basics_Supercomputing

Outline

- Why use Linux?
- What is Linux?
- Making Life Easier
- Logging In
- The Command Line
- Some Linux Commands
- File Permissions and Ownership
- Redirecting and piping output
- Editing files
- Resources

Why Use Linux?

- It’s the only option in many cases
  - HPC systems use Linux due to cost and performance
  - Open source support
  - Workarounds to avoid the command line can be problematic
- Linux is very useful and appropriate to Science and Engineering
- Great for collaboration
  - Sharing files, directories is easy and has fine grained control
  - Facilitates borrowing and expanding on others’ tools and knowledge

- Why does Linux seem deliberately obfuscated and difficult?
  - The history extends back into the time when keystrokes cost more
  - Many different contributors over many years
  - No design integration, organic growth
  - Some command names make sense once explained
  - Many are whimsical or funny (this helps you remember them)

What is Linux? What is the "Shell?"

- **Linux is an Operating System**
  - **You use it to execute commands and run programs**
  - **Programs use it to interface with the "bare metal"**
- **Part of the Unix family of Operating Systems**
- **Really great story: Look it up on the Wikipedia or wherever.**
- **RC uses RedHat (Enterprise 7 on Summit)**
- **Other distributions are similar , there are standards and standard tools**
- **The "Shell" is the command interpreter (interprets your commands)**
  - **We are using BASH – The Bourne Again SHell**
  - **There are a few others commonly used but BASH is most common in RC**
  - **Some people use tcsh, based on the C programming language (NCAR)**
  - **It's possible to run a script written for a different interpreter**
  - **i.e. a tcsh script can be run in BASH if the top line calls it**

Making life easier

- Terminal tweaks
  - Colors
  - Scollback Buffer size
- Memorizing Syntax: Don’t!
  - Copy/paste from a cheat sheet
  - Search your history
    - Control-R and start typing
    - `grep <string> ~/.bash_history |tail -n5`
  - Search for syntax examples on the web
  - Directed google searches for the better sites
  - Syntax will become natural… eventually…

Logging In

- **OSX has a Terminal window to a BASH shell but…**
- **ssh to an RC login node for this session**
  - **Windows – usually use PuTTY**
  - **OSX – Open the Terminal and use the command line**
  - **Linux laptop users – play Minecraft or something instead.**
  - `$ ssh user00XX@tutorial-login.rc.colorado.edu`
  - `$ ssh -l user00XX tutorial-login.rc.colorado.edu`
  - Typical non-workshop RC login uses hostname login.rc.colorado.edu
- **Use your supplied username and password for the exercise**
- **Typical RC logins use "2-factor" authentication**
- **After your login is authenticated**
  - **Shell starts**
  - **Hidden files are parsed, environment variables set**
  - **You should get a $ prompt**

The Command Line or Shell

- Run commands
  - Start an editor
  - Queue or check on your jobs
  - Transfer files
  - etc.
- Navigate the filesystem
- Features
  - Tab completion
  - History file ~/.bash_history
  - You can program right from here (or script)
    - It’s easier than it sounds!
    - Super powerful

Some Linux Commands

- **Moving around:** `cd, pwd, .`
- **Create/Delete directories:** `mkdir, rmdir, rm –r`
- **Create/Delete files:** `touch, rm,` **(editing)**
- **Create via redirection > and >>**
- **Listing files and directories:** `ls, ls –l, ls -1, ls –ld`
- **Copy/Move(rename) files/directories:** `cp, mv`
- **Working with permissions and ownership:** `chmod, chown`

File Permissions and Ownership

- chmod: set rights for User, Group and Others
- Set mine to Read Write eXecute, Group's to Read eXecute, Others to Read
  - chmod u=rwx,g=rx,o=r myfile
- Copy my rights to my group, recursively
  - chmod –R g=u mydir
- chown: Change Owner (and Group)
- Change all files in this directory to my advisor and his group
  - chown -R advisor:advisorgrp /files/data
- It is common for a group to combine files in a directory with Group rights
- Copying files in does not usually set the ownership, it carries over
- Permissions and ownership often need to be cleaned up
- Process: Copy over files, set permissions, set owner/group

Redirecting and Piping output

- **Output sent to a file**
  - **“>” – sends the output to a file, creating or overwriting file.**
  - **“>>” – Creates file or appends output to existing file**
  - **Redirecting stderr or stdout**
  - `command > out 2>error`
  - `command 2>/dev/null`
- **Output “piped” to another command**
  - `grep "zip" bootcamp.file |wc -l`

Editing Files

- **Editors: nano, vi, emacs…**
- **nano is light and easy, the commands are on the screen**

Resources

- Websites
    - stackoverflow.com
    - unix.stackexchange.com
    - www.gnu.org for man pages
- Google – results are a mixed bag
- Directed google search examples:
- site:stackoverflow.com list files with first line match
- site:unix.stackexchange.com list files over size

To get help:

- RC can help with Linux questions (as appropriate to using RC resources)
- …but searching or posting on more specific help forums can be faster
- When posting in an online help forum:
    - Better to show your attempt and results than to just ask for help
    - Be specific if you can

Thanks!

- Questions? Joel.Frahm@Colorado.EDU