Intro to Unix

Michael Sandoval

HPC Engineer - User Assistance Group

Oak Ridge Leadership Computing Facility (OLCF)

Oak Ridge National Laboratory (ORNL)

December 13, 2022

Overview

- This presentation will focus on using basic Unix and Linux commands in an HPC environment
  - Assumes you're using a 'remote' system, ie: that you are

  logged in via terminal, PuTTY, Powershell, etc.

- Cover the basics
  - The terminal window, command line
  - The filesystem structure & how to navigate it
  - Common commands to create, delete, edit, move, copy, etc., directories and files.

- Follow along here:
  - https://github.com/olcf/foundational_hpc_skills/blob/master/intro_to_unix/README.md

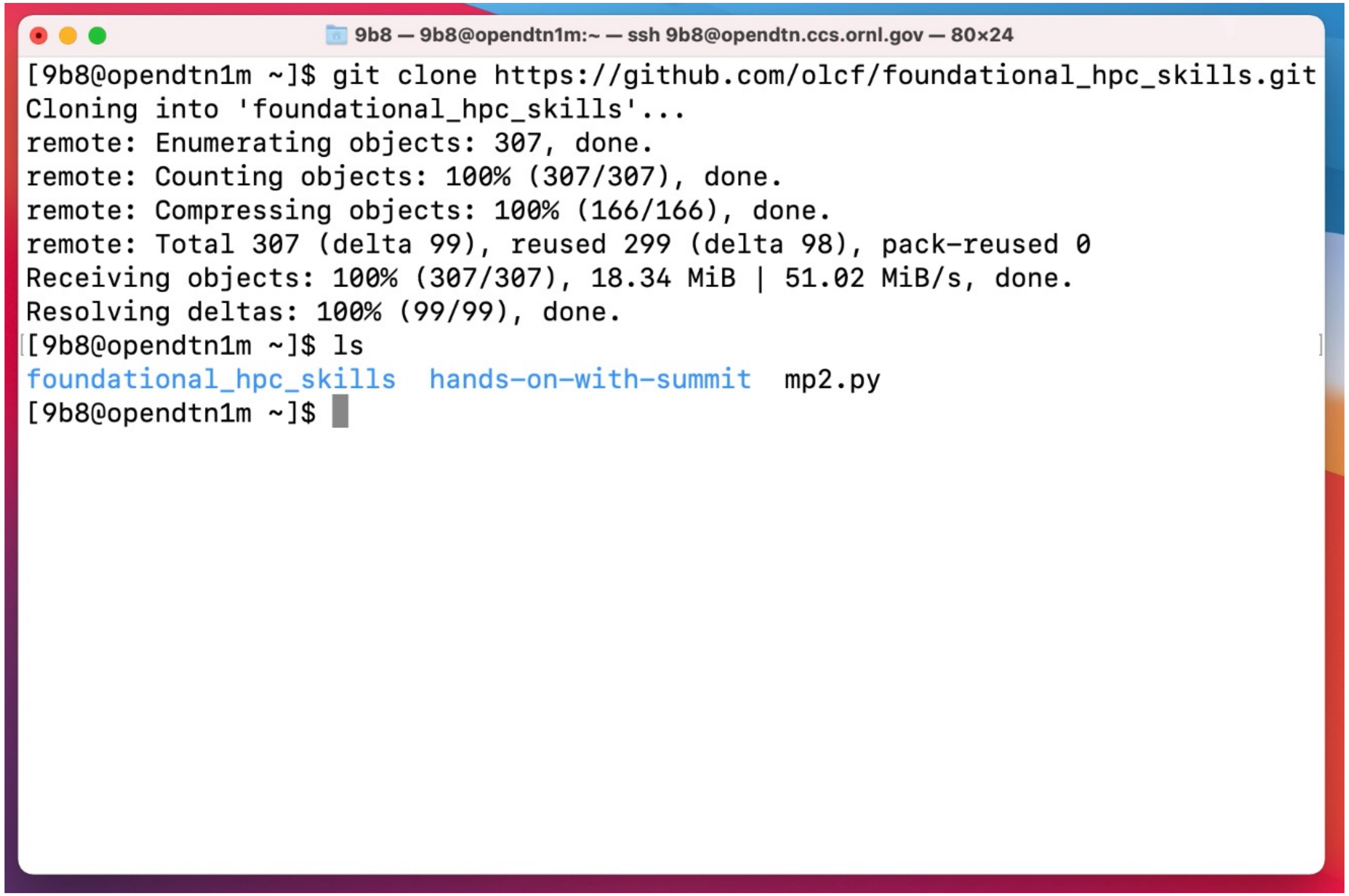
- Google is your friend!

First things first: Login and Clone!

- **Login** to the remote machine using the username and password you used when registering for the crash course.
  - Look for email from "OLCF Accounts"
  - Reset password here: https://xcams.ornl.gov/xcams/
- **Recall, the syntax is:** username@opendtn.ccs.ornl.gov
- **Type** the following and hit **enter:**

```
git clone https://github.com/olcf/foundational_hpc_skills.git
```

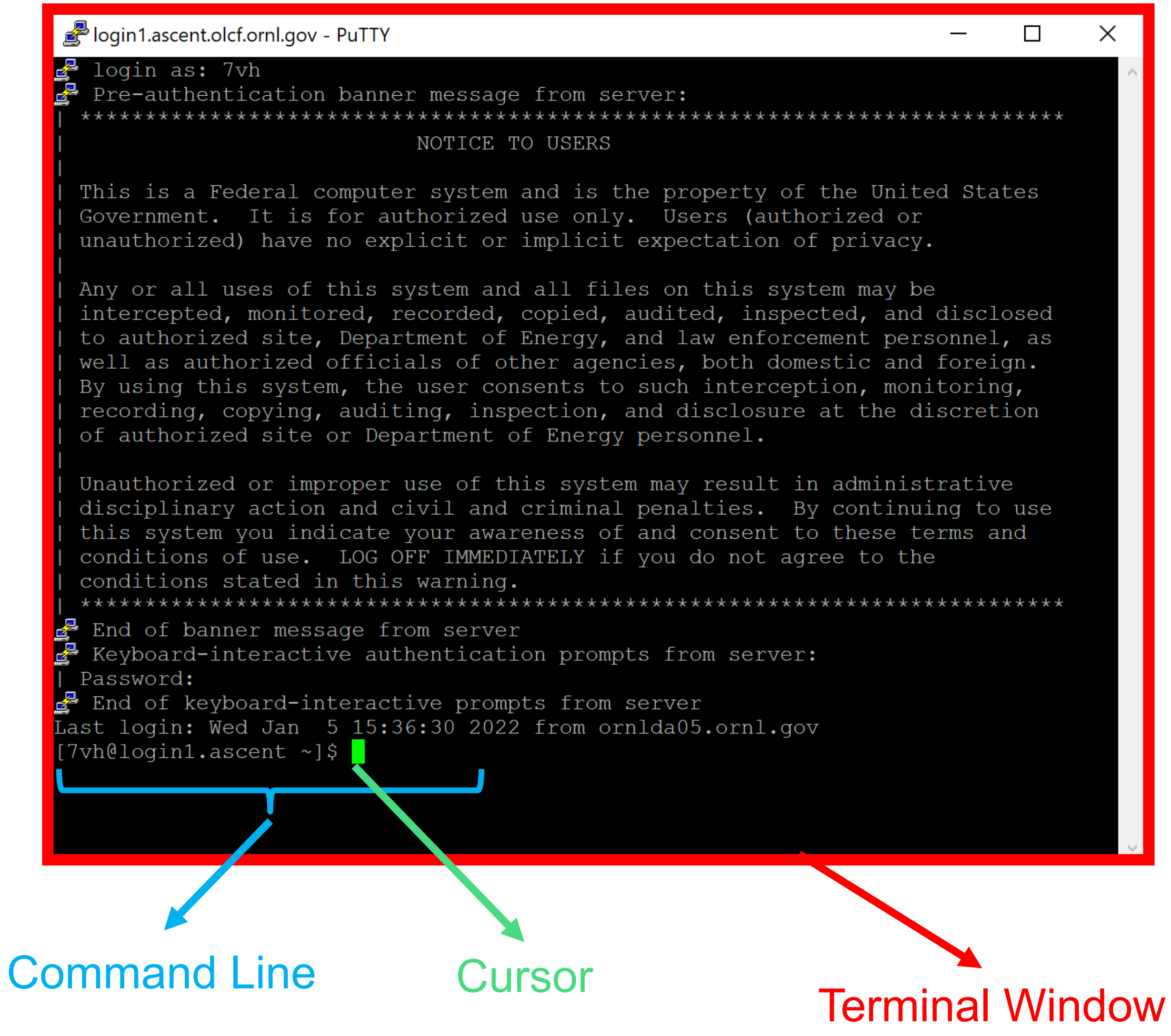
Connecting and Cloning


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m ~]$ git clone https://github.com/olcf/foundational_hpc_skills.git
Cloning into 'foundational_hpc_skills'...
remote: Enumerating objects: 307, done.
remote: Counting objects: 100% (307/307), done.
remote: Compressing objects: 100% (166/166), done.
remote: Total 307 (delta 99), reused 299 (delta 98), pack-reused 0
Receiving objects: 100% (307/307), 18.34 MiB | 51.02 MiB/s, done.
Resolving deltas: 100% (99/99), done.
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$
```

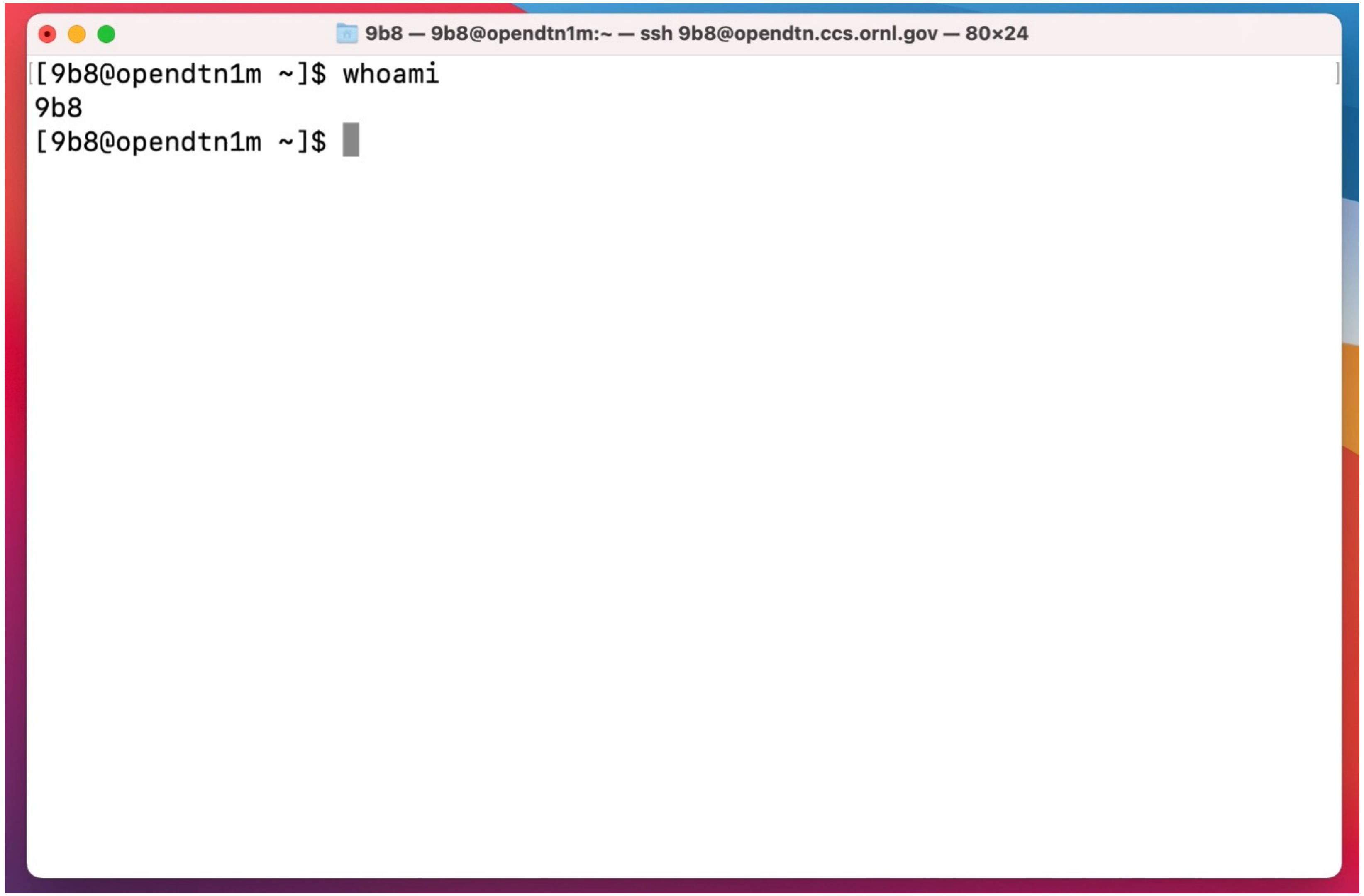
The Command Line

- Commands are typed as text on the command line and executed by pressing "enter"
- Many examples show the "$" symbol from the command line before the actual command; you don't need to type this symbol
- The command line often contains login and username info.
- Try typing `$ whoami` and pressing enter



```
login as: 7vh
Pre-authentication banner message from server:
***************************************************************************
|                         NOTICE TO USERS
|
| This is a Federal computer system and is the property of the United States
| Government.  It is for authorized use only.  Users (authorized or
| unauthorized) have no explicit or implicit expectation of privacy.
|
| Any or all uses of this system and all files on this system may be
| intercepted, monitored, recorded, copied, audited, inspected, and disclosed
| to authorized site, Department of Energy, and law enforcement personnel, as
| well as authorized officials of other agencies, both domestic and foreign.
| By using this system, the user consents to such interception, monitoring,
| recording, copying, auditing, inspection, and disclosure at the discretion
| of authorized site or Department of Energy personnel.
|
| Unauthorized or improper use of this system may result in administrative
| disciplinary action and civil and criminal penalties.  By continuing to use
| this system you indicate your awareness of and consent to these terms and
| conditions of use.  LOG OFF IMMEDIATELY if you do not agree to the
| conditions stated in this warning.
***************************************************************************
End of banner message from server
Keyboard-interactive authentication prompts from server:
| Password:
End of keyboard-interactive prompts from server
Last login: Wed Jan  5 15:36:30 2022 from ornlda05.ornl.gov
[7vh@login1.ascent ~]$
```

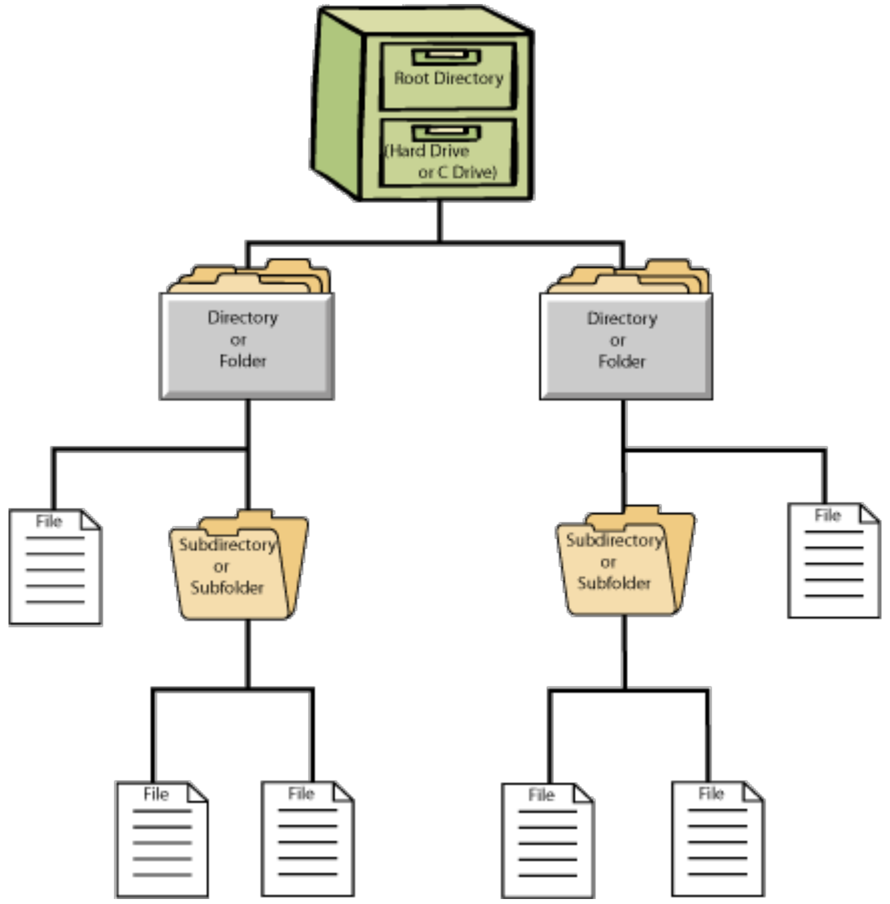
Who am I? (really though)


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m ~]$ whoami
9b8
[9b8@opendtn1m ~]$
```

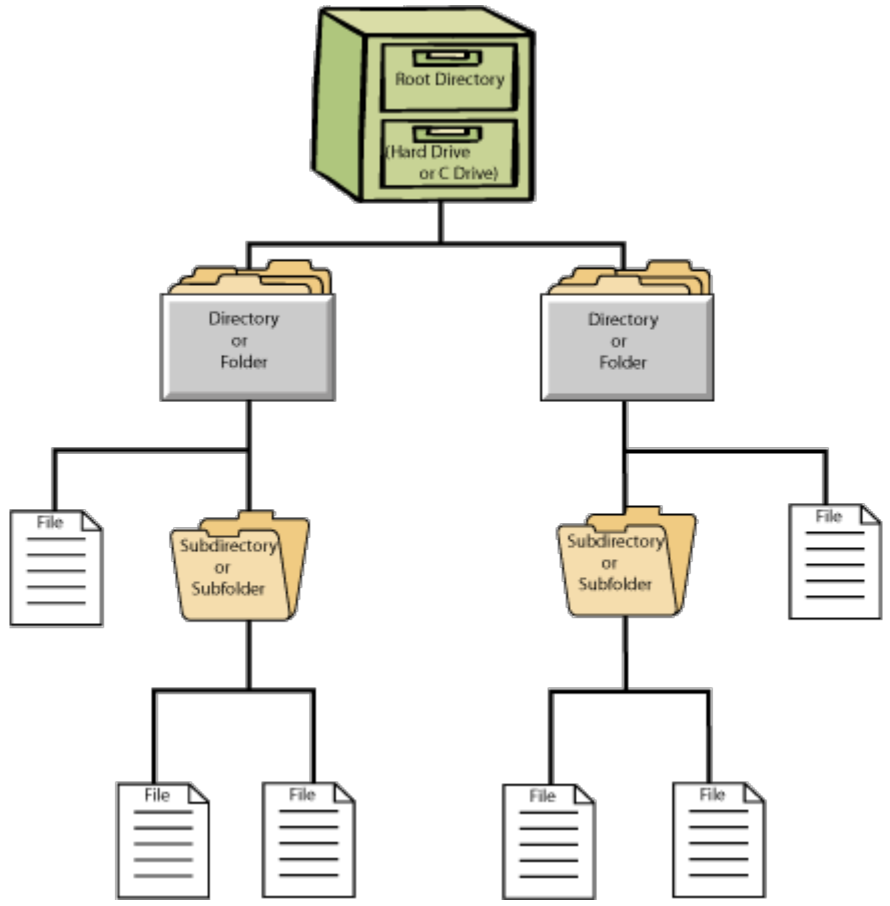
The Filesystem 1

- **File** – A collection of data
  - Plain text, Input/output, a program (executable), an image, etc.
- **Directory** – A logical structure to help organize files (think "folder")
- **Filesystem** – A collection of files and directories
- As a user, you land in your "home" directory when you log in, and can create directories and files there, or move elsewhere to do that.



The Filesystem 2

- Think of the filesystem as a tree. Instead of branches, there are "directories. Instead of leaves, there are "files"
- It starts at /, which is called the "root" directory
- The slash (/) is the directory separator; thus when we move into subdirectories it's used to separate things (i.e. /ccsopen/home/your_username)



If you are ever "lost", use the pwd command. It prints your location.

```
$ pwd
/ccsopen/home/your_username
```

Basic Commands

- Commands are usually abbreviations of words (or a series of words)
  - cp for “Copy”, rm for “Remove”, cd for “change directory”
- Commands tend to follow the following syntax:

```
[command] [option flags] [file/directory/object to act upon]
```

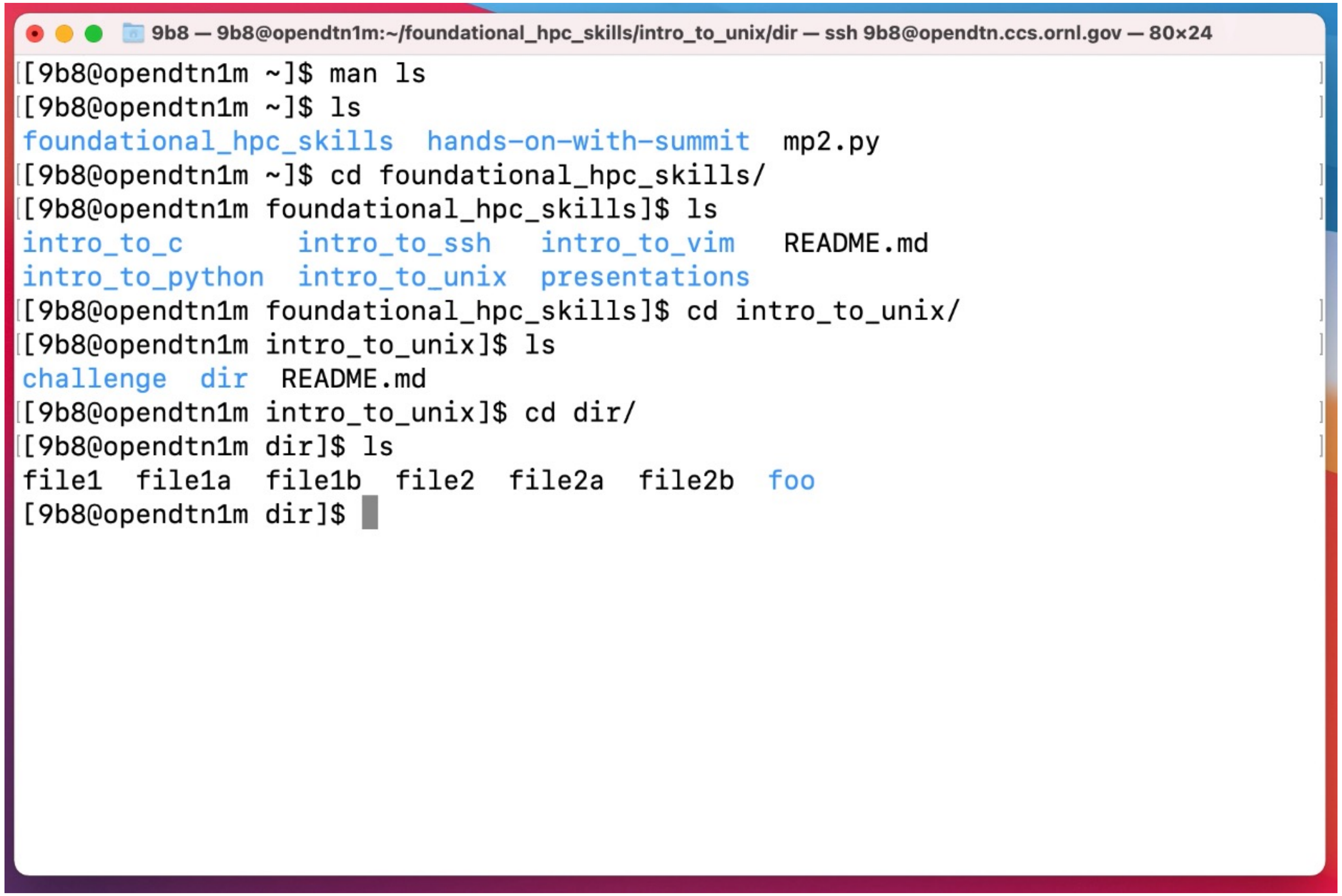
- Almost all commands take various options to control what they do.
- Some help is available via an online manual
  - The **`man`** command
  - Example: Want info about `ls`? Type the following:

```
man ls
```

Basic File/Directory Commands

Command	Description
pwd	**P**rint **w**orking (i.e. current) **d**irectory
man	Display the **Man**ual for a command
whoami	Display the current user's username
mkdir	Create a directory (**M**a**K**e **DIR**ectory)
rmdir	Delete a directory (**R**e**M**ove **DIR**ectory)
cd	**C**hange (into a) **d**irectory
ls	**L**i**s**t files
cp	**C**o**p**y a file
mv	**M**o**v**e a file (also used to rename a file)
rm	**R**e**m**ove (delete) a file
cat	Display the contents (con**cat**enate) a (hopefully text) file
less	Display the contents of a text file in a viewing mode
find	**Find** files by filename
grep	Search for text within files
diff	Identify **diff**erences in the contents of files

“ls” and “cd” In Action 1



9b8 — 9b8@opendtn1m:~/foundational_hpc_skills/intro_to_unix/dir — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m ~]$ man ls
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ cd foundational_hpc_skills/
[9b8@opendtn1m foundational_hpc_skills]$ ls
intro_to_c       intro_to_ssh   intro_to_vim   README.md
intro_to_python  intro_to_unix  presentations
[9b8@opendtn1m foundational_hpc_skills]$ cd intro_to_unix/
[9b8@opendtn1m intro_to_unix]$ ls
challenge  dir  README.md
[9b8@opendtn1m intro_to_unix]$ cd dir/
[9b8@opendtn1m dir]$ ls
file1  file1a  file1b  file2  file2a  file2b  foo
[9b8@opendtn1m dir]$ 
```

Directories

Use `pwd`, `cd`, `mkdir`, and `rmdir` commands to navigate the filesystem and manipulate directories

```
$ pwd
/home/user1

$ mkdir dir1

$ ls
dir1

$ cd dir1

$ pwd
/home/user1/dir1

$ cd ..

$ rmdir dir1
```

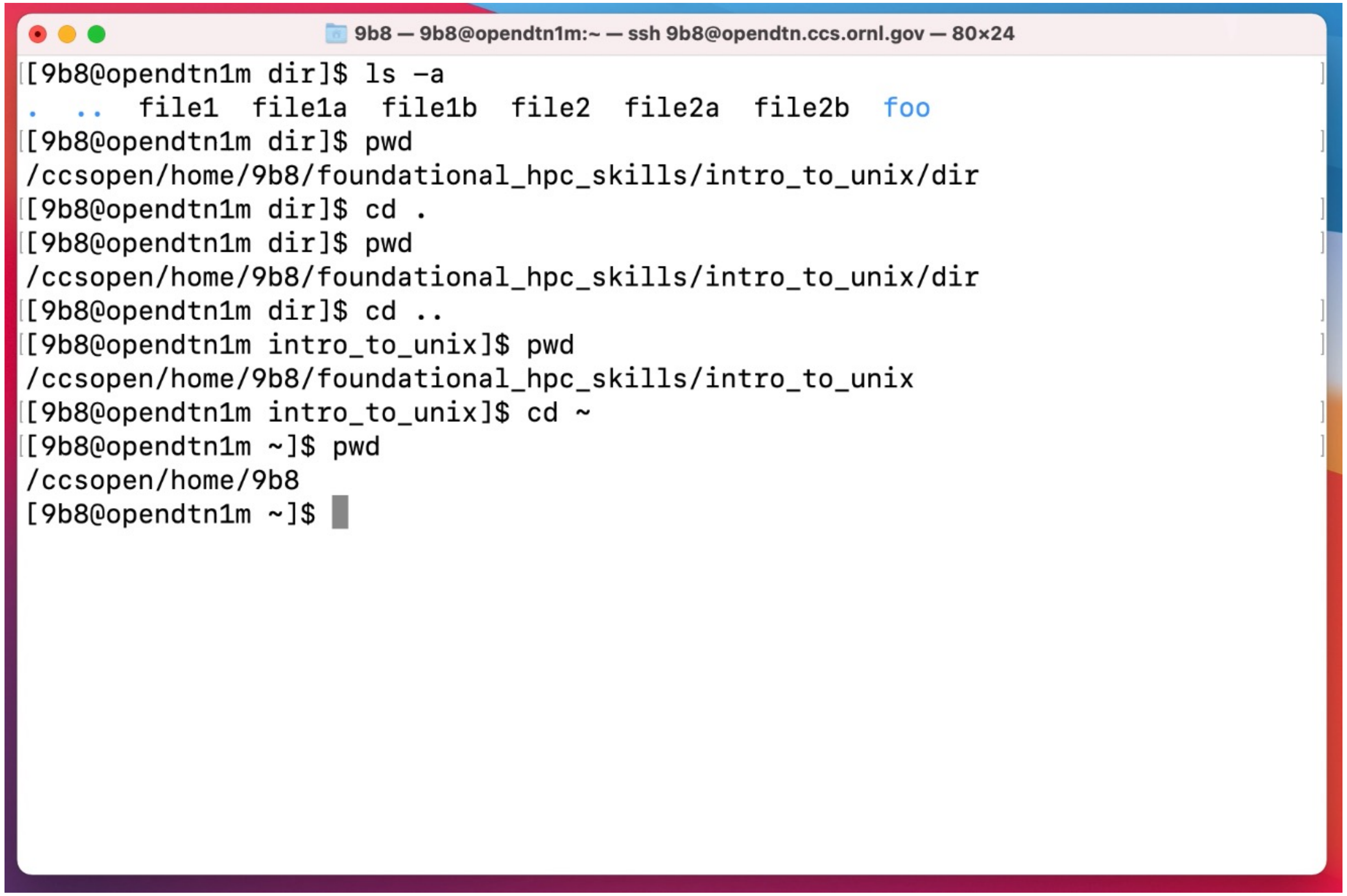
Typing just `cd` will always take you back to home no matter where you are.

Directories must be empty in order to delete them with `rmdir`

Special Directories

- Every directory contains two special directory entries: ”.” and “..”
- . is a reference to the current directory
- .. is a reference to the parent directory (so we can do things like `cd ..`)
- ~ can be a reference to home directories
  - ~/ is yours
  - ~user1/ is user1’s
- You’ll see how these are useful later

“ls” and “cd” In Action 2

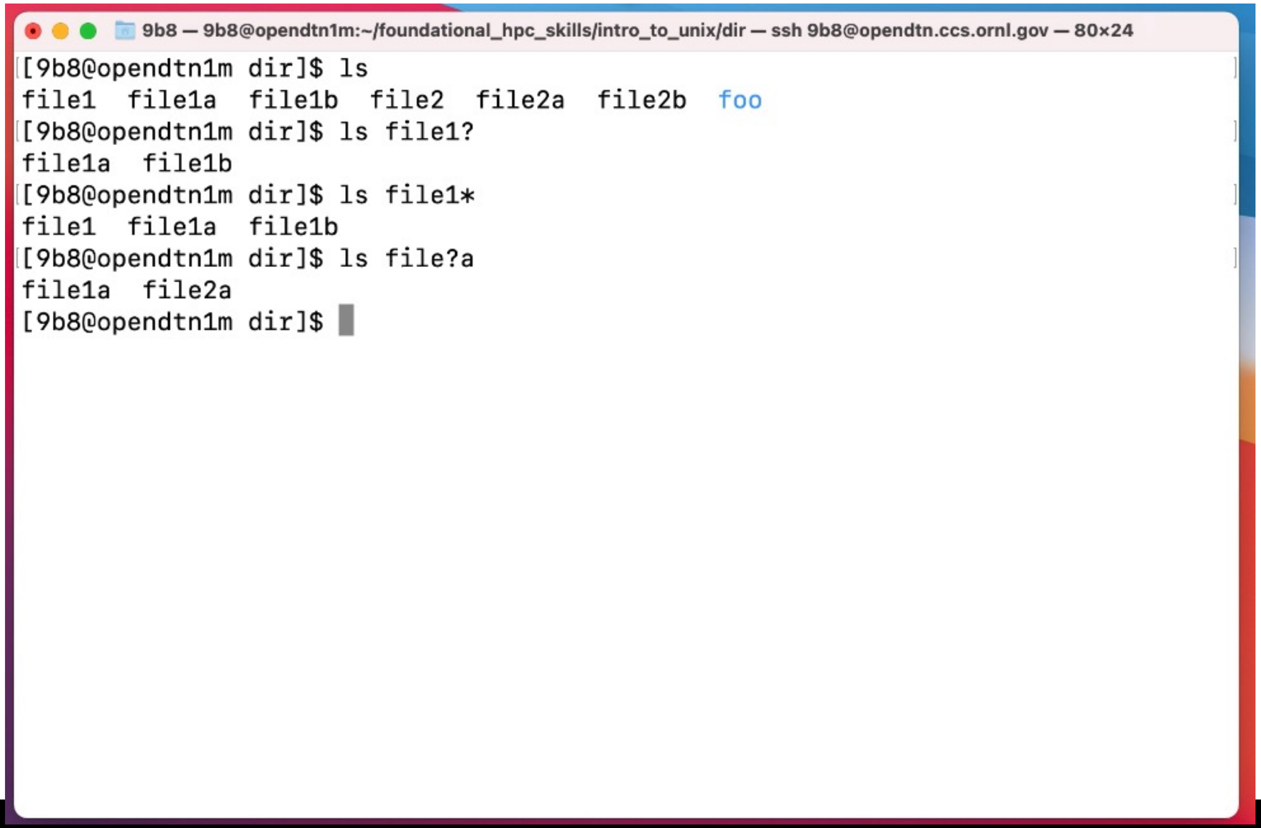

9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m dir]$ ls -a
.  ..  file1  file1a  file1b  file2  file2a  file2b  foo
[9b8@opendtn1m dir]$ pwd
/ccsopen/home/9b8/foundational_hpc_skills/intro_to_unix/dir
[9b8@opendtn1m dir]$ cd .
[9b8@opendtn1m dir]$ pwd
/ccsopen/home/9b8/foundational_hpc_skills/intro_to_unix/dir
[9b8@opendtn1m dir]$ cd ..
[9b8@opendtn1m intro_to_unix]$ pwd
/ccsopen/home/9b8/foundational_hpc_skills/intro_to_unix
[9b8@opendtn1m intro_to_unix]$ cd ~
[9b8@opendtn1m ~]$ pwd
/ccsopen/home/9b8
[9b8@opendtn1m ~]$ 
```

Wildcards

- When dealing with multiple files, it’s nice to type a single command for all files at once vs. typing many separate commands for each file
- Wildcards help with this: They are generic characters that “fill in” for other characters
  - `*` means match zero or more character
  - `?` Matches 1 character
  - Example follows (the `ls` command lists files in a directory, we’ll worry about specific later)

Wildcards (example)



```
9b8 — 9b8@opendtn1m:~/foundational_hpc_skills/intro_to_unix/dir — ssh 9b8@opendtn.ccs.ornl.gov — 80×24
[9b8@opendtn1m dir]$ ls
file1  file1a  file1b  file2  file2a  file2b  foo
[9b8@opendtn1m dir]$ ls file1?
file1a  file1b
[9b8@opendtn1m dir]$ ls file1*
file1  file1a  file1b
[9b8@opendtn1m dir]$ ls file?a
file1a  file2a
[9b8@opendtn1m dir]$
```

* means match zero or more characters

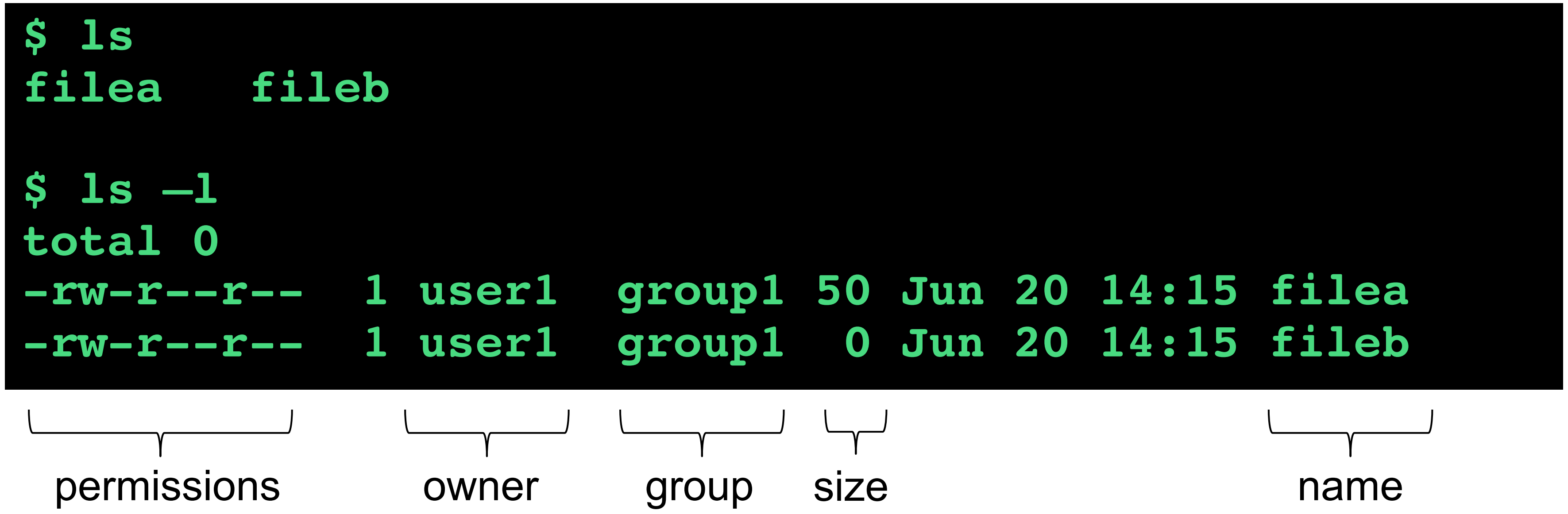
? Matches 1 character

```
$ ls
file1   file1a  file1b  file2   file2a  file2b  file3   file3a  file3b

$ ls file1?
file1a  file1b

$ ls file2*
file2   file2a  file2b

$ ls file?a
file1a  file2a  file3a
```

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

More About “ls”

- Lists directory contents
- Helpful option: -l (shows many file attributes)



```
$ ls
filea    fileb

$ ls –l
total 0
-rw-r--r--  1 user1  group1 50 Jun 20 14:15 filea
-rw-r--r--  1 user1  group1  0 Jun 20 14:15 fileb
```

Even More “ls” Info (Other Useful Options)

Option	Meaning
-1	Show one file per line (helpful in scripting)
-F	Show file types (directories, links, etc)
-a	Show all files (including hidden files)
-r	Reverse the order of the listing
-t	Sort files by timestamp
-d	List the (attributes of) the directory itself rather than listing its contents
…And many, many (many) more	

- You can combine options: `ls -altr` is the same as `ls -a -l -t -r` (but more concise & w/less typing)

Manipulating Files 1

- The `rm` command is used to delete a file.
- The `mv` command is used to move and rename files
- There are multiple ways to create and view text files. In the challenge, we will look at various ways to use `cat` and `less` commands to do this.
- Utilities such as `less` and `cat` are intended only for text files. The system will not stop you from running them on a non-text file
  - If you do, you'll get a screenful of unintelligible characters
  - You might get a recognizable prompt (you might not)
  - There's no shame in closing that session's window & re-connecting

Manipulating Files 2

- The `cp` command is used to copy a file, and the `mv` command is used to move and rename files.

```
$ ls
dir1    filea   fileb

$ cat filea
This is a file that
contains three lines
of text.

$ cp filea filea1

$ ls
dir1    filea   filea1  fileb

$ mv filea1 filec

$ ls
dir1    filea   fileb   filec
```

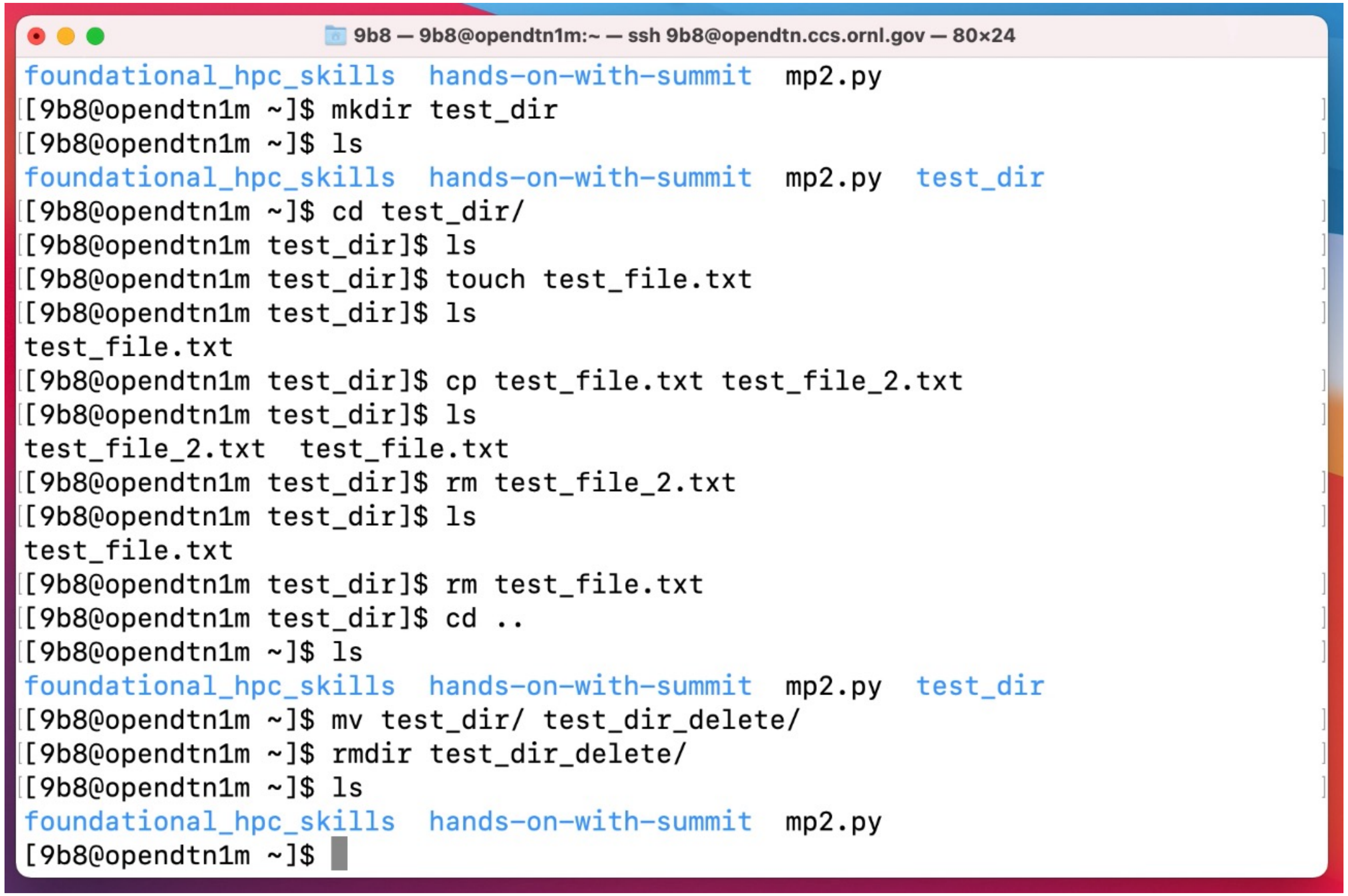
Oak Ridge National Laboratory | Leadership Computing Facility

Manipulating Files 3

- The `rm` command is used to delete a file. `cat` prints contents of a file to the screen. `less` displays the contents of a file in a viewing mode. (Press “q” to exit).

```
$  cat filec
This is a file that
contains three lines
of text.

$ less filec

$ mv filec dir1

$ ls dir1
filec

$ rm dir1/filec

$ ls
dir1    filea   fileb
```

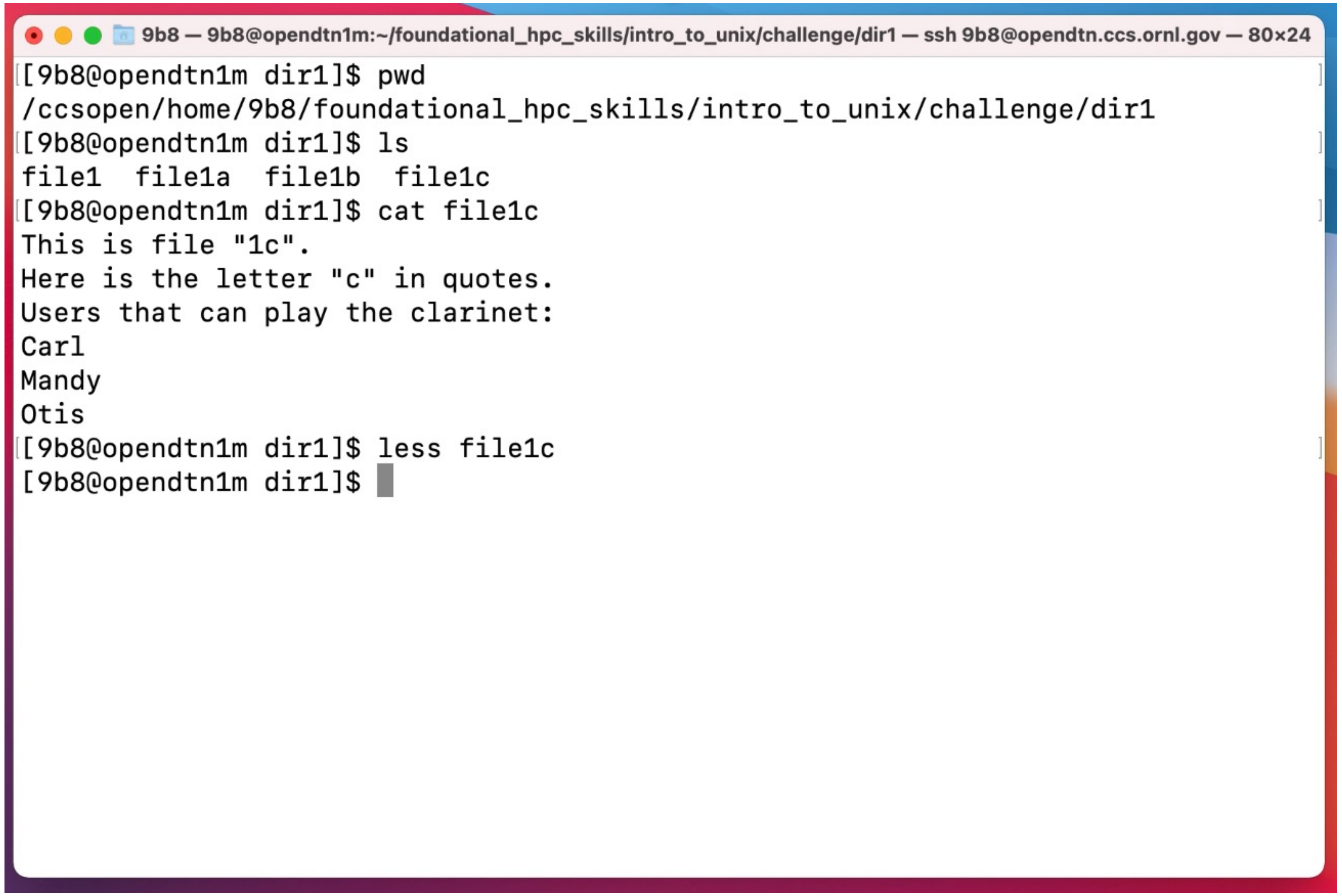
Manipulating Files In Action



9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ mkdir test_dir
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py  test_dir
[9b8@opendtn1m ~]$ cd test_dir/
[9b8@opendtn1m test_dir]$ ls
[9b8@opendtn1m test_dir]$ touch test_file.txt
[9b8@opendtn1m test_dir]$ ls
test_file.txt
[9b8@opendtn1m test_dir]$ cp test_file.txt test_file_2.txt
[9b8@opendtn1m test_dir]$ ls
test_file_2.txt  test_file.txt
[9b8@opendtn1m test_dir]$ rm test_file_2.txt
[9b8@opendtn1m test_dir]$ ls
test_file.txt
[9b8@opendtn1m test_dir]$ rm test_file.txt
[9b8@opendtn1m test_dir]$ cd ..
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py  test_dir
[9b8@opendtn1m ~]$ mv test_dir/ test_dir_delete/
[9b8@opendtn1m ~]$ rmdir test_dir_delete/
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$
```

“cat” and “less” In Action


9b8 — 9b8@opendtn1m:~/foundational_hpc_skills/intro_to_unix/challenge/dir1 — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m dir1]$ pwd
/ccsopen/home/9b8/foundational_hpc_skills/intro_to_unix/challenge/dir1
[9b8@opendtn1m dir1]$ ls
file1  file1a  file1b  file1c
[9b8@opendtn1m dir1]$ cat file1c
This is file "1c".
Here is the letter "c" in quotes.
Users that can play the clarinet:
Carl
Mandy
Otis
[9b8@opendtn1m dir1]$ less file1c
[9b8@opendtn1m dir1]$ 
```

Many uses of cat

Command	Explanation
cat file1.txt	Display contents of file
cat file1.txt file2.txt	Concatenate two text files and display the result in the terminal
cat file1.txt file2.txt > newcombinedfile.txt	Concatenate two text files and write them to a new file
cat >newfile.txt	Create a file called newfile.txt. Type the desired input and press CTRL+D to finish. The text will be in file newfile.txt.
cat -n file1.txt file2.txt > newnumberedfile.txt	Some implementations of cat, with option -n, can also number lines
cat file1.txt > file2.txt	Copy the contents of file1.txt into file2.txt
cat file1.txt >> file2.txt	Append the contents of file1.txt to file2.txt
cat file1.txt file2.txt | less	Run the program "less" with the concatenation of file1 and file2 as its input

Searching Within Files – grep 1

- Sometimes you want to search for patterns/strings in a file. As with other commands, `grep` takes many options. The grep command searches for "regular expressions"…strings that contain characters with special meaning

- Simple case: find lines with the string 'user' in file1

```
grep “user” file1
```

- More complex: show lines ending with 'user' in file1

```
grep “user$” file1
```

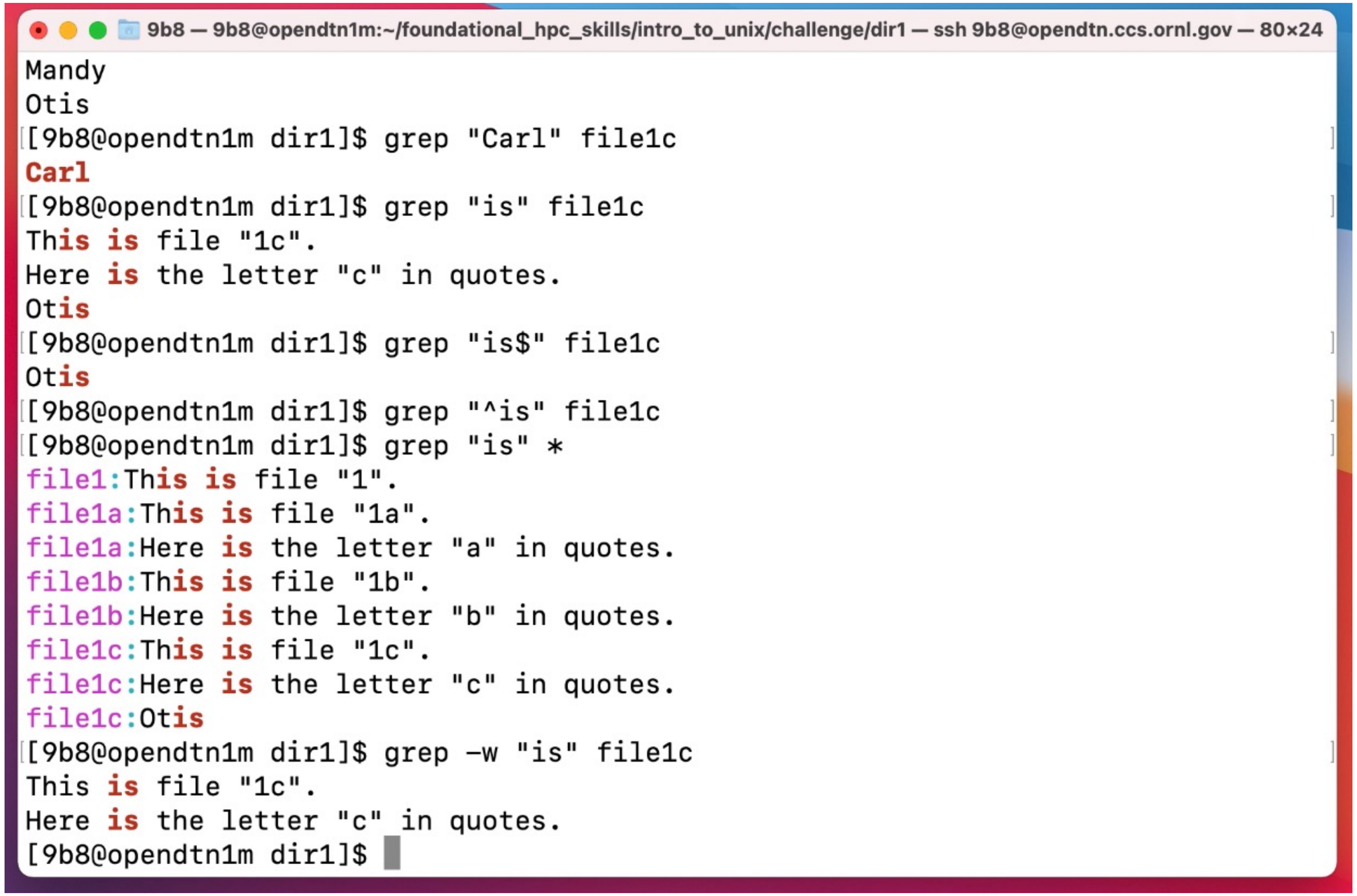
- …or perhaps lines beginning with 'user'

```
grep “^user” file1
```

- Search all files in a dir1 for the string 'user'

```
grep –r “user” dir1/
```

"grep" In Action



9b8 — 9b8@opendtn1m:~/foundational_hpc_skills/intro_to_unix/challenge/dir1 — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
Mandy
Otis
[9b8@opendtn1m dir1]$ grep "Carl" file1c
Carl
[9b8@opendtn1m dir1]$ grep "is" file1c
This is file "1c".
Here is the letter "c" in quotes.
Otis
[9b8@opendtn1m dir1]$ grep "is$" file1c
Otis
[9b8@opendtn1m dir1]$ grep "^is" file1c
[9b8@opendtn1m dir1]$ grep "is" *
file1:This is file "1".
file1a:This is file "1a".
file1a:Here is the letter "a" in quotes.
file1b:This is file "1b".
file1b:Here is the letter "b" in quotes.
file1c:This is file "1c".
file1c:Here is the letter "c" in quotes.
file1c:Otis
[9b8@opendtn1m dir1]$ grep -w "is" file1c
This is file "1c".
Here is the letter "c" in quotes.
[9b8@opendtn1m dir1]$
```

Searching Within Files – grep 2

- Normally, `grep` will treat anything beginning with a hyphen as an option…even if it's in quotes
- The workaround is the `--` option, which tells grep that you're done giving it options (and therefore any other hyphen is meant as an actual hyphen)

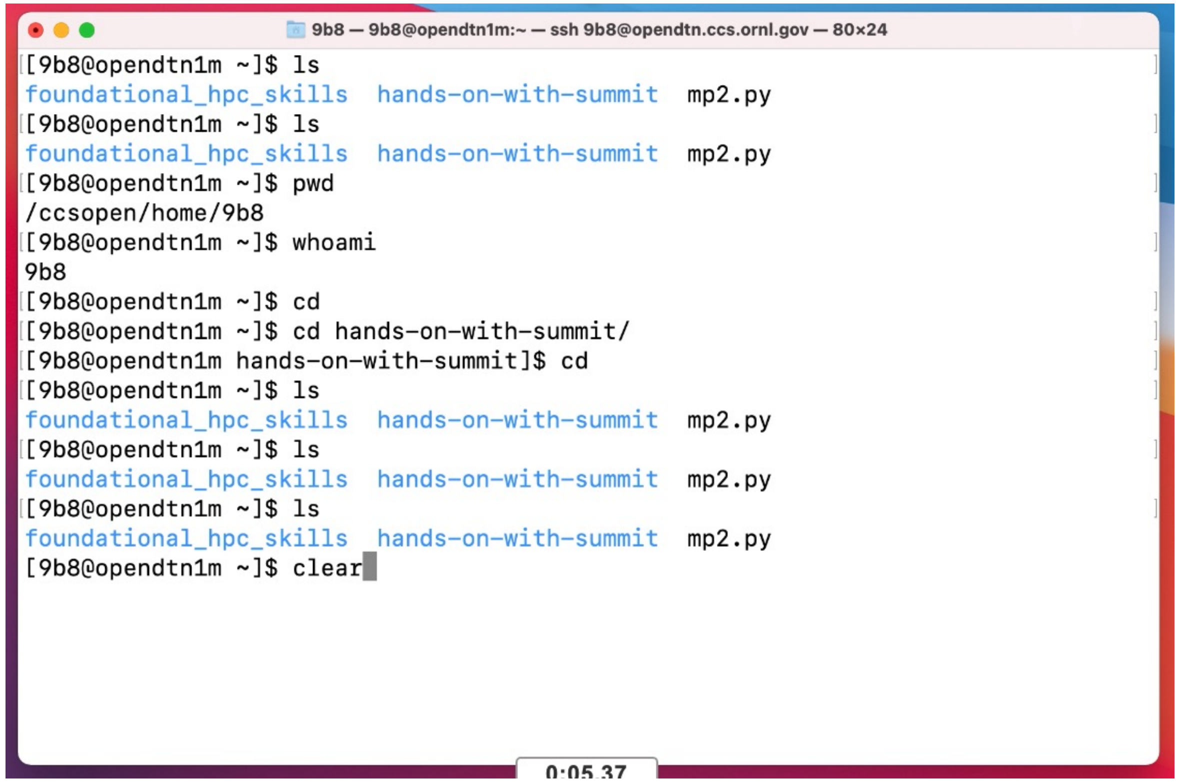
  ```
  grep -- “-2” file1
  ```

- Grep does NOT recognize wildcards in the quoted string to search for. Use “.” instead:

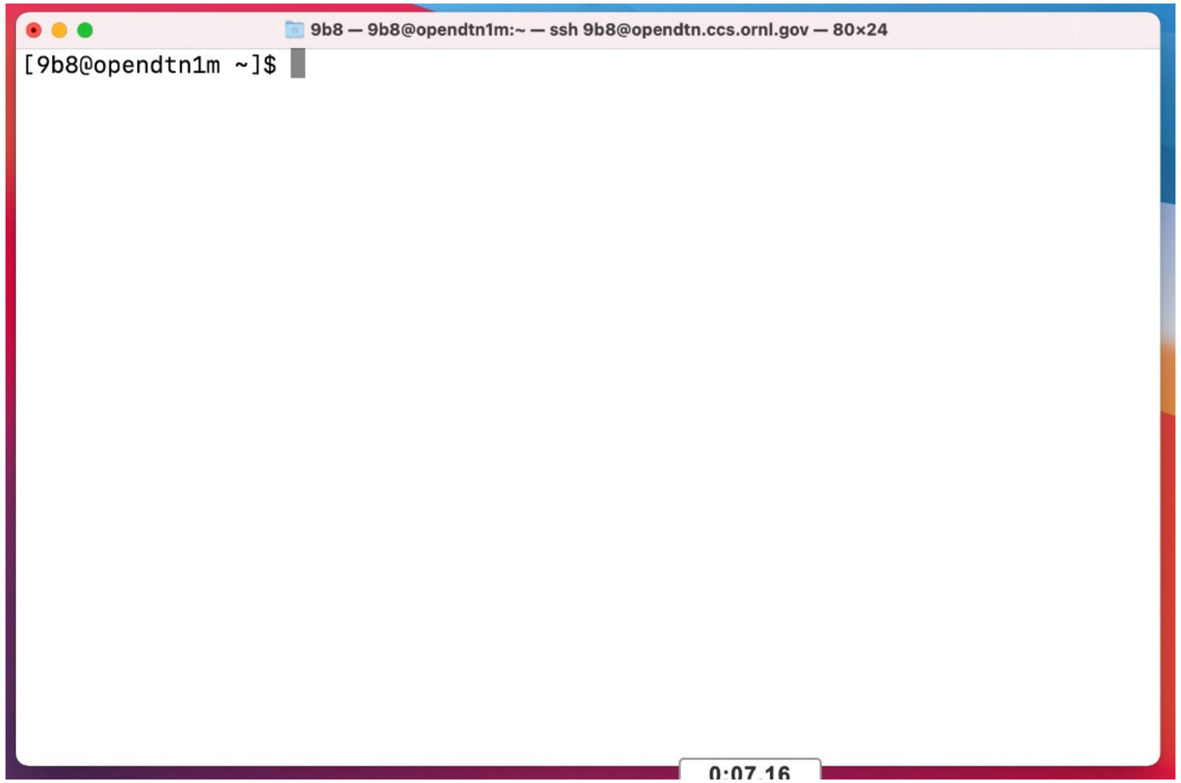
```
grep “user.” file1
```

Bonus Tips 1

- “clear” is your friend:


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24
0:05.37

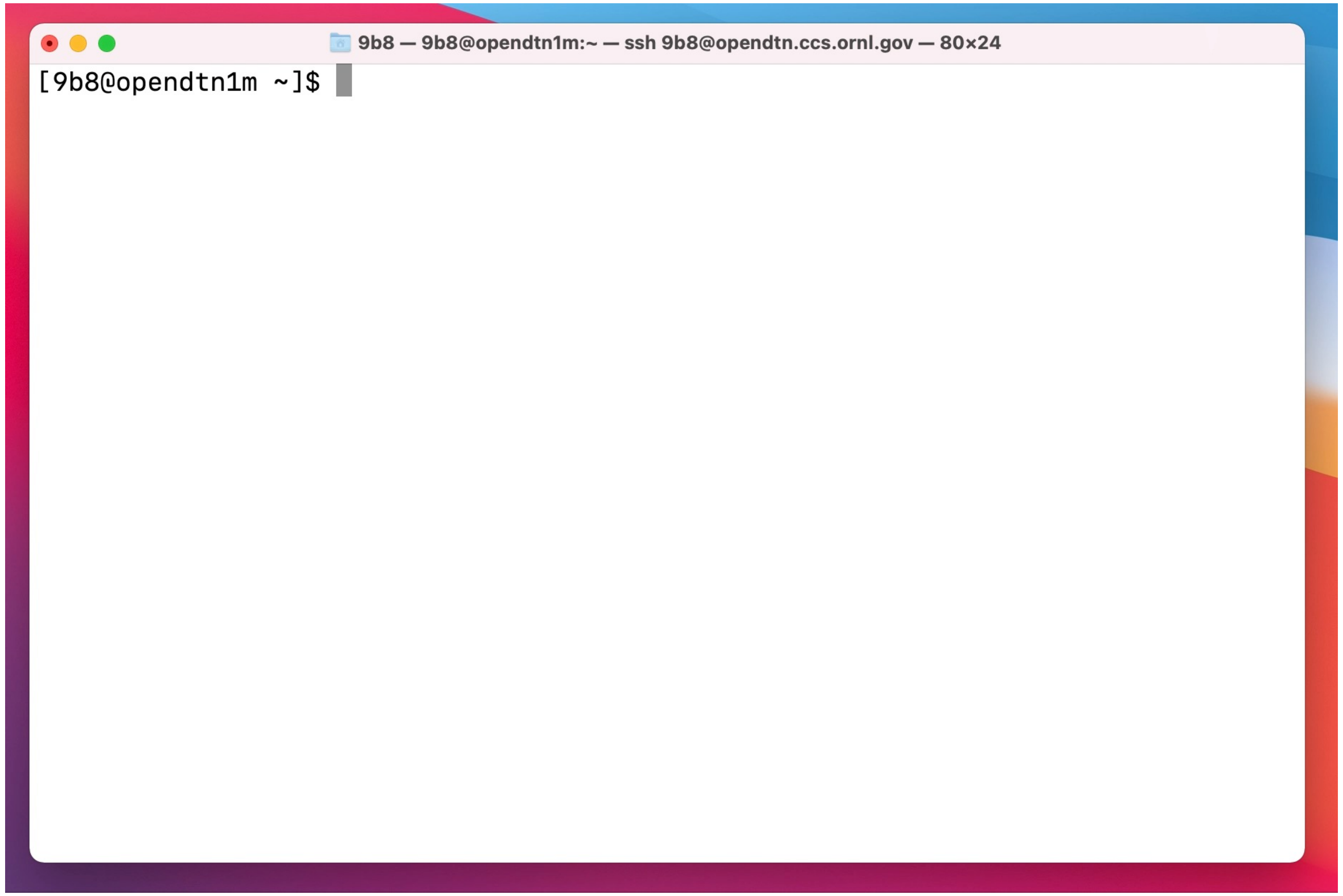
```
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ pwd
/ccsopen/home/9b8
[9b8@opendtn1m ~]$ whoami
9b8
[9b8@opendtn1m ~]$ cd
[9b8@opendtn1m ~]$ cd hands-on-with-summit/
[9b8@opendtn1m hands-on-with-summit]$ cd
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ ls
foundational_hpc_skills  hands-on-with-summit  mp2.py
[9b8@opendtn1m ~]$ clear
```


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24
0:07.16

```
[9b8@opendtn1m ~]$
```

Bonus Tips 2

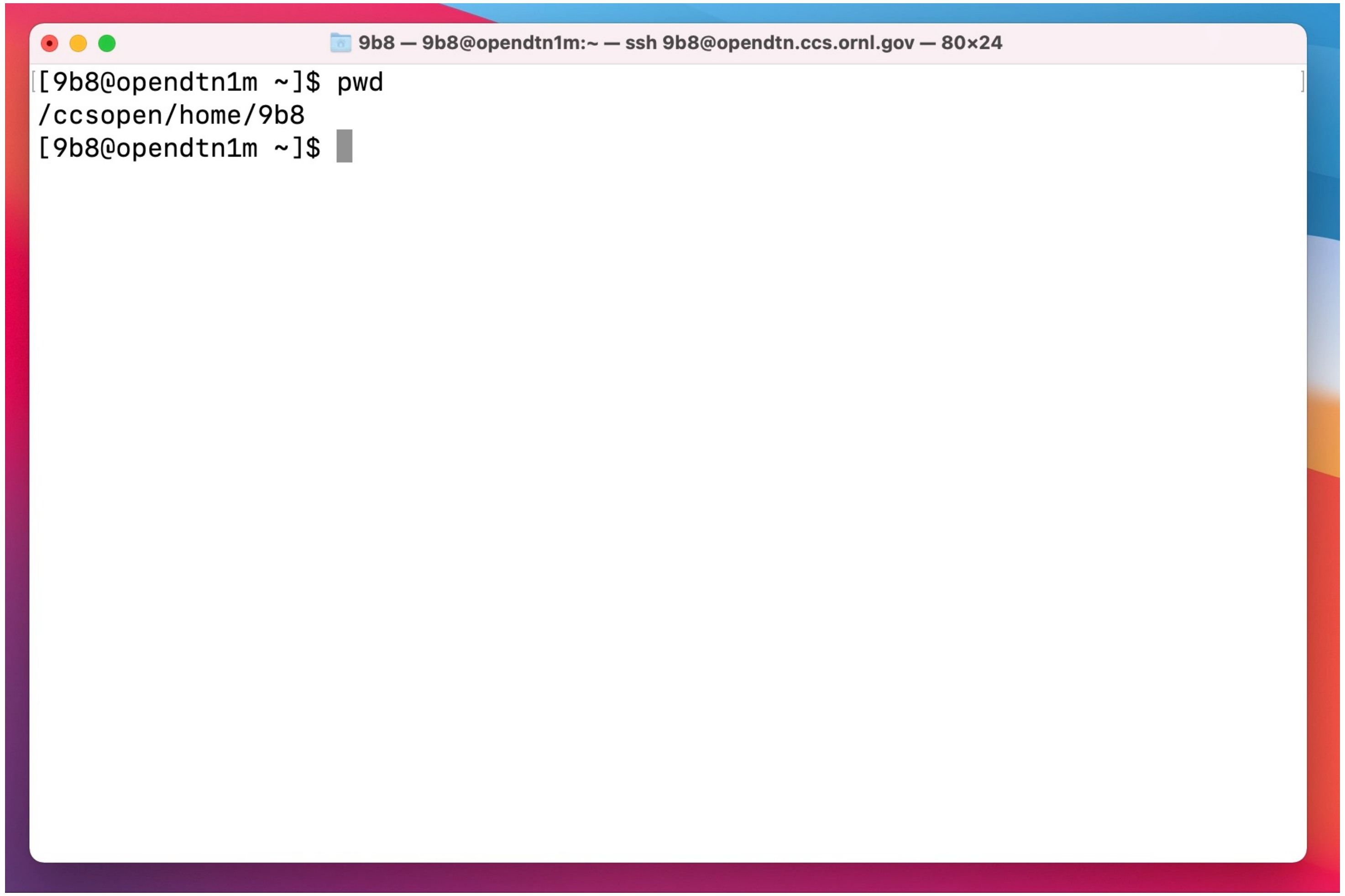
- Up/down arrows are also your friends:


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m ~]$ 
```

Bonus Tips 3

- Tab-complete is your best friend:


9b8 — 9b8@opendtn1m:~ — ssh 9b8@opendtn.ccs.ornl.gov — 80×24

```
[9b8@opendtn1m ~]$ pwd
/ccsopen/home/9b8
[9b8@opendtn1m ~]$ 
```

Basic File/Directory Commands

Command	Description
pwd	**P**rint **w**orking (i.e. current) **d**irectory
man	Display the **Man**ual for a command
whoami	Display the current user's username
mkdir	Create a directory (**M**a**K**e **DIR**ectory)
rmdir	Delete a directory (**R**e**M**ove **DIR**ectory)
cd	**C**hange (into a) **d**irectory
ls	**Lis**t files
cp	**C**o**p**y a file
mv	**M**o**v**e a file (also used to rename a file)
rm	**Rem**ove (delete) a file
cat	Display the contents (con**cat**enate) a (hopefully text) file
less	Display the contents of a text file in a viewing mode
find	**Find** files by filename
grep	Search for text within files
diff	Identify **diff**erences in the contents of files

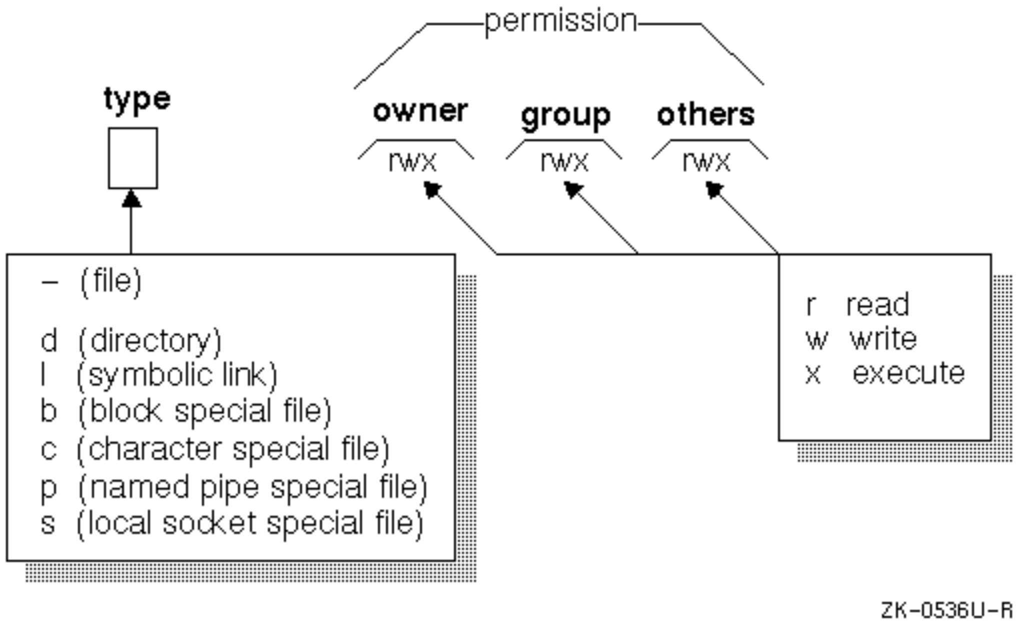
Extra Slides

Files

- Files are the basic entity for storing data
- Might contain a program, data, configuration info, an image, etc.
- Files have several attributes
  - Permissions: who can do what to/with a file
  - Owner: whose file is this
  - Group: to which group does this file belong

Permissions

- In the permissions string, the characters r, w, and x mean read, write, and execute permission is granted
- A - means the permission is not granted
- The permission groups always show read, write, execute in that order



Searching for files

- The `find` command lets you search for files on a huge variety of criteria
- It can also run commands on those files; this makes it one of the most powerful commands available
- General Syntax: `find [path/paths] [expression]`

```
$ find . -name “*data*” -print

$ find . -newer some_file

$ find /home/user1 ! -user user1

$ find . -group users -exec chgrp staff {} \;
```