

High Performance Research Computing

A Resource for Research and Discovery



TEXAS A&M
UNIVERSITY.

HPRC Short Course

Introduction to Linux



DIVISION OF RESEARCH
TEXAS A & M UNIVERSITY



Texas A&M University

High Performance Research Computing – <https://hprc.tamu.edu>

For Assistance...

Website: hprc.tamu.edu
Email: help@hprc.tamu.edu
Telephone: (979) 845-0219
Visit us in person: Henderson Hall, Room 114A

Help us, help you -- we need more info

- Which Cluster
- UserID/NetID
- Job ID(s) if any
- Location of your job files, input/output files
- Application used & module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem

Course Outline

1. Accessing the System - Secure shell
2. Directories
3. Gedit - An easy text editor
4. File Manipulation - Copy, rename/move & remove
5. Passing output & Redirection
6. The Linux file system
7. Environment Variables & Search Path
8. Basic Scripting

Jimmy John's will be provided for lunch at ~12:30pm

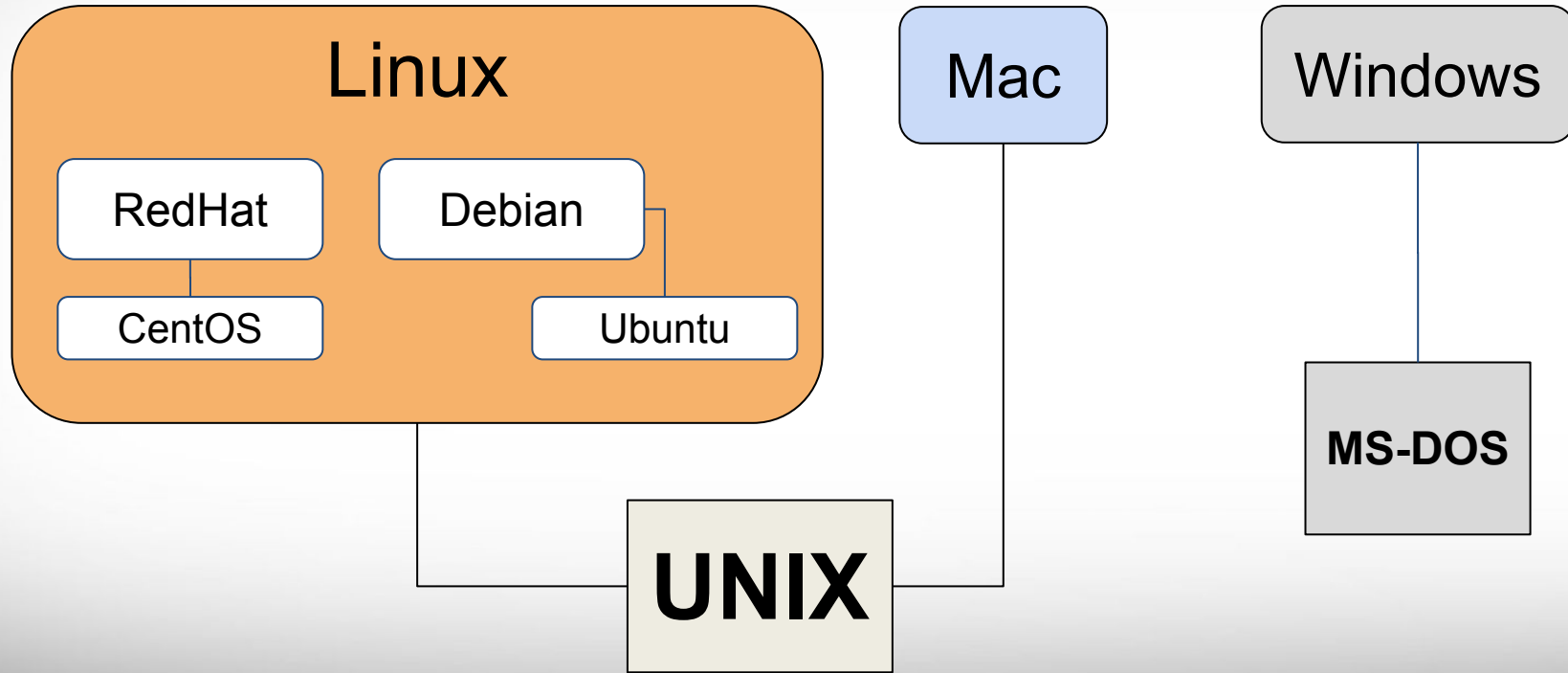
Logging in to the system

- SSH (secure shell)
 - The only program allowed for remote access; encrypted communication; freely available for Linux/Unix and Mac OS X hosts;
- For Microsoft Windows PCs, use *MobaXterm*
 - <https://hprc.tamu.edu/wiki/HPRC:MobaXterm>
 - You are able to view images and use GUI applications with MobaXterm
 - or *PuTTY*
 - https://hprc.tamu.edu/wiki/HPRC:Access#Using_PuTTY
 - You can not view images or use GUI applications with PuTTY

Your Login Password

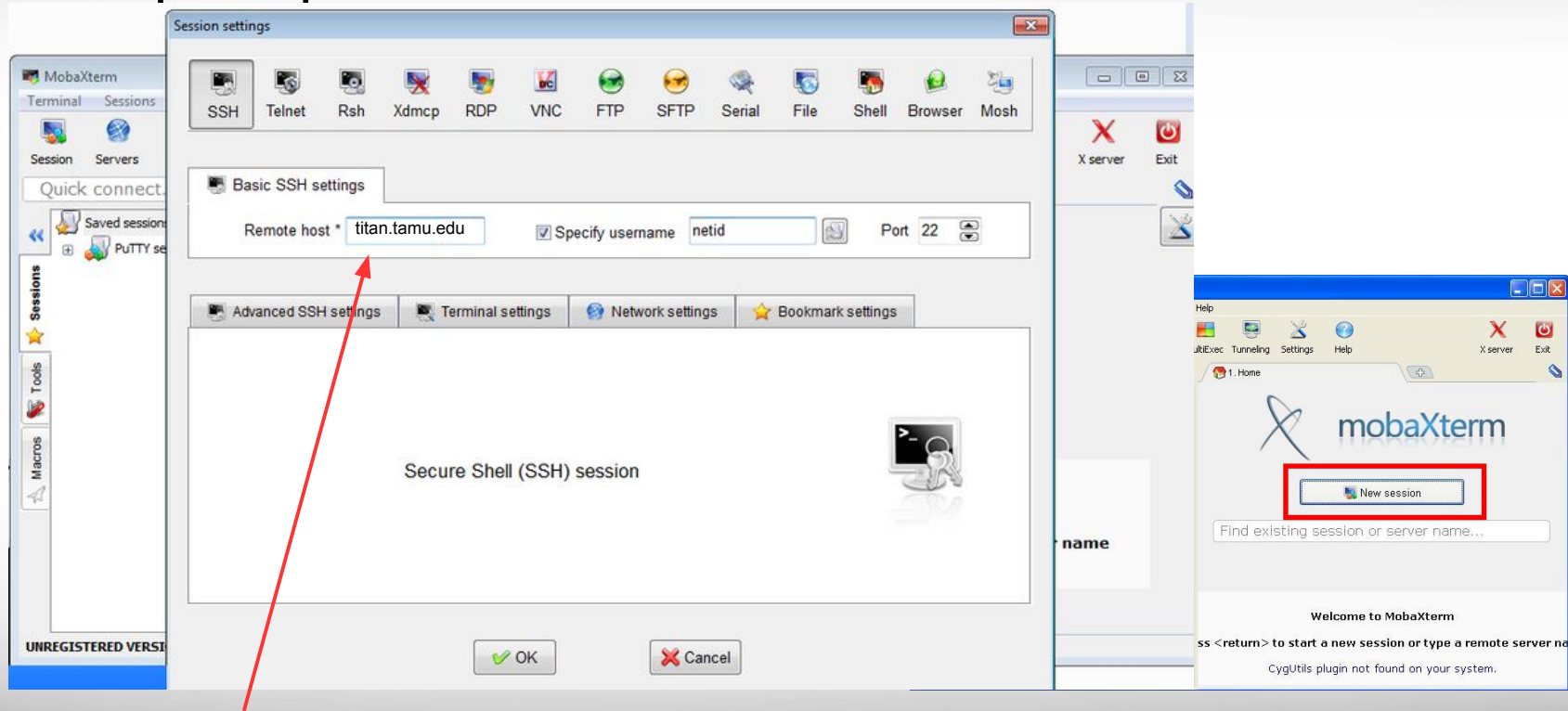
- Both State of Texas law and TAMU regulations prohibit the sharing and/or illegal use of computer passwords and accounts;
- Don't write down passwords;
- Don't choose easy to guess/crack passwords;
- Change passwords frequently

Common Operating Systems (OS)



Using SSH - MobaXterm (on Windows)

<https://hprc.tamu.edu/wiki/HPRC:MobaXterm>



Use **titan.tamu.edu** as Remote host name.

Using SSH (on a Linux Client)

```
ssh -X NetID@titan.tamu.edu
```

You may see something like this the first time you connect to the remote machine from your local machine:

```
Host key not found from the list of known hosts.  
Are you sure you want to continue connecting (yes/no)?
```

Type yes, hit enter and you will then see the following:

```
Host 'titan.tamu.edu' added to the list of known hosts.  
NetID@titan.tamu.edu's password:
```

Mac users may need to use `ssh -Y` to enable X11 so you can view images and use GUI software

```
ssh -Y NetID@titan.tamu.edu
```

The host `titan.tamu.edu` is for this Linux class only and is not an HPRC cluster.
To use the HPRC clusters, you must apply for an HPRC account.

UNIX Terminal Attributes

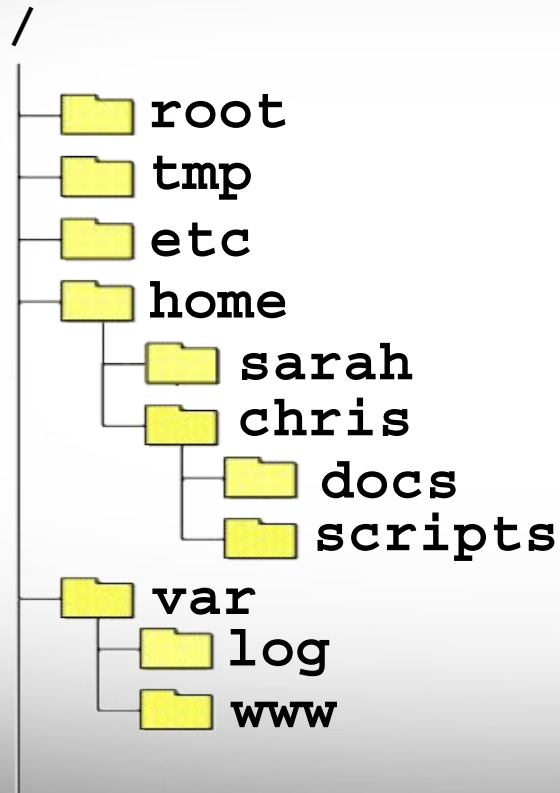
File and directory names are colored
based on their attributes such as permissions and extension

```
AAF -> AAF.py
AAF.py
aaf_tip.py
data.gz
image.jpg
phylip_src
phylokmer
README
run_aaf.sh
```

TURQUOISE
GREEN
RED
PURPLE
BLUE
WHITE

Symbolic link
Executable file
Compressed files
Image files
Directories
Text files

Finding your way around the Linux directory structure



```
/
/root
/tmp
/etc
/home
/home/sarah
/home/chris
/home/chris/docs
/home/chris/scripts
/var
/var/log
/var/www
```

Where Am I?

pwd command (print work directory)

Linux commands in green for you to type

```
pwd
```

command output in blue

```
/home/user_NetID
```

list contents of your pwd

```
ls
```

Linux Commands Have Options

Leave a space between the command and the options

double dash means there is a single option which is usually a descriptive word

```
ls --all
```

--all show all files including
hidden files which begin with .

single dash means each character is an option

```
ls -a -l
```

-a show all files including hidden
-l show file details

most options can be combined behind one dash

```
ls -al
```

-a show all files including hidden
-l show file details

. current working directory
.. parent directory

Search for Linux Commands Options

Search the manual page for the Linux command `ls`

`man ls`

Page up

Move up one page

Page down

Move down one page

Spacebar

Move down one page

Mouse scroll wheel

Move up and down

`/all`

search the man page for the text 'all'

`n`

search forward for next found match

`N`

search backwards next found match

`g`

go to first line

`G`

Go to last line

`q`

quit

File and directory names

Commonly used:

A-Z

a-z

0-9

.

- dash

_ underscore

Avoid using:

spaces

() parenthesis

" ' quotes

? Question mark

\$ Dollar sign

* Asterisk

\ back slash

/ forward slash

: colon

- Avoid spaces.
- File and directory names are case sensitive
- Avoid spaces in the file name ("my data file.txt" vs "my_data_file.txt").
- Avoid creating files on your Windows computer and copying to UNIX especially with spaces in the file name

Common Directory Commands

mkdir command to make a new directory:

```
mkdir my_dir
```

cd to change to another directory:

```
cd my_dir
```

rmdir to remove an empty directory:

```
rmdir my_dir
```

Changing Directories: the `cd` cmd

- Return to your home directory

```
cd
cd ~
cd ~/
```

- To switch to the parent directory of the current directory:

```
cd ..
```

- Return to previous `pwd`

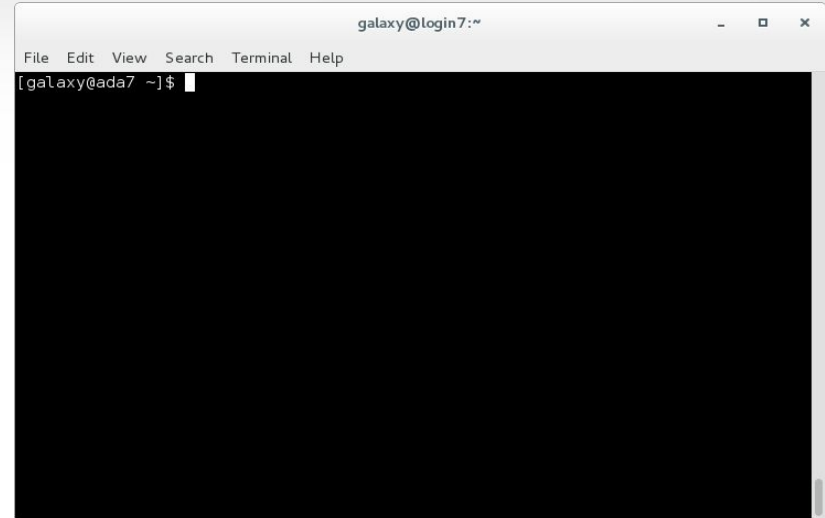
```
cd -
```

```
cd
mkdir temp
mkdir temp/hg19
cd temp
pwd
cd hg19
pwd
cd ../..
pwd
cd -
pwd
cd ..
pwd
cd
pwd
```


Clear Contents on Screen

Type `clear` command to clear screen contents.

`clear`



You can still scroll up in your terminal to see past contents

You can also use `Ctrl+I` to clear contents

Absolute vs. Relative Path

```
/
/root
/tmp
/etc
/home
/home/sarah
/home/chris/project
/home/chris/docs/README
/var
/var/log
/var/www
```

If you are in the **project** directory

```
pwd
```

```
/home/chris/project
```

The relative path to the README
file is `../docs/README`

```
ls ../docs/README
```

The absolute path to the README
file `/home/chris/docs/README`

```
ls /home/chris/docs/README
```

The *Gedit* Text Editor

From the Linux terminal command line enter this command to start gedit and edit a file called my_favorite_foods.txt

```
gedit my_favorite_foods.txt &
```

The ‘&’ will detach gedit from the terminal so you can continue to use the terminal

Recommendations for naming files:

- Use all lowercase characters

- Separate words with an underscore

- Make the filename concise and very descriptive of the file contents even if the name seems long

Some common file extensions are :

- .pl a Perl script

- .py a Python script

- .gz a file that has been compressed (zipped) to reduce file size (.zip)

- .txt a generic text file

- .tsv tab separated values (columns are separated by a tab)

- .csv comma separated values

- .jar a Java Archive file

Editing an ASCII file

- There are many editors available under Linux.
- Text mode
 - nano (simple)
 - vi or vim (more advanced)
 - emacs (more advanced)
- Graphic mode (require X11)
 - gedit
 - xemacs / gvim
- Be aware that a text file edited under Windows editors will most likely add CRLF characters. Use `dos2unix` to convert a DOS/Windows edited text file to UNIX format.

Use Tab to Complete a File Name

Type the first few characters of the file name

```
ls my
```

then hit the **tab key** to autocomplete the file name

```
ls my_favorite_foods.txt
```

then hit enter to see the command results

If the tab key did not complete the file name then either the file does not exist or there are two or more files that begin with the same characters in which case you need to hit tab twice then type a few more characters and hit tab again to complete.

Count the Lines in a File

```
wc my_favorite_foods.txt
```

```
29 109 876 my_favorite_foods.txt
```

What does the output mean?

Use the man page for `wc` to find out.

How do you just print the newline counts?

Download a File from the Web to Your pwd



hg19 gzipped fasta file download



All

Maps

News

Videos

Images

More ▼

Search tools

About 4,760 results (0.60 seconds)

Click this link



[Index of /goldenpath/hg19/chromosomes](#)

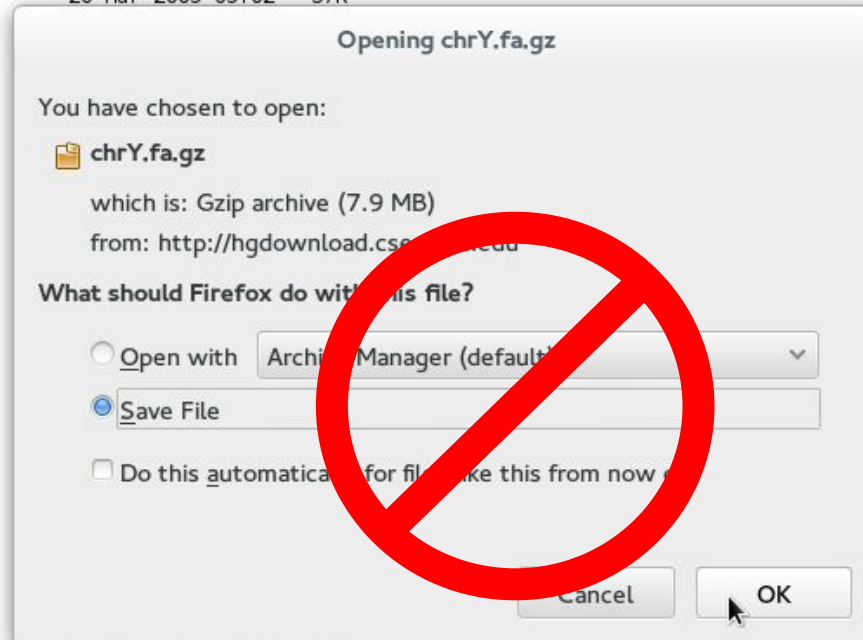
hgdownload.cse.ucsc.edu/.../hg19/ch... ▼ University of California, Santa Cruz ▼

Files included in this directory: - chr*.fa.gz: **compressed FASTA** sequence of ... we recommend that you use ftp rather than **downloading** the **files** via our website.

Don't Left Click and download a file to your desktop

[chrUn_gl000220.fa.gz](#)
[chrUn_gl000221.fa.gz](#)
[chrUn_gl000222.fa.gz](#)
[chrUn_gl000223.fa.gz](#)
[chrUn_gl000224.fa.gz](#)
[chrUn_gl000225.fa.gz](#)
[chrUn_gl000226.fa.gz](#)
[chrUn_gl000227.fa.gz](#)
[chrUn_gl000228.fa.gz](#)
[chrUn_gl000229.fa.gz](#)
[chrUn_gl000230.fa.gz](#)
[chrUn_gl000231.fa.gz](#)
[chrUn_gl000232.fa.gz](#)
[chrUn_gl000233.fa.gz](#)
[chrUn_gl000234.fa.gz](#)
[chrUn_gl000235.fa.gz](#)
[chrUn_gl000236.fa.gz](#)
[chrUn_gl000237.fa.gz](#)
[chrUn_gl000238.fa.gz](#)
[chrUn_gl000239.fa.gz](#)
[chrUn_gl000240.fa.gz](#)
[chrUn_gl000241.fa.gz](#)
[chrUn_gl000242.fa.gz](#)
[chrUn_gl000243.fa.gz](#)
[chrUn_gl000244.fa.gz](#)
[chrUn_gl000245.fa.gz](#)
[chrUn_gl000246.fa.gz](#)
[chrUn_gl000247.fa.gz](#)
[chrUn_gl000248.fa.gz](#)
[chrUn_gl000249.fa.gz](#)
[chrX.fa.gz](#)
[chrY.fa.gz](#)
[md5sum.txt](#)

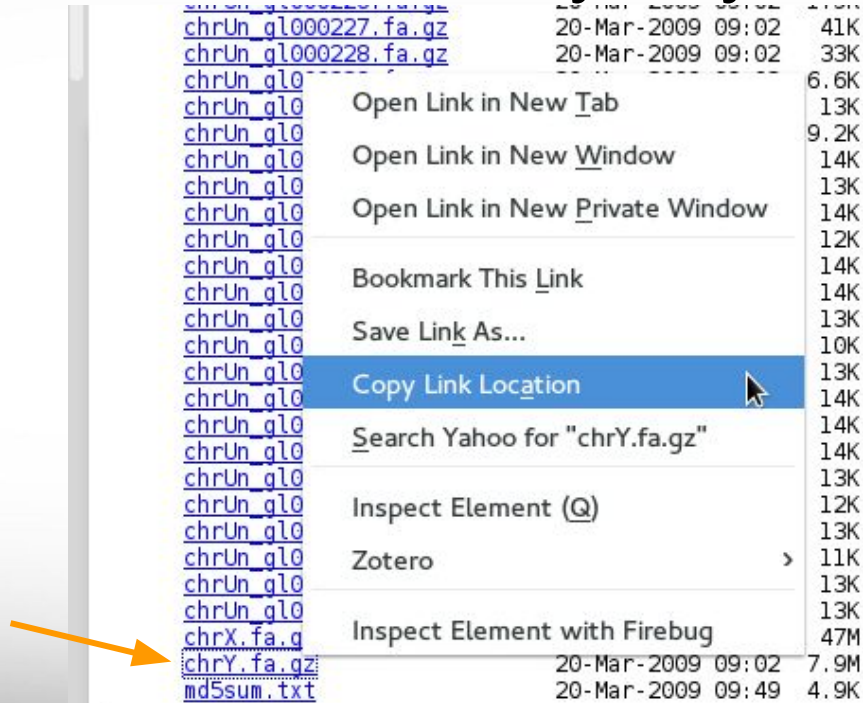
20-Mar-2009	09:02	51K
20-Mar-2009	09:02	50K
20-Mar-2009	09:02	60K
20-Mar-2009	09:02	57K



go to the
bottom of the
web page

20-Mar-2009	09:00	47M
20-Mar-2009	09:02	7.9M
20-Mar-2009	09:49	4.9K

Right Click and Copy the URL so you can download it directly to your Linux pwd



Copying a File Directly to a LINUX Directory

```
cd ~/temp/hg19
```

Use the **wget** command to get a file from a URL

Type **wget** then a space then right click and paste the URL

```
wget http://hgdownload.cse.ucsc.edu/goldenpath/hg19/chromosomes/chrY.fa.gz
```

```
wget http://hgdownload.cse.ucsc.edu/goldenpath/hg19/chromosomes/md5sum.txt
```

List the directory contents to see the file with details (-l)
and human readable file sizes (-h)

```
ls -lh
```

Copying and Renaming Files

Use the tab key to help prevent typos when typing filenames and directories

1a. `cp ch` (then hit tab)

UNIX will complete the file name for you

1b. `cp chrY.fa.gz`

Make a copy of the `chrY.fa.gz` file called `chrY_copy.fa.gz`

1c. `cp chrY.fa.gz chrY_copy.fa.gz`

Rename the `chrY_copy.fa.gz` file to `chrY_hg19.fa.gz`

```
mv chrY_copy.fa.gz chrY_hg19.fa.gz
```

Deleting Files: The `rm` Command

```
rm [options] [file_name]
```

- Commonly used options with the `rm` command
 - i prompt user before any deletion
- Use the wildcard `*` to list all files ending with `gz`

Exercise:

```
ls *gz  
  
rm -i chrY.fa.gz  
  
ls
```

Displaying File Contents

- Check the file size before attempting to open with a text editor

```
ls -lh md5sum.txt
```

```
cat md5sum.txt
```

- `cat` prints all the contents of a file(s) to the screen.
- The `more` command, and its improved version `less`, display a text file one page (screen) at a time.
 - Hit space bar for next page; `less` search function is similar to man pages
 - Type `q` to quit
 - Use `zmore` or `zless` for compressed files (.gz)

```
more md5sum.txt  
less md5sum.txt  
zmore chrY_hg19.fa.gz
```

Displaying File Contents

- Use `head` and `tail` commands to see first and last 10 lines of a file respectively

```
head md5sum.txt
```

```
tail md5sum.txt
```

- `head` and `tail` are not for compressed files (.gz)
- There is not a `zhead` or `ztail` command
- Use `zcat` together with the `head` or `tail` command
- Use the pipe `|` to send output of first command to the second command

```
zcat chrY_hg19.fa.gz | head
```

```
zcat chrY_hg19.fa.gz | tail
```



grep – Searching for Pattern(s) in Files

```
grep [options] PATTERN [files ...]
```

```
grep chrX md5sum.txt
```

the -i option is to ignore case

```
grep -i chrX md5sum.txt
```

Count the number of lines that match pattern

```
grep -c random md5sum.txt
```

grep – Searching For Pattern(s) in Files

Search multiple matches

```
grep -e chrX -e chrY md5sum.txt
```

Exclude a pattern; show non-matching lines

```
grep -v random md5sum.txt
```

Use **zgrep** for compressed files (.gz)

```
zgrep chr chrY_hg19.fa.gz
```


Redirection Operators

<	redirects input	<code>bsub < job_script.sh</code>
>	redirects output	<code>command > out.txt</code>
>>	appends output	<code>command >> out.txt</code>
<<	input from <i>here document</i> (search the web for examples)	

2>	redirects error	<code>command 2> error.txt</code>
&>	redirects output and error	<code>command &> out-error.txt</code>
>&	redirects output and error	<code>command >& out-error.txt</code>
2>&1	redirects error to where output is going	
1>&2	redirects output to where error is going	

Passing the Output of Linux Commands

Use the pipe character `|` to send results to another command

Search for the string 'chrX' in the output of md5sum.txt

```
cat md5sum.txt | grep chrX
```

Use the standard output redirect operator `>` to **create** a new file

```
grep chrX md5sum.txt > chr_xy.txt
```

Use the standard output redirect operator `>>` to **append** to a file

```
grep chrY md5sum.txt >> chr_xy.txt
```

History of Your Commands

- Your commands are saved to a file in your home directory (`.bash_history`)
- You can use the up/down arrows to scroll through your previous commands
- Type `history` to see your previously entered commands

```
history
```

```
history | tail
```

See the last 10 commands

- Search your history commands using `|` and `grep`

```
history | grep wget
```

Types of File: the `file` cmd

```
file [name]
```

Displays a brief description of the contents or other information for a file or related object.

```
file md5sum.txt
```

```
md5sum.txt: ASCII text
```

```
file chrY_hg19.fa.gz
```

```
chrY_hg19.fa.gz: gzip compressed data,  
was "chrY.fa", last modified ...
```

CRLF Line Terminators

Windows editors such as Notepad will add hidden Carriage Return Line Feed (CRLF) characters that will cause problems with many applications

```
cd ~/intro_to_linux/scripts/
```

```
file DOS_script.sh
```

```
DOS_script.sh: ASCII English text, with CRLF line terminators
```

```
dos2unix DOS_script.sh  
file DOS_script.sh
```

```
DOS_script.sh: ASCII English text
```

The Linux Filesystem

File Attributes: A look with `ls`

```
[user_NetID@titan ~]$ ls -l
```

```
total 37216
```

drwx-----	7	user_NetID	user_NetID	121	Sep	9	10:41	abaqus_files
-rw-----	1	user_NetID	user_NetID	2252	Aug	24	10:47	fluent-unique.txt
-rw-----	1	user_NetID	user_NetID	13393007	Aug	24	10:40	fluent-use1.txt
-rw-----	1	user_NetID	user_NetID	533	Aug	24	11:23	fluent.users
drwxr-xr-x	3	user_NetID	user_NetID	17	May	7	16:56	man
-rw-----	1	user_NetID	user_NetID	24627200	Sep	9	10:49	myHomeDir.tar
lrwxrwxrwx	1	root	root	21	May	28	16:11	README -> /usr/local/etc/README
-rwx-----	1	user_NetID	user_NetID	162	Sep	7	12:20	spiros-ex1.bash
-rwx--x--x	1	user_NetID	user_NetID	82	Aug	24	10:51	split.pl
drwxr-xr-x	2	user_NetID	user_NetID	6	May	5	11:32	verifyOLD

file permissions

hard link count

user name

group name

file size in bytes

file modification date

file modification time

file name

Directory Permissions

```
drwx----- 7 user_NetID staff 121 Sep 9 10:41 abaqus_files
```

↑ ↑ ↑

permissions user group

- The meanings of the permission bits for a directory are slightly different than for regular files:
 - `r` permission means the user can list the directory's contents
 - `w` permission means the user can add or delete files from the directory
 - `x` permission means the user can `cd` into the directory; it also means the user can execute programs stored in it
- Notice that if the file is a directory, the leading bit before the permissions is set to `d`, indicating directory.

File Ownership and Permissions

```
-rwx--x--x  1 user NetID staff      82 Aug 24 10:51 split.pl
```

↑
permissions

↑
user

↑
group

- <u>r</u> <u>w</u> <u>x</u>	- - <u>x</u>	- - <u>x</u>
user	group	other

- There are 3 sets of permissions for each file
 - 1st set - user (the owner)
 - 2nd set - group (to which file owner belongs)
 - 3rd set - other (all other users)
- The r indicates read permission
- The w indicates write permission
- The x indicates execute permission

Changing Attributes: The `chmod` cmd

```
chmod [options] [permission mode] [target_file]
```

```
cd ~/temp/hg19
```

```
chmod 755 chr_xy.txt ( the permissions will be set to -rwxr-xr-x )
```

```
chmod o-x chr_xy.txt ( the permissions will change to -rwxr-xr-- )
```

```
chmod ug-x chr_xy.txt ( the permissions will change to -rw-r--r-- )
```

```
chmod g+w chr_xy.txt ( the permissions will change to -rw-rw-r-- )
```

u = user

r = read

g = group

w = write

o = other

x = execute

-x = remove executable permissions

+x = enable executable permissions

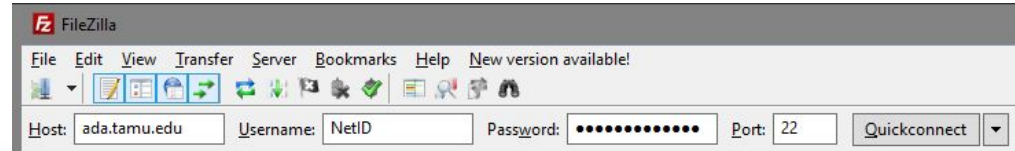
Transfer Data From Windows Host to Linux Host

On a Windows system, there are several applications to transfer files between remote machines:

- MobaXterm
- WinSCP
- FileZilla*



*personal favorite



Data transfer, including management practices such as **tar & compression** will be covered in our Data Management Practices short course

https://hprc.tamu.edu/training/data_management.html

Bash Environment Variables

Use all caps for Bash Environment variable. **A-Z** 0-9 _
Use lowercase for the variables that you create. **a-z** 0-9 _

HOME Pathname of current user's home directory
PATH The search path for commands.

Use the `echo` command to see the contents of a variable

```
echo $HOME
```

```
/home/user_NetID
```

list contents of your \$HOME directory

```
ls $HOME
```

The Search Path

- The shell uses the PATH environment variable to locate commands typed at the command line
- The value of PATH is a colon separated list of full directory names.
- The PATH is searched from left to right. If the command is not found in any of the listed directories, the shell returns an error message
- If multiple commands with the same name exist in more than one location, the first instance found according to the PATH variable will be executed.

```
echo $PATH
```

```
PATH=/opt/TurboVNC/bin:/software/tamusc/local/bin:  
/software/lsf/9.1/linux2.6-glibc2.3-x86_64/bin:  
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:  
/sbin:/usr/lpp/mmfs/bin:/opt/ibutils/bin:/home/user_NetID/bin
```

add a directory to the PATH for the current Linux session

```
export PATH=$PATH:/home/user_NetID/bin
```

Shell Scripts

- A shell script is a file that contains one or more UNIX commands that can be run as a single batch of commands

Shell Script Exercise

create a shell script

```
gedit my_script.sh &
```

make your shell script executable

```
chmod 755 ./my_script.sh
```

run your shell script


```
./my_script.sh
```

run the shell script and send stdout to file

```
./my_script.sh > output.txt
```

send stdout and stderr to files

```
./my_script.sh > output.txt 2> err.txt
```



```
#!/bin/bash
# HPRC shell script exercise

my_city='College Station'

echo "Howdy $my_city" > my_city.txt
echo "Howdy $my_city" >> my_city.txt

mkdir outdir
mv my_city.txt outdir/
cd outdir
cat my_city.txt
cd ..

cd newdir
```

```
exit
```

exit the terminal session

can also use Ctrl+d to detach session

References

Here are some slides from TACC on the similar subject.

- Linux/Unix Basics for HPC: October 9, 2014 (with video) [TACC]
 - <https://portal.tacc.utexas.edu/-/linux-unix-basics-for-hpc>
- Express Linux Tutorial: Learn Basic Commands in an Hour [TACC]
 - https://portal.tacc.utexas.edu/c/document_library/get_file?uuid=ed6c16e9-bcbc-4b70-9311-5273b09508b8&groupId=13601