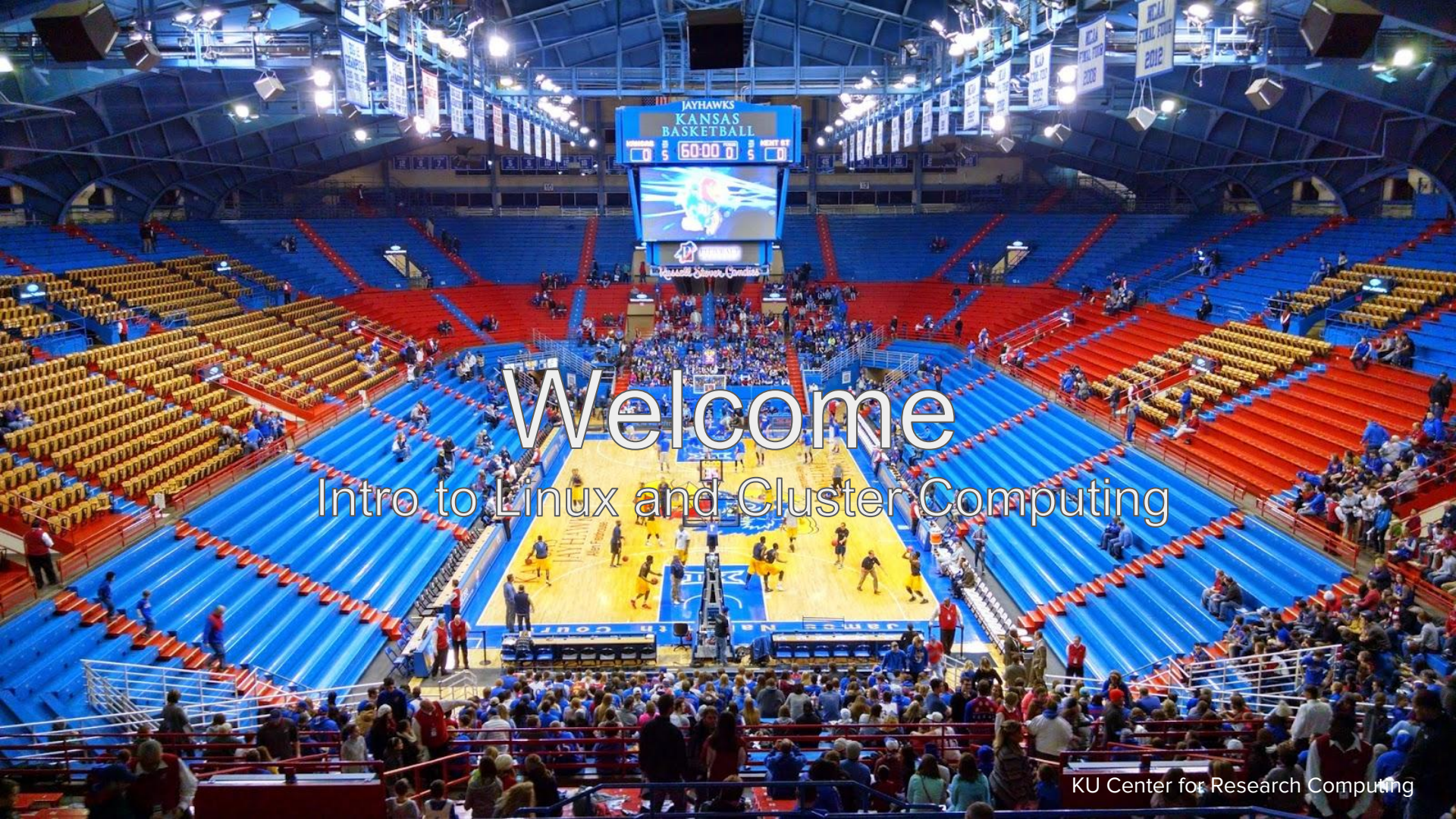




Welcome

Intro to Linux and Cluster Computing



0 - What is Linux?

1 - I've used Linux once or twice

2 - I can use cd and ls

3 - I prefer Linux over Windows

4 - I grep and awk daily

5 - I run Gentoo and write my own drivers



What is Linux

Operating System, just like Windows or MacOS

Completely Free

Open Source - Anyone can view the source and make modifications

Much faster and uses far fewer resources leaving more for computation



SSH?

Stands for Secure Shell

Allows you to sign into a Linux machine remotely

Gives the same control as sitting in front of the box

Can see graphical windows using X-Forwarding (`ssh -X`)

Just because you've ssh'd doesn't mean your files transfer over



(S)FTP

Stands for Secure File Transfer Protocol

Can use WinSCP or Filezilla to move files to and from another Linux machine

Gives remote file access to a machine

At KU you likely won't use this because we have a service called Globus, but it's good information to know



Basic Navigation

`pwd` - Print working directory

`ls` - List directory contents

`cd` - Change directory

`.` (dot) - Reference to the directory you're currently in

`..` (dot dot) - Reference to the parent directory of the directory you're currently in

Tab complete - A gift from the Linux gods



'man' is your friend

man stands for manual

Nearly any command you want to know more about you can type

```
man <command>
```

Gives switches and modifiers for command syntax

example: `man ls`

If that doesn't work try `<command> --help` or just plain old Google



Common Directories

~ Tilde is your home (`cd` shortcut)

`/home/<username>` -or- `/usr/local/home/<username>`

`/etc/`

`/opt/thingy`

`$HOME`, `$WORK`, and `$SCRATCH`, useful but only work for the KU Community Cluster



File manipulation

`mkdir`

`rm`

`rmdir (or rm -r)`

`cp`

`mv` (this also renames)

THERE IS NO 'UNDO' so be absolutely sure these commands are what you want



Text editing from the command line

Different editors for different styles (nano, vim, emacs)

Nano is simple

Write simple job files, or complex programs

If you're feeling adventurous, I'd recommend vim

If you just don't like CLI, you can also use Notepad and copy in files through SFTP
That's just more hassle, and rarely works for complicated reasons



*-no longer just for obscuring swear words

The * character is a 'wildcard' meaning it can stand in for varying parts of a name

example:

You want to delete files win95.txt, win98.txt, winxp.txt, winvist.txt, win7.txt, win8.txt

You can call `rm win*.txt` and it will delete everything that follows that pattern

As with any operation, whatever you do in the command line is permanent,
be careful!



Environment Variables

Used to save things you'll need again and again

File paths are a good example

example: `export POINT=/home/n227e821/work_stuff/pointwise_stuff`

Now I can `'cd $POINT'` instead of the long path

Works for other things you'd like to store as a variable

All-caps is good practice, but not a requirement



Useful tools

head View the first n lines of data.

tail View the last n lines of data.

sort Organise the data into order.

nl Print line numbers before data.

wc Print a count of lines, words and characters.

awk Filter

cut Cut the data into fields and only display the specified fields.

uniq Remove duplicate lines.

cat Print data from a file or stream, or concatenate streams/files

tac Print the data in reverse order.

grep Search for a string [or anything really]

sed Search and replace utility



Piping and Redirection

Use pipe to send the output of one process to the input of another process

ex: `ps aux | grep n227e821`

Redirect the output of a process to a file or stream

ex: `ps aux > runningProcesses.txt`



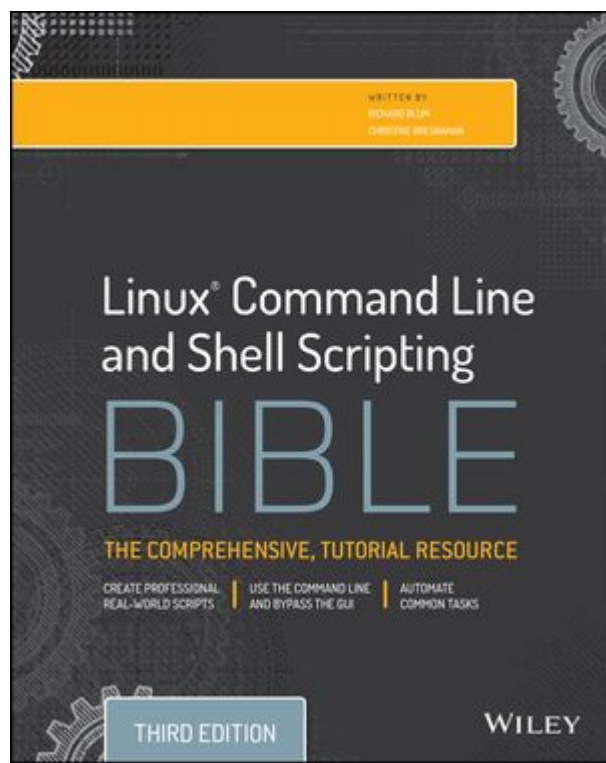
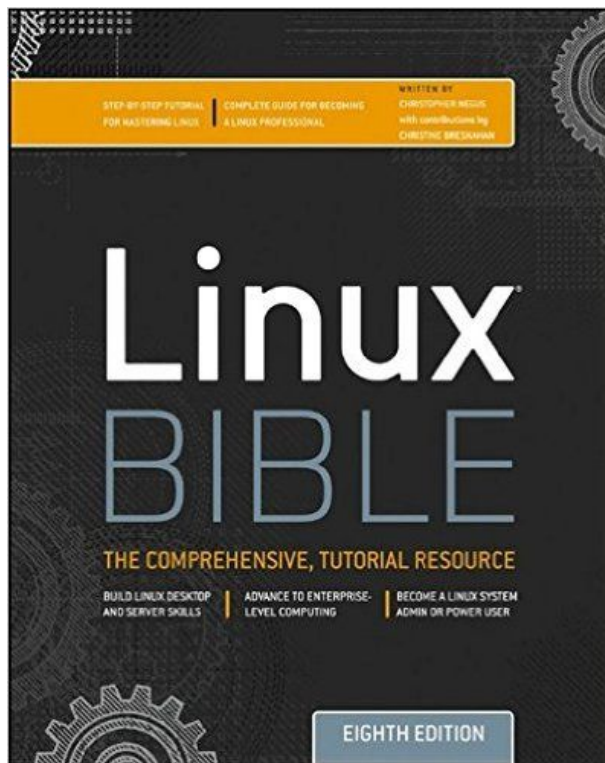
Scripting

All of the commands we've covered so far and even more can be combined to make little programs called scripts

You need to use an interpreter to interpret these commands, we use BASH.

Sample:

```
#!/bin/bash
for WORD in some_list
do
    rm $WORD.txt
done
```





General guidelines on installing software

Load modules you'll need. Compilers, MPI Libraries, and dependencies

Start an interactive job to install in a similar environment to where the software will run

Keep in mind file system quotas, your home directory is only 100 GB

Makefiles are your new friend, but are fickle

`./configure, make, make install`

`--prefix` is a great option



General guidelines on installing software - cont'd

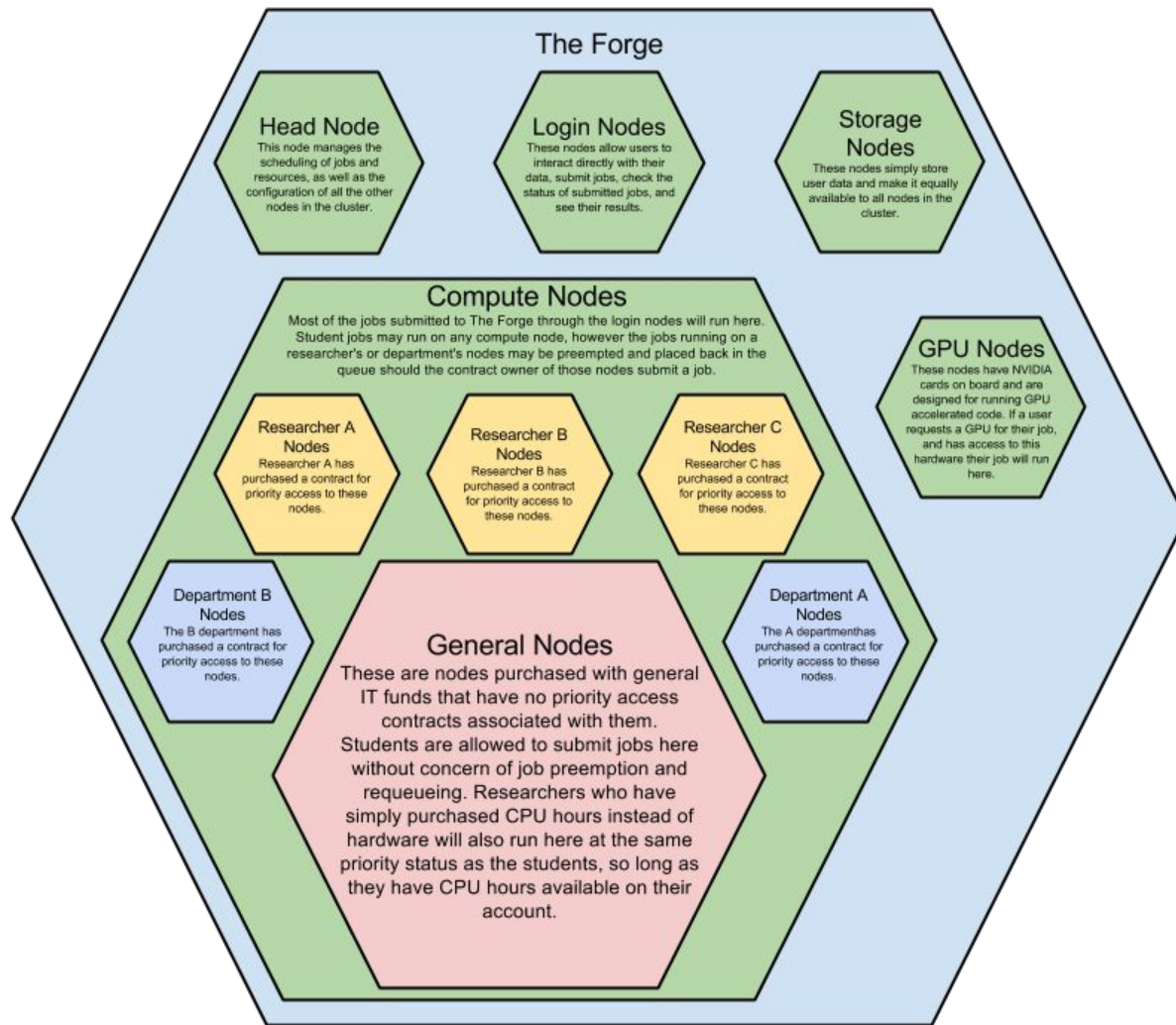
Not all software is created equal. Some may come precompiled for you.

Check for a "bin" directory, that's where treasures lie

If you need help, give us as much information as you can



And now what you've all been waiting for...





Account Creation

- You get access to what you (or your group) paid for
- We want to have accurate record keeping for everyone
- Fill out the form at the link below

Reference:

<https://crc.ku.edu/access>



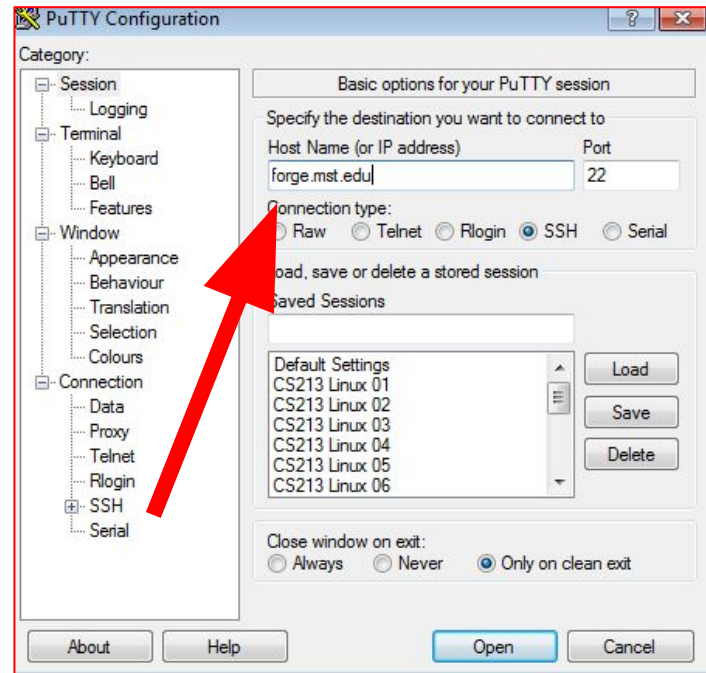
SSH cont'd

Windows: **Open PuTTY**

Host Name (or IP address): <KU Online ID>@hpc.crc.ku.edu

Mac and Linux: **Open Terminal**

```
ssh <KU Online ID>@hpc.crc.ku.edu
```





Cluster Specific Commands

module avail, module load, module list

#MSUB All commands within the job file

msub Command on the cluster to submit a job file

msub -I Start an interactive shell on a compute node

showq / qstat How to check the status of your job



Queues

- Each group that purchased hardware gets at least one queue
- Queue covers all hardware your group purchased, requesting specialty nodes done with flags in submit script
- Special 'sixhour' queue exists for submitting jobs to anywhere on the cluster. Runs at a lower priority and can only run for ≤ 6 hours.
 - `#MSUB -q sixhour`
- `showq -w class=<queue_name>` will show you all the jobs running in a given queue
- Your default queue is shown on login and by running `mystats`

Reference:

<https://crc.ku.edu/using-hpc#Queues>



Sample Submit Script

```
#!/bin/bash
#MSUB -N MyJob
#MSUB -l procs=48,pmem=48m,walltime=05:00
#MSUB -m abe
#MSUB -M crchelp@ku.edu
#MSUB -j oe
```

```
module load compiler/gcc/6.3
module load openmpi/2.0
```

```
mpirun $HOME/myprogram.out
```

Reference:

<https://crc.ku.edu/using-hpc#Script>



Viewpoint and RemoteViz

Moab VIEWPOINT



Recap & useful links

<https://crc.ku.edu/access>

<https://crc.ku.edu/using-hpc>

<http://docs.adaptivecomputing.com/9-1-0/MWM/help.htm#topics/moabWorkloadManager/topics/moabCommands/commands/msub.html>



Questions?

Find these slides at: <https://goo.gl/WC4qVx>

Email us at crchelp@ku.edu