

# Introduction to Linux Shell Basics

Martin Čuma & Wim Cardoen

Center for High Performance Computing

June 7, 2016

# Linux or Mac Desktop

- First step is to open up terminal
- If you have a CHPC account:
  - ssh `unid@linuxclass.chpc.utah.edu`
- If not:
  - ssh [userXX@linuxclass.chpc.utah.edu](mailto:userXX@linuxclass.chpc.utah.edu)
- Class info, slides, etc:
  - <https://www.chpc.utah.edu/presentations/IntroLinux2parts.php>

# Windows

- Need ssh client
  - PuTTY
    - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
  - XShell4
    - [http://www.netsarang.com/download/down\\_xsh.html](http://www.netsarang.com/download/down_xsh.html)
- For X applications also need X-forwarding tool
  - Xming (use Mesa version as needed for some apps)
    - <http://www.straightrunning.com/XmingNotes/>
  - Make sure X forwarding enabled in your ssh client

**OR** use FastX2

see <https://www.chpc.utah.edu/documentation/software/fastx2.php>

# Terminal Basics

- Shell is the interface between your commands and the operating system (OS)
- Two basic shells - slightly different command syntax
  - csh/tcsh
  - sh/**bash**
- Type commands on command line, send command by pressing enter
- Linux is case sensitive!
- Commands can take flags that modify their behaviour
  - flags are formed with – (dash) and letter
- There are online manual pages for all commands which list the flags – access by `man somecommand` (e.g., `man ls`)

# Prompts, Usernames & Directories

- When you first login you will see a prompt  
[unid@linuxclass:~]\$ or [userxx@linuxclass:~]\$
- To see your username: **whoami**
- To see your current directory: **pwd**  
/uufs/chpc.utah.edu/common/home/<yourusername>  
/home/<yourusername>
- Shortcuts  
~**yourusername** → your home directory  
**\$HOME** → your home directory

# Paths and Working with Directories

- `ls` – list contents of a directory
- `mkdir` – make directory (`mkdir test`)
- `cd` – change to directory (`cd test`)
- `rmdir` – remove directory (`rmdir test`)

## Additional Information:

- `/path/from/root` → absolute path
- `path/without/leading/slash` → relative path from current location
- `.` → current directory
- `..` → parent directory (up one level)

# Wildcards

- more files can be specified via wildcards
- \* - matches any number of letters including none
- ? - matches any single character
- [ ] - encloses set of characters that can match the single given position
- - used within [ ] denotes range of characters

## Examples:

```
*.csh , *.*sh , figure?.jpg , *.txt ,  
figure[0-9].*
```

# Exercise 1

- Make a directory called `IntroLinux1` and change into this directory
- Copy over the contents of my (u0101881) `talks/IntroLinux1-Jun2016` directory  

```
cp /uufs/chpc.utah.edu/common/home/u0101881/talks/IntroLinux1-Jun2016/* .
```
- List contents of this directory – see difference of a normal `ls`, `ls -l`, `ls -ltr`, and `ls -ltra`
- See what output you get when you do a `ls` of: `figure?.jpg`, `figure[0-9].*`
- Make a new directory called `Work` inside of `IntroLinux1` and copy all files with the `txt` extension from the `IntroLinux1` directory to your new directory
- Open man page for some command (e.g. `ls`) and see what these flags do



# File commands

- **cat** – display contents of file
- **more** – display contents of file with page breaks (next page with Space key)
- **head** – display top of file
- **tail** – display end of file
- **grep** – search for pattern in file (`grep "pattern" test1`)
- **vi** – edit file (more on this later)
- **cp** – copies file to a new name (`cp file1 file2`)
- **mv** – renames file to a new file (`mv old new`)

# Exercise 2

- Change into the `Work` directory you created in Exercise1
- View `states-capital.txt` using `cat`, `more`, `head` and `tail`
- Vary number of lines viewed with `head` and `tail`
- Search for the string `New` in this file with `grep`

# Command output redirection

- ▣ > redirect to a new file
  - ▣ `cat file1.dat > file4.dat`
- ▣ >> - append to a file
  - ▣ `cat file1.dat >> file3.dat`
- ▣ | - pipe – redirect command output to another command
  - ▣ `head -10 list.txt | tail -2`

# Exercise 3

- ❑ In the `Work` directory, combine the contents of `geom1.txt` and `geom2.txt` into one file named `geom3.txt`
- ❑ Using `grep` and the file `states.dat` create a file `Mstates.dat` with only the states that start with the letter `M`
- ❑ Create the same file content using `head` and `tail`

# Unix File Permissions

- ❑ Shown with `ls -l`
- ❑ User (u), group (g), other (o), all (a)
- ❑ Permissions are read (r), write (w), execute or search for a directory (x)
- ❑ `chmod` – to change permissions of file or directory
- ❑ Format `chmod g+x file`
- ❑ Executable files (programs and scripts) must have executable permissions

# Some other useful commands

- ▣ `wc - e.g. wc -l file.txt`
  - ▣ `cut - e.g. cut -f 2 -d : file.txt`
  - ▣ `du - e.g. du -hs`
  - ▣ `df - e.g. df -h`
  - ▣ `ln - e.g. ln -s ~/bin/prog.exe prog1.exe`
- 
- On your own -- Use man pages to find out what these commands do.

# The vi editor

- Two modes – command, input
- Use arrow keys to move cursor to location
- Command mode – commands input via keyboard keys
  - `i`, `a`, `r`, `R` – moves to the input mode – insert, append, replace character, replace
  - `G` – go to (`1G` – go to line 1, `G` – go to end of file)
  - `x`, `dd` – delete character, line
  - `:` – enter external command (`:w` – write file, `:q` – quit, `:q!` – quit discarding changes, `:wq` – write and quit)
  - `/`, `?` – search forward, backward (`/test`); `n` goes to next occurrence
- input mode – type in content

# Use of the vi editor

- to input text, enter input mode
- to quit input mode, push Esc key
- searching, deleting,... done in command mode
- search and replace:
  - :s/old\_text/new\_text – replace next occurrence on current line
  - :%s/old\_text/new\_text – replace all occurrences on whole file
  - :s/old\_text/new\_text/g – replace all occurrences in the current line



# Exercise 4

- ❑ Open a file with vi - `vi script-csh.slurm`
- ❑ Go to the start of the string `youraccount`
- ❑ Delete this string and insert the string `owner-geust` in its place
- ❑ Exit edit mode (`ESC`), and save the file (`:w`).
- ❑ Oops – we found a typo – it should be `guest`. Use replace to fix it.
- ❑ Save file again and quit (`:wq`)
- ❑ Practice some more editing with `vi`

# Other Useful Items

- Up/down arrows go through past commands
- **history** – provides list of all recent commands; can ! followed by number from history list will put that command at the prompt
- Tab completion – of commands, paths, filenames

# Have Questions?

- ❑ Martin: [martin.cuma@utah.edu](mailto:martin.cuma@utah.edu)
- ❑ Wim: [wim.cardoen@utah.edu](mailto:wim.cardoen@utah.edu)
- ❑ CHPC has an issue tracking system:  
[issues@chpc.utah.edu](mailto:issues@chpc.utah.edu)
- ❑ Slides and files  
<https://www.chpc.utah.edu/presentations/IntroLinux2p arts.php>
- ❑ Some useful websites  
<http://www.ctssn.com/>  
<http://unix.t-a-y-l-o-r.com/Unix.html>