

Bioinformatics Lessons Schedule

NOTE: It's already changed 3 times, so it will continue to change as we go forward.

Date	Subject
10-22	Terminus
10-29	Server Basics
11-5	Server Basics, continued
11-12	Server Basics, continued
11-19	How to run software on the server
11-26	No lesson, week of Thanksgiving
12-03	Basic Git
12-10	Start running through RRBS (or RNA-seq)
12-17	Continue from the previous week
12-24	Christmas break
12-31	Christmas break
01-07	Resume lessons

How to Use the Server

Command-Line Basic Navigation (continued)

Viewing Files

- Viewing files

```
$ ls
```

```
file1.txt file2.txt
```

Viewing Files

- Viewing files
 - head = prints the first 10 lines

```
$ ls
file1.txt file2.txt
$ head file1.txt
A is for aardvark
B is for bumblebee
C is for chihuahua
D is for donkey
E is for elephant shrew
F is for flamingo
G is for Galapagos tortoise
H is for hippopotamus
I is for iguana
J is for jackal
```

Viewing Files

- Viewing files
 - head = prints the first 10 lines
 - tail = prints the last 10 lines

```
$ tail file1.txt
Q is for quetzal
R is for raccoon
S is for squirrel
T is for tarantula
U is for unicorn
V is for vicuna
W is for warthog
X is for x-ray tetra
Y is for yak
Z is for zebra
```

Viewing Files

- Viewing files
 - head = prints the first 10 lines
 - tail = prints the last 10 lines
 - more = fills screen with file; hit enter to advance one line
 - less = page through text in a different screen; can go backwards and forwards; q to exit

Demo more, less on
terminal

Viewing Files

- Viewing files
 - head = prints the first 10 lines
 - tail = prints the last 10 lines
 - more = fills screen with file; hit enter to advance one line
 - less = page through text in a different screen; can go backwards and forwards; q to exit
 - cat = concatenate; used with no other flags it will print the entire file; use zcat for gzipped files

```
$ cat file1.txt
A is for aardvark
B is for bumblebee
C is for chihuahua
D is for donkey
E is for elephant shrew
F is for flamingo
G is for Galapagos tortoise
H is for hippopotamus
I is for iguana
J is for jackal
K is for kangaroo
L is for llama
M is for mongoose
N is for narwhal
O is for octopus
P is for penguin
Q is for quetzal
R is for raccoon
S is for squirrel
T is for tarantula
U is for unicorn
V is for vicuna
W is for warthog
X is for x-ray tetra
Y is for yak
Z is for zebra
```


Viewing Files

- Viewing files
 - head = prints the first 10 lines
 - tail = prints the last 10 lines
 - more = fills screen with file; hit enter to advance one line
 - less = page through text in a different screen; can go backwards and forwards; q to exit
 - cat = concatenate; used with no other flags it will print the entire file ; use zcat for gzipped files
- Making new files
 - touch

```
$ touch new_file.txt
```

```
$ head new_file.txt
```

```
$
```

Viewing Files

- Viewing files
 - head = prints the first 10 lines
 - tail = prints the last 10 lines
 - more = fills screen with file; hit enter to advance one line
 - less = page through text in a different screen; can go backwards and forwards; q to exit
 - cat = concatenate; used with no other flags it will print the entire file ; use zcat for gzipped files
- Making new files
 - touch
 - With editors
 - nano
 - vim

Demo nano, vim on terminal

PRACTICE

BREAK

Navigating the Filesystem

Navigating the Filesystem (Vocabulary)

- Root directory = /
- Absolute path = the path starting from the root directory
- Relative path = path relative to your current directory

Navigating the Filesystem (Commands)

- `pwd` = "print working directory";
prints the absolute path of the
current folder

```
$ pwd  
/home/user/examples
```

Navigating the Filesystem

- pwd = "print working directory"; prints the absolute path of the current folder
- cd = "change directory"

```
$ pwd  
/home/user/examples  
$ cd /home
```

Navigating the Filesystem

- `pwd` = "print working directory"; prints the absolute path of the current folder
- `cd` = "change directory"

```
$ pwd  
/home/user/examples
```

```
$ cd /home
```

```
$ ls
```

```
user user1 user2 user3
```


Navigating the Filesystem

- `pwd` = "print working directory"; prints the absolute path of the current folder
- `cd` = "change directory"
 - Can do multiple directories at once
 - `~` = your home directory

```
$ pwd
/home/user/examples
$ cd /home
$ ls
user user1 user2 user3
$ cd ~
$ pwd
/home/user/
```

Navigating the Filesystem

- `pwd` = "print working directory"; prints the absolute path of the current folder
- `cd` = "change directory"
 - Can do multiple directories at once
 - `~` = your home directory
 - `.` = current directory
 - `..` = directory above my directory

```
$ pwd
/home/user/examples
$ cd /home
$ ls
user user1 user2 user3
$ cd ~
$ pwd
/home/user/
$ cd ..
$ pwd
/home/
```

Navigating the Filesystem

- pwd = "print working directory"; prints the absolute path of the current folder
- cd = "change directory"
 - Can do multiple directories at once
 - ~ = your home directory
 - . = current directory
 - .. = directory above my directory
 - - = last directory

```
$ pwd
/home/user/examples
$ cd /home
$ ls
user user1 user2 user3
$ cd ~
$ pwd
/home/user/
$ cd ..
$ pwd
/home/
$ cd -
$ pwd
/home/user/
```

PRACTICE

BREAK

File Wrangling

- cp = “copy”

```
$ ls
```

```
file1.txt file2.txt
```

File Wrangling

- cp = “copy”

```
$ ls
```

```
file1.txt file2.txt
```

```
$ head file2.txt
```

```
This is an example  
file!
```

File Wrangling

- cp = “copy”

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example
file!
$ cp file2.txt ../
```

File Wrangling

- cp = “copy”

```
$ ls
```

```
file1.txt file2.txt
```

```
$ head file2.txt
```

```
This is an example file!
```

```
$ cp file2.txt ../
```

```
$ cd ../
```

```
$ ls
```

```
folder file.txt file2.txt
```


File Wrangling

- cp = “copy”

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ cp file2.txt ../
$ cd ../
$ ls
folder file.txt file2.txt
$ ls folder
file1.txt file2.txt
```

File Wrangling

- cp = “copy”

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ cp file2.txt ../
$ cd ../
$ ls
folder file.txt file2.txt
$ ls folder
file1.txt file2.txt
$ head file2.txt
This is an example file!
```

File Wrangling

- cp = “copy”

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example
file!
$ cp file2.txt ../
$ cd ../
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
```

```
file1.txt file2.txt
```

```
$ head file2.txt
```

```
This is an example file!
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location AND rename the file

```
$ ls
```

```
file1.txt file2.txt
```

```
$ head file2.txt
```

```
This is an example file!
```

```
$ mv file2.txt ../
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location AND rename the file

```
$ ls
```

```
file1.txt file2.txt
```

```
$ head file2.txt
```

```
This is an example file!
```

```
$ mv file2.txt ../
```

```
$ cd ../
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
$ ls folder
file1.txt
```


File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
$ ls folder
file1.txt
$ head file2.txt
This is an example file!
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
$ ls folder
file1.txt
$ head file2.txt
This is an example file!
$ mv file2.txt example.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
$ ls folder
file1.txt
$ head file2.txt
This is an example file!
$ mv file2.txt example.txt
$ ls
folder file.txt example.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file

```
$ ls
file1.txt file2.txt
$ head file2.txt
This is an example file!
$ mv file2.txt ../
$ cd ../
$ ls
folder file1.txt file2.txt
$ ls folder
file1.txt
$ head file2.txt
This is an example file!
$ mv file2.txt example.txt
$ ls
folder file2.txt example.txt
$ head example.txt
This is an example file!
```

File Wrangling

```
$ ls
```

```
folder file2.txt example.txt
```

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
```

```
folder file.txt example.txt
```

```
$ rm example.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
folder file.txt example.txt
$ rm example.txt
$ ls
folder
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
```

```
folder file.txt example.txt
```

```
$ rm example.txt
```

```
$ ls
```

```
folder
```

```
$ rm folder
```


File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
```

```
folder file.txt example.txt
```

```
$ rm example.txt
```

```
$ ls
```

```
folder
```

```
$ rm folder
```

```
rm: cannot remove 'folder/': Is a  
directory
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
```

```
folder file.txt example.txt
```

```
$ rm example.txt
```

```
$ ls
```

```
folder
```

```
$ rm folder
```

```
rm: cannot remove 'folder/': Is a  
directory
```

```
$ rm -r folder
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
folder file2.txt example.txt
$ rm example.txt
$ ls
folder
$ rm folder
rm: cannot remove 'folder/': Is a
directory
$ rm -r folder
$ ls
file2.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files

```
$ ls
folder file.txt example.txt
$ rm example.txt
$ ls
folder
$ rm folder
rm: cannot remove 'folder/': Is a
directory
$ rm -r folder
$ ls
file.txt
```

WARNING: There is NO RECYCLING BIN; when you delete something it's gone forever

```
$ ls
```

```
file.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

File Wrangling

```
$ ls
```

```
file.txt
```

```
$ mkdir new_folder
```

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
```


File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file2.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file2.txt
$ mkdir second_folder
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file2.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file2.txt
$ mkdir second_folder
$ touch second_folder/file.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file.txt
$ mkdir second_folder
$ touch second_folder/file.txt
$ rmdir second_folder
Rmdir: failed to remove
'second_folder/'; Directory not
empty
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file.txt
$ mkdir second_folder
$ touch second_folder/file.txt
$ rmdir second_folder
Rmdir: failed to remove
'second_folder/'; Directory not
empty
$ rm second_folder/file.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file.txt
$ mkdir second_folder
$ touch second_folder/file.txt
$ rmdir second_folder
Rmdir: failed to remove
'second_folder/'; Directory not
empty
$ rm second_folder/file.txt
$ rmdir second_folder
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty

```
$ ls
file.txt
$ mkdir new_folder
$ ls
file.txt new_folder
$ rmdir new_folder
$ ls
file.txt
$ mkdir second_folder
$ touch second_folder/file.txt
$ rmdir second_folder
Rmdir: failed to remove
'second_folder/'; Directory not
empty
$ rm second_folder/file.txt
$ rmdir second_folder
$ ls
file.txt
```

File Wrangling

- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research  
44 Jul 9 14:45 file1.txt.gz
```


File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
-rwxr--r--. 1 kkeith research 44
Jul 9 14:45 file1.txt.gz
```

The diagram illustrates the components of the `ll` command output. Red brackets and boxes are used to identify the following parts:

- file permissions**: `-rwxr--r--.`
- file owner**: `kkeith`
- file group**: `research`

Other visible components in the output include the file size (`44`), the date and time (`Jul 9 14:45`), and the filename (`file1.txt.gz`).

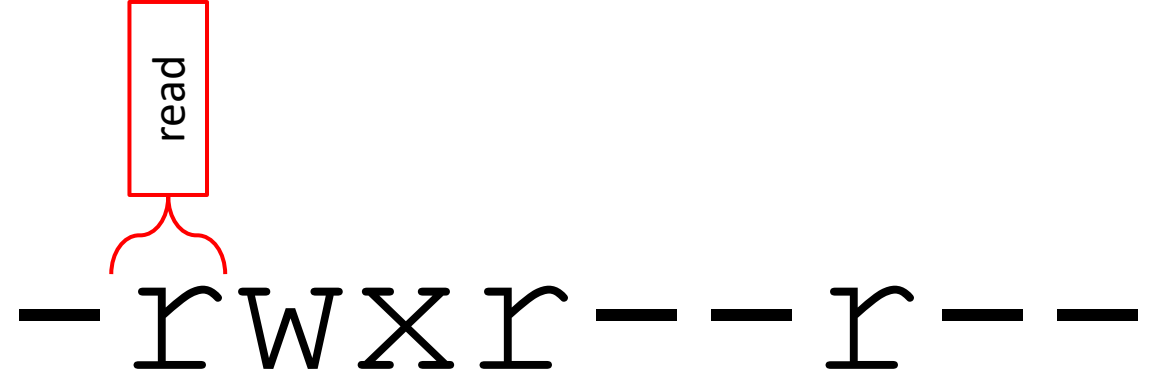
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

- rwxr--r--

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



The diagram shows the file permissions `-rwxr--r--`. A red bracket is drawn under the first `r` (owner permissions), and a red box labeled "read" is positioned above it, indicating that the owner has read permission.

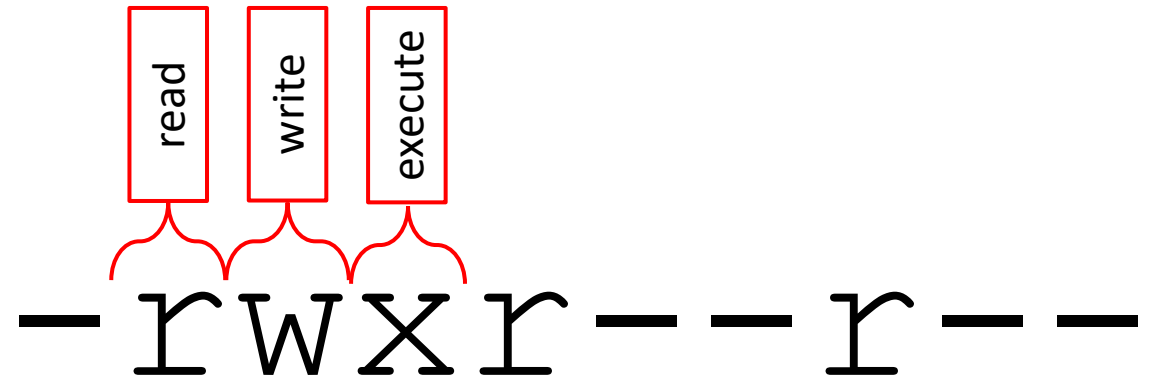
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

The diagram shows the file permissions `-rwxr--r--`. Above the first three characters `rwx`, there are two red boxes: one labeled `read` pointing to the `r` and one labeled `write` pointing to the `w`. Above the next three characters `r--`, there is a red box labeled `read` pointing to the `r`. The final three characters `r--` are not annotated.

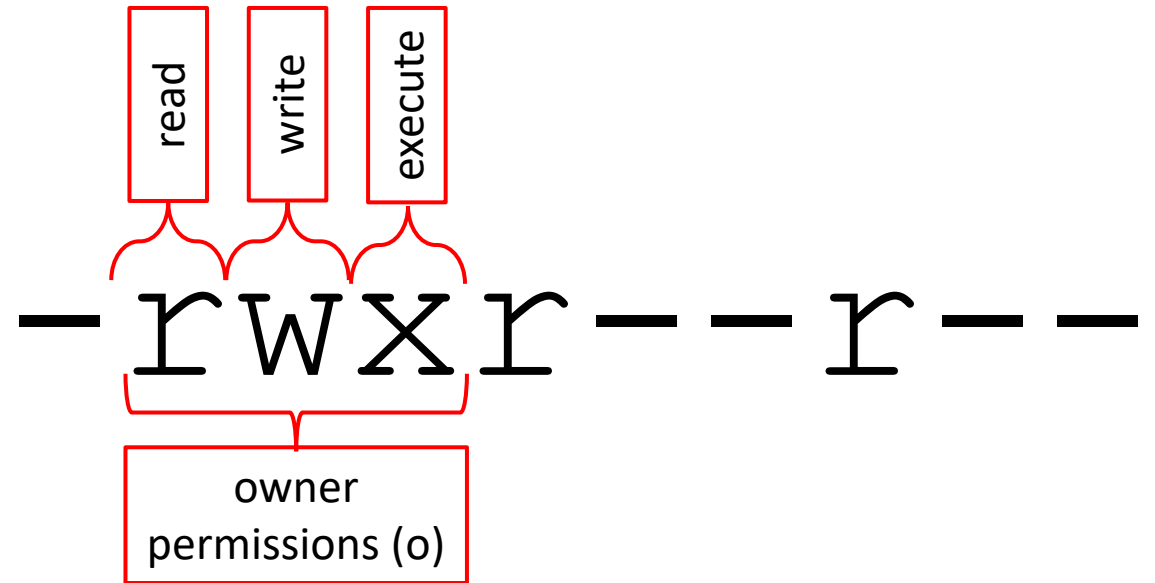
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



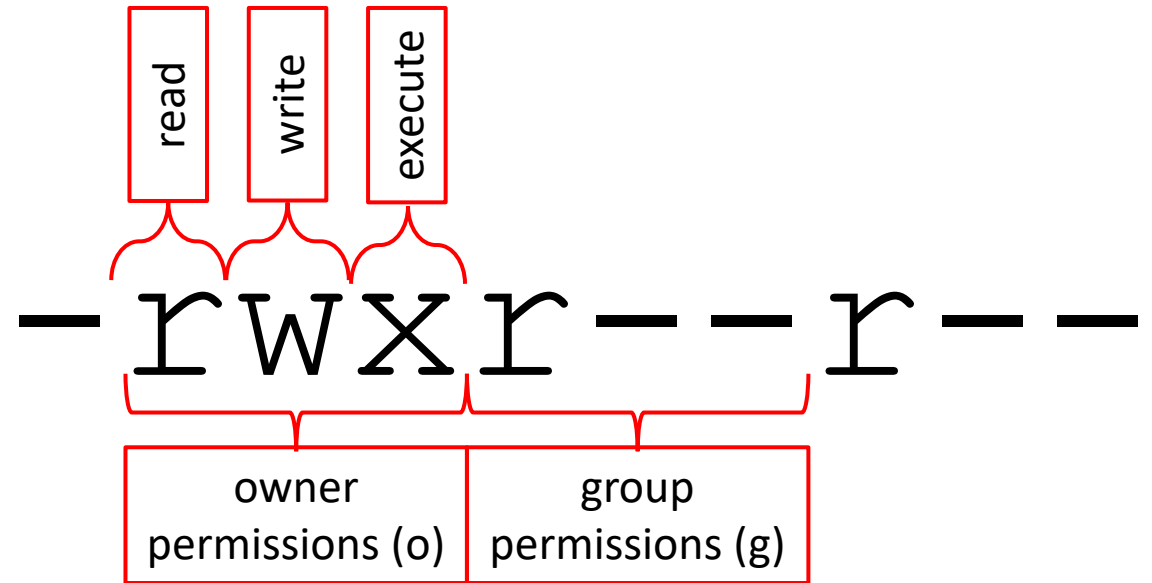
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



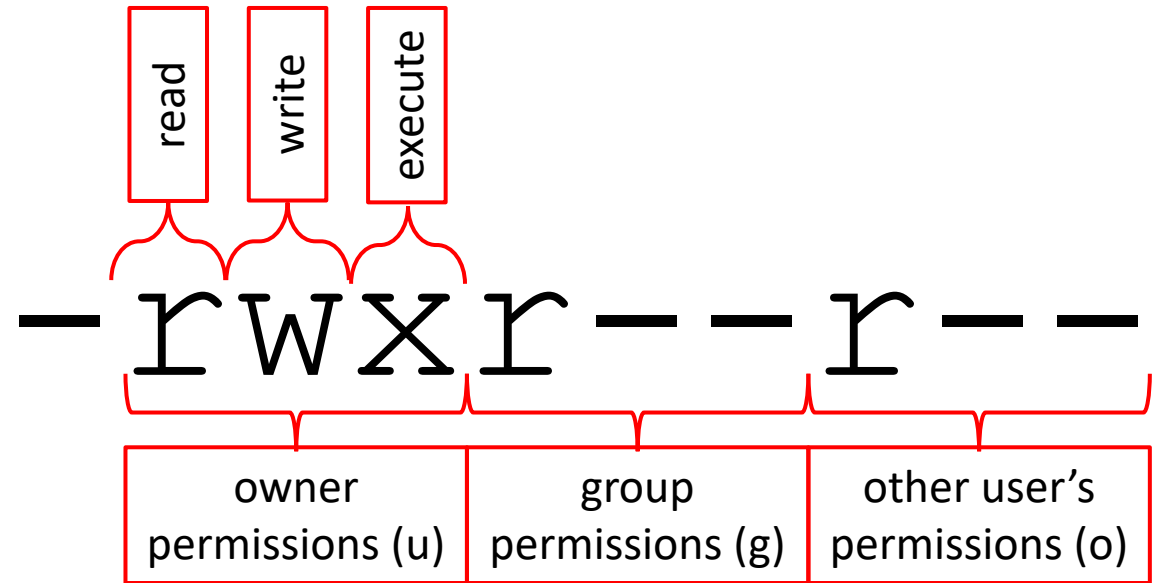
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



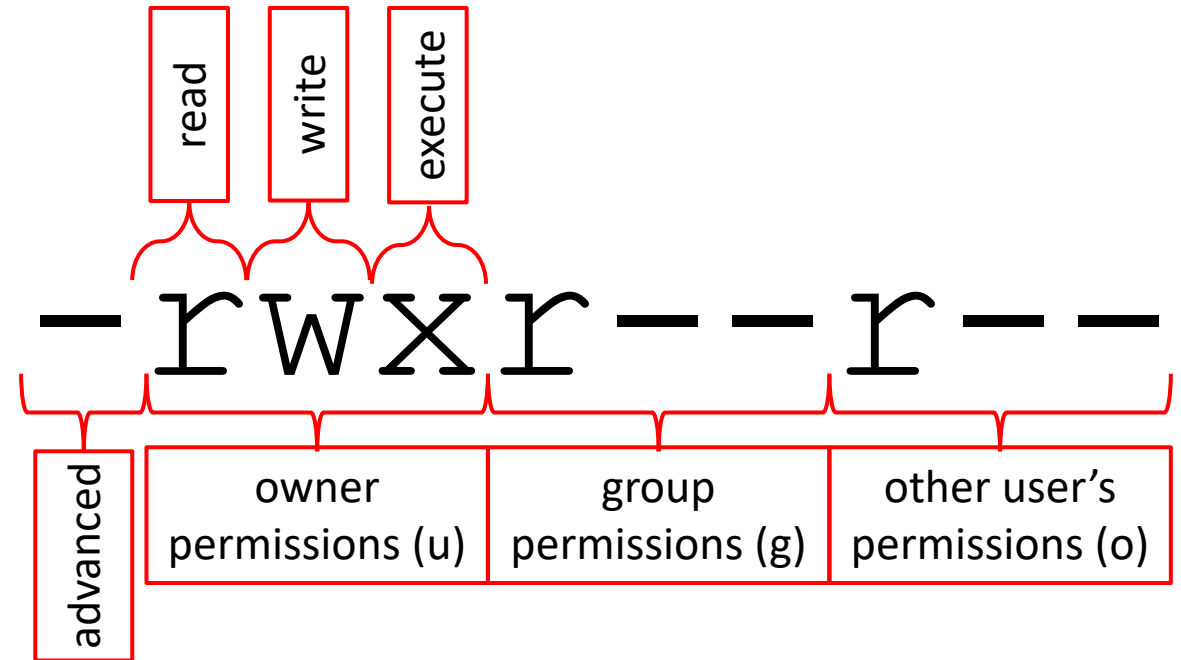
File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group



File Wrangling

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul  9 14:45 file1.txt.gz
```

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul  9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul  9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

file owner
remove
execute permissions

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul  9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul  9 14:45 file1.txt.gz
```

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o+wx file1.txt.gz
```

other users
add
write and execute
permissions

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o+wx file1.txt.gz
```

```
$ ll
```

```
-rw-r--rwx. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o+wx file1.txt.gz
```

```
$ ll
```

```
-rw-r--rwx. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o-x file1.txt.gz
```

other users
remove
execute permissions

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o+wx file1.txt.gz
```

```
$ ll
```

```
-rw-r--rwx. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o-x
```

```
$ chmod a+rw file1.txt.gz
```

All users
add
read and write
permissions

File Wrangling

- making new files
 - touch
 - nano or vim editors
- cp = “copy”
- mv = “move”; Can move the file location OR rename the file
- rm = “remove”; rm -r to remove a directory and all its files
- mkdir = “make directory”; make a new directory/folder
- rmdir = “remove directory”; only works if the directory/folder is empty
- Changing file/folder permissions and ownership
 - chmod = change the file permissions
 - chown = change the file owner
 - chgrp = change the file group

```
$ ll
```

```
-rwxr--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod u-x file1.txt.gz
```

```
$ ll
```

```
-rw-r--r--. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o+wx file1.txt.gz
```

```
$ ll
```

```
-rw-r--rwx. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

```
$ chmod o-x
```

```
$ chmod a+rw
```

```
$ ll
```

```
-rw-rw-rw-. 1 kkeith research 44  
Jul 9 14:45 file1.txt.gz
```

PRACTICE

BREAK