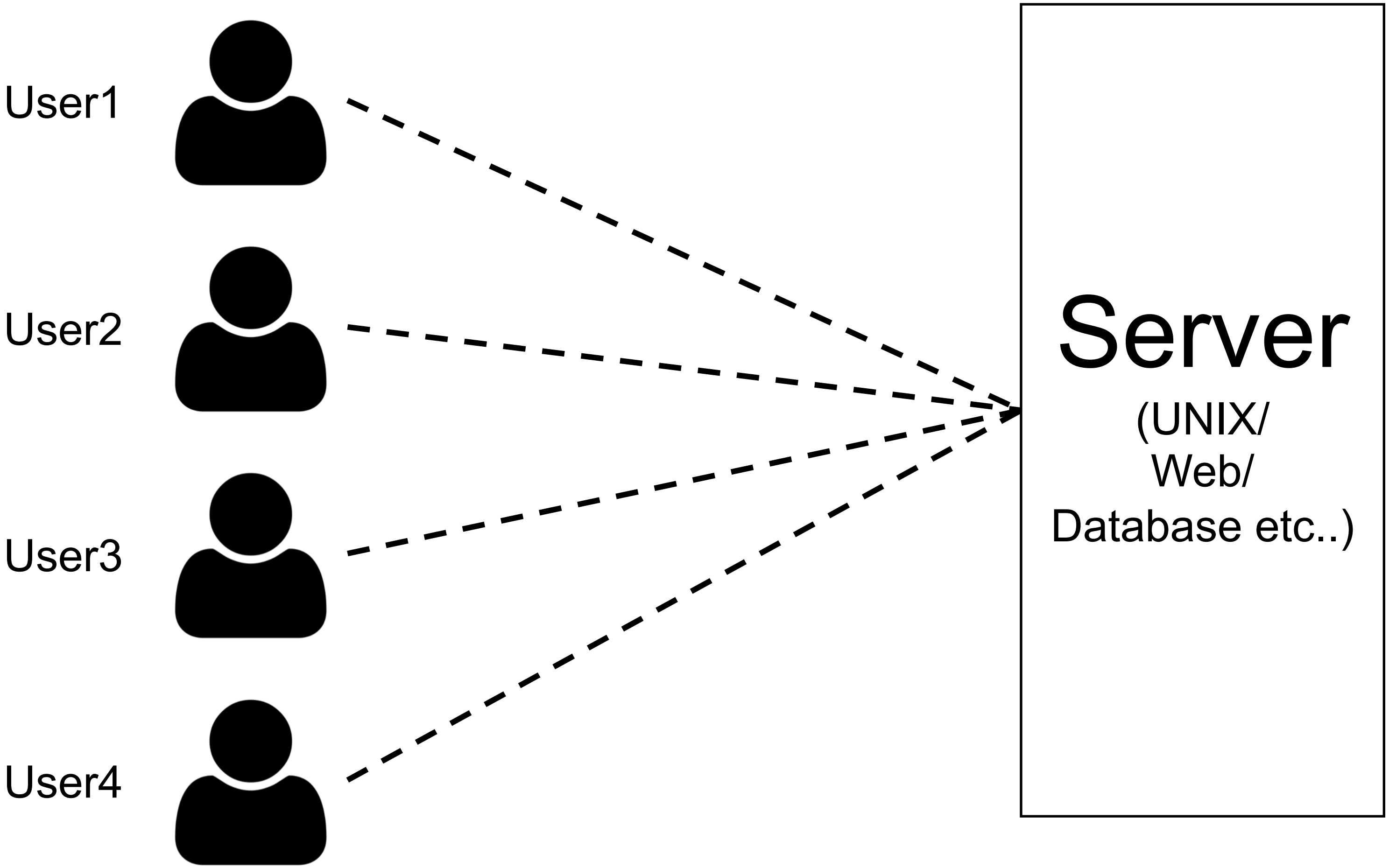
Basic UNIX commands

HORT 59000

Lab 2

Instructor: Kranthi Varala

Client/Server architecture



User1

User2

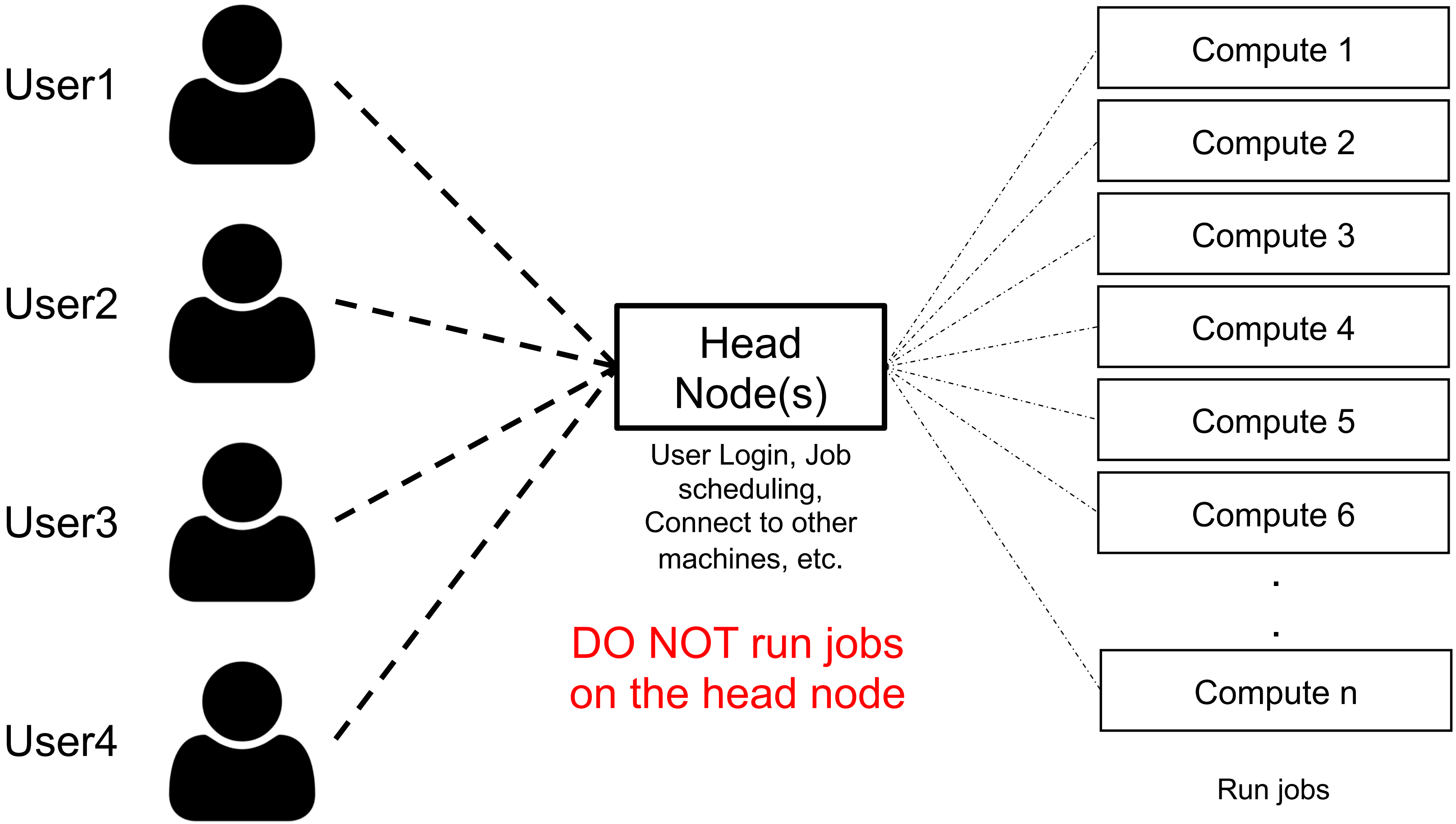
User3

User4

Server

(UNIX/
Web/
Database etc..)

High Performance Compute (HPC) cluster



User1

User2

User3

User4

Head Node(s)

User Login, Job scheduling, Connect to other machines, etc.

DO NOT run jobs on the head node

Compute 1

Compute 2

Compute 3

Compute 4

Compute 5

Compute 6

.

.

Compute n

Run jobs

Scholar : Our class server

Sub-Cluster	Number of Nodes	Processors per Node	Cores per Node	Memory per Node	Interconnect
Head	7	Two 10-Core Intel CPUs	20	512-754 GB	56 Gbps FDR Infiniband
A	8	Two 10-Core Intel CPUs	20	64 GB	

- Scholar is a Linux cluster maintained by Purdue and is available for all computational classes.
- Secure Shell (SSH) protocol is the most common way to connect to remote UNIX/LINUX servers.
- The specific SSH client you use depends on your client (laptop) OS.
- Linux and MacOS have built-in clients.
- PuTTY is the most common client on Windows machines.

Today’s pairs

Pair #1	Pair #2	Pair #3	Pair #4	Pair #5	Pair #6	Pair #7	Pair #8
chandr95, hwillso	hchikssa, ctraugh	clark367, ingle0	cook311, flynn27	shailem, kriveraz	rhiles, wang2852	ljessup, yang1511	rperon, tsuttiyu

If your partner is missing, let me know and we’ll adjust the pairs based on availability.

Using a compute node interactively

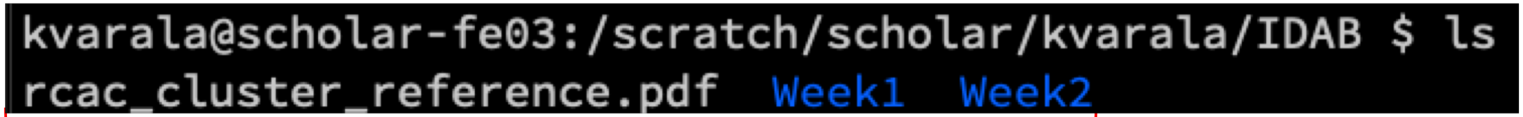
```
1.  ssh <yourID>@scholar.rcac.purdue.edu ↵
```

Log in to the head node.

```
2.  qsub -I -l nodes=1:ppn=1 -l walltime=02:00:00 ↵
```

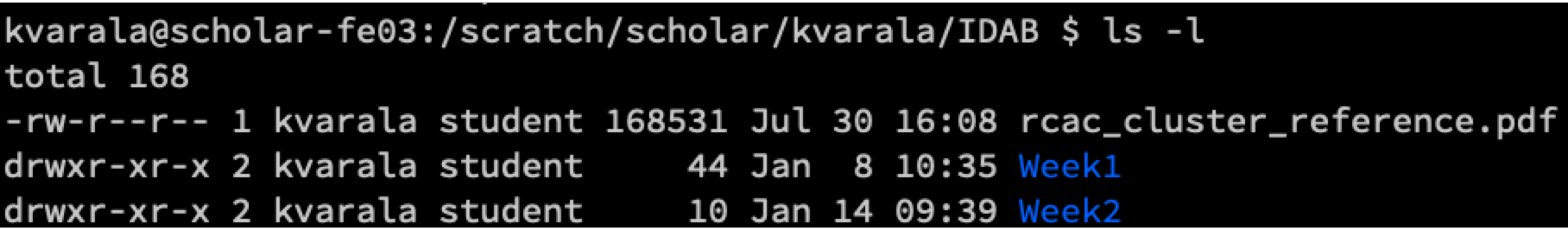
Log in to the compute node to run jobs interactively.

ls on my scratch



```
kvarala@scholar-fe03:/scratch/scholar/kvarala/IDAB $ ls
rcac_cluster_reference.pdf  Week1  Week2
```

Contents of my scratch folder



```
kvarala@scholar-fe03:/scratch/scholar/kvarala/IDAB $ ls -l
total 168
-rw-r--r-- 1 kvarala student 168531 Jul 30 16:08 rcac_cluster_reference.pdf
drwxr-xr-x 2 kvarala student     44 Jan  8 10:35 Week1
drwxr-xr-x 2 kvarala student     10 Jan 14 09:39 Week2
```

Metadata about the contents

Contents of my scratch folder

Metadata on files and directories

- Metadata is information about the file that are not part of the contents of the file.
- Three main parts to it:
  - Ownership and access permissions
  - Size
  - Timestamp

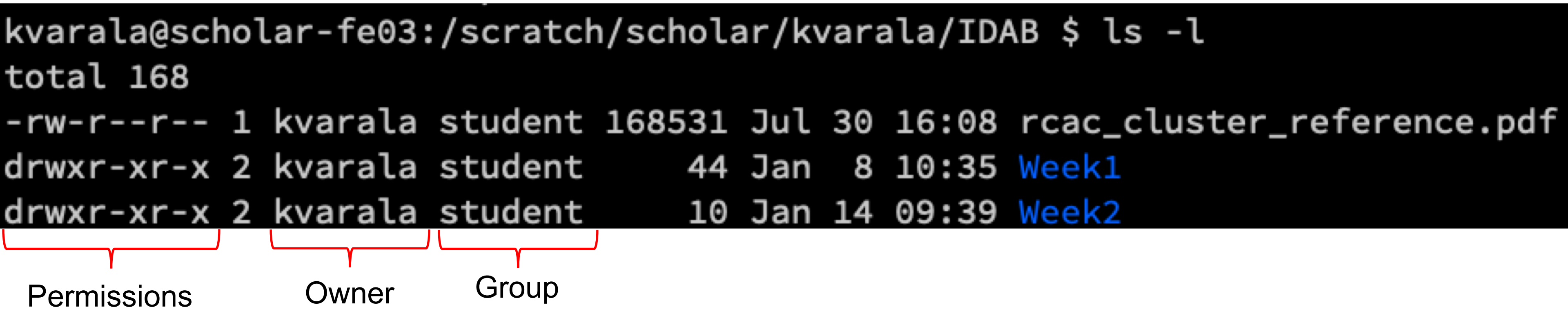
UNIX permissions

- Execute == x == 1 (Means traverse for directories)
- Write == w == 2
- Read == r == 4

Common Permission settings	Indicator	Numeric code
Read-only	r--	4
Read & execute	r-x	5
Read & write	rw-	6
Read, write, execute	rwx	7
??	??	3

Ownership and Access

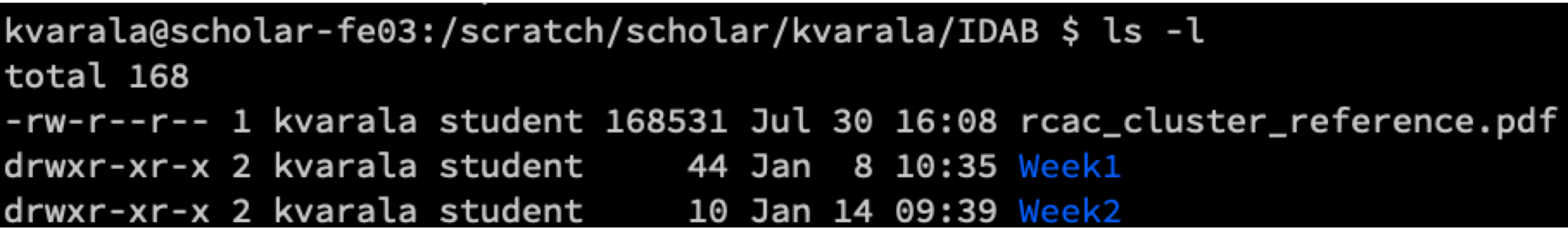
- Every file/directory has a defined owner, which is one user.



```
kvarala@scholar-fe03:/scratch/scholar/kvarala/IDAB $ ls -l
total 168
-rw-r--r-- 1 kvarala student 168531 Jul 30 16:08 rcac_cluster_reference.pdf
drwxr-xr-x 2 kvarala student     44 Jan  8 10:35 Week1
drwxr-xr-x 2 kvarala student     10 Jan 14 09:39 Week2
```

- Owner controls who can access the file/directory by setting the permissions.
- Each user is a part of one or more groups. Each file belongs to one of the groups that the user belongs to.

UNIX permissions



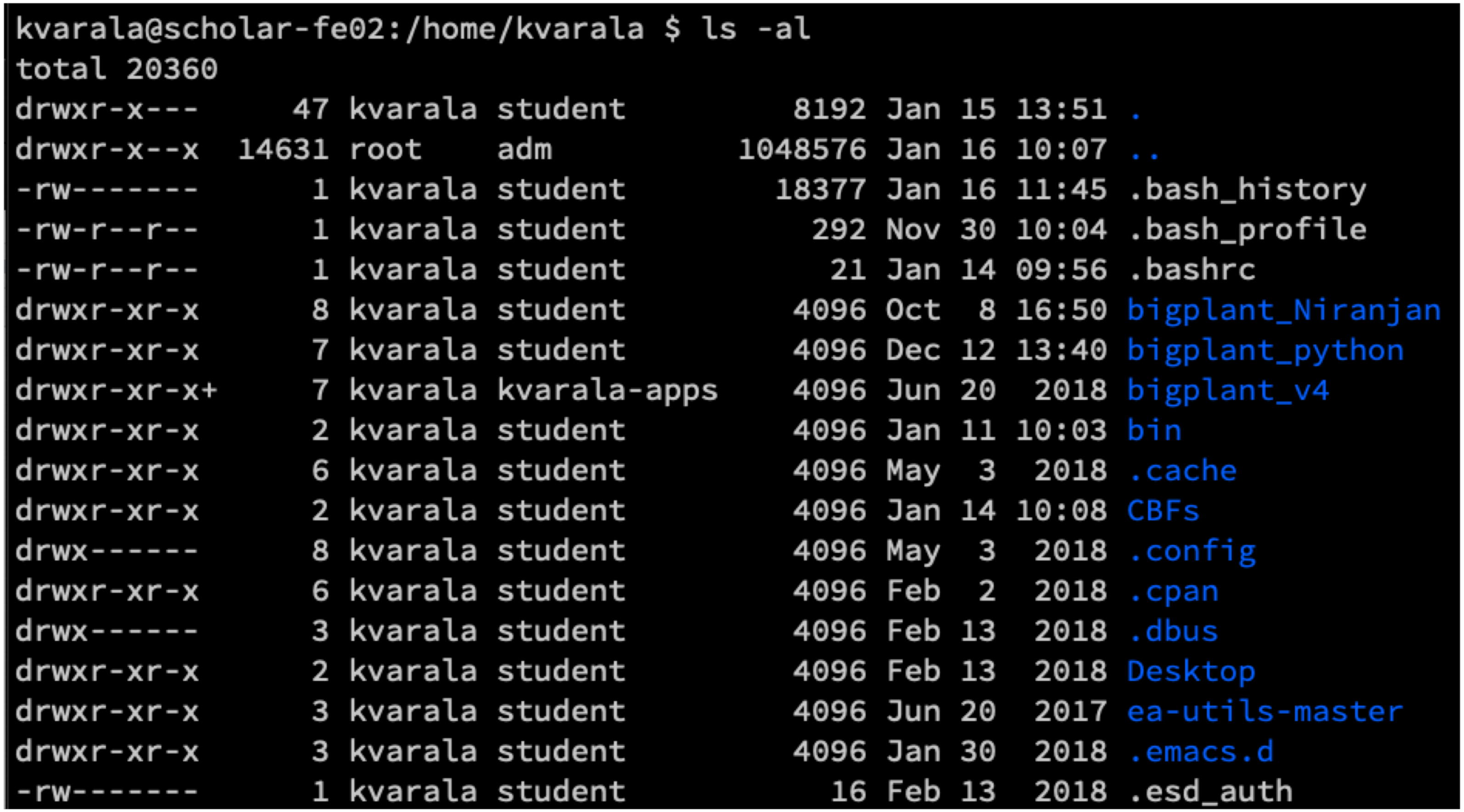
```
kvarala@scholar-fe03:/scratch/scholar/kvarala/IDAB $ ls -l
total 168
-rw-r--r-- 1 kvarala student 168531 Jul 30 16:08 rcac_cluster_reference.pdf
drwxr-xr-x 2 kvarala student     44 Jan  8 10:35 Week1
drwxr-xr-x 2 kvarala student     10 Jan 14 09:39 Week2
```

Permissions

- First character is - for a file and d for a directory.
- Characters 2-4 refer to permissions the owner sets for himself.
- Characters 5-7 are permissions for the group listed.
- Characters 8-10 are permissions for the world (i.e., every other user)

Common Permission settings	Indicator	Numeric code
Read-only	r--	4
Read & execute	r-x	5
Read & write	rw-	6
Read, write, execute	rwx	7

Hidden files and directories



```
kvarala@scholar-fe02:/home/kvarala $ ls -al
total 20360
drwxr-x---     47 kvarala student          8192 Jan 15 13:51 .
drwxr-x--x  14631 root    adm          1048576 Jan 16 10:07 ..
-rw-------      1 kvarala student         18377 Jan 16 11:45 .bash_history
-rw-r--r--      1 kvarala student           292 Nov 30 10:04 .bash_profile
-rw-r--r--      1 kvarala student            21 Jan 14 09:56 .bashrc
drwxr-xr-x      8 kvarala student          4096 Oct  8 16:50 bigplant_Niranjan
drwxr-xr-x      7 kvarala student          4096 Dec 12 13:40 bigplant_python
drwxr-xr-x+     7 kvarala kvarala-apps     4096 Jun 20  2018 bigplant_v4
drwxr-xr-x      2 kvarala student          4096 Jan 11 10:03 bin
drwxr-xr-x      6 kvarala student          4096 May  3  2018 .cache
drwxr-xr-x      2 kvarala student          4096 Jan 14 10:08 CBFs
drwx------      8 kvarala student          4096 May  3  2018 .config
drwxr-xr-x      6 kvarala student          4096 Feb  2  2018 .cpan
drwx------      3 kvarala student          4096 Feb 13  2018 .dbus
drwxr-xr-x      2 kvarala student          4096 Feb 13  2018 Desktop
drwxr-xr-x      3 kvarala student          4096 Jun 20  2017 ea-utils-master
drwxr-xr-x      3 kvarala student          4096 Jan 30  2018 .emacs.d
-rw-------      1 kvarala student            16 Feb 13  2018 .esd_auth
```

Files or directories whose name starts with . are considered hidden so ls does not list them

Working with directories

- `pwd` -> lists the present working directory
- `mkdir` -> makes a new directory
- `cd` -> change directory
- `rmdir` -> remove directory
- Try using cd with path:
  - `cd /scratch/scholar/kvarala/IDAB`
  - `cd ./Week1`
  - `cd ../Week2`

File commands

- `mv` is the move command that moves a file. This command is also used for renaming files.
- `rm` is the remove command and will remove the file or empty directory listed as argument.
- `cat` is the concatenate command that joins the contents of all files given as arguments.

Exercises

Basic commands

History: Keeping a record

- First let’s make sure we keep a record of all the commands we use.
- In the Bash shell (your default shell on scholar) every command you enter is stored in memory while shell is active.
- When you quit the shell, this ‘`history`’ is stored in a file called .bash_history that is in your home folder.
- Default size for history in memory and in the file is 1000 commands.
- You can recover your history of commands by typing: `history` in your command line.

Customizing your shell

- In the Bash shell, you can add aliases or modify PATH using the .bash_profile file.
- ~/.bash_profile is read every time a new shell is created.
- Create a new file called bash_profile.txt on your local computer and add the following two lines to it:

```
alias ltr='ls -ltr’
alias scratch= ’cd /scratch/scholar/k/kvarala’

export PATH=$PATH:$HOME/bin
```

- Now copy this file to your home folder on Scholar using your SCP client.

Creating and working with directories

```
cd /scratch/scholar/<YourID>/

mkdir Lab2_Exercises

cd Lab2_Exercises
```

Copy file

- Copy file from /scratch/scholar/kvarala to your scratch: Absolute path
- `cp /scratch/scholar/kvarala/IDAB/rcac_cluster_reference.pdf /scratch/scholar/<YourID>/Lab2_Exercises`
- `ls /scratch/scholar/<YourID>/Lab2_Exercises`
- `ls -l /scratch/scholar/<YourID>/Lab2_Exercises`

Relative path

- `cd /scratch/scholar/<YourID>/Lab2_Exercises`
- `cp ../kvarala/IDAB/rcac_cluster_reference.pdf .`
- Now try the `cp` command with the `-p` switch

Move file

- Move file from your scratch to your home:
  Absolute path
- `cd /scratch/scholar/<YourID>/Lab2_Exercises`

-
  ```
  mv /scratch/scholar/<YourID>/Lab2_Exercises
  /rcac_cluster_reference.pdf /home/<yourID>
  ```
- `ls -l /scratch/scholar/<YourID>/`
- `ls -l /home/<YourID>/`

Relative path

- `cd /scratch/scholar/<YourID>/Lab2_Exercises`
- `mv rcac_cluster_reference.pdf ~/`
- `mv ~/rcac_cluster_reference.pdf .`
- `ls -l /scratch/scholar/<YourID>/Lab2_Exercises`
- `ls -l /home/<YourID>/`

Customizing your shell

- Now move the bash_profile.txt file in your home directory to .bash_profile
- `mv bash_profile.txt ~/.bash_profile`
- Type the command: `ltr`
- logout of the shell and log back in.
- Now, again type the command: `ltr`

Remove file

- Remove file from your scratch:

Absolute path

- `cp /scratch/scholar/<YourID>/Lab2_Exercises/rcac_cluster_reference.pdf /scratch/scholar/<yourID>/temporary_copy`
- `ls -l /scratch/scholar/<YourID>/Lab2_Exercises`
- `rm /scratch/scholar/<YourID>/Lab2_Exercises/temporary_copy`
- `ls -l /scratch/scholar/<YourID>/`

Relative path (be VERY careful)

- `cd /scratch/scholar/<YourID>/`
- `cp rcac_cluster_reference.pdf ~/temporary_copy`
- `rm ~/temporary_copy`
- `ls -l /home/<YourID>/`

Viewing files

```
cd /scratch/scholar/<YourID>/Lab2_Exercises
cp -pr /scratch/scholar/kvarala/IDAB/Week2/Files .
ls –l
cd Files
cat Pasture.txt
cat WoodPile.txt
less North_of_Boston.txt
   quit less display with the ‘q’ key
head North_of_Boston.txt
tail North_of_Boston.txt
```

Changing file permissions

```
cd /scratch/scholar/<YourID>/Lab2_Exercises/Files
ls -l Pasture.txt

chmod 755 Pasture.txt
ls -l Pasture.txt

chmod -x Pasture.txt
ls -l Pasture.txt

chmod 666 Pasture.txt
ls -l Pasture.txt

chmod -w Pasture.txt
ls -l Pasture.txt
```

I/O streams

- Each command has 3 Input/Output streams:
  - STDIN : Standard Input is the default stream that inputs data into a command. Example: keyboard, file etc.
  - STDOUT : Standard Output is the default output stream of the command. Example: Terminal
  - STDERR: Standard Error is where the errors from the program are displayed: Example: Terminal

Creating pipelines from commands

- The STDIN and STDOUT of each command can be redirected to combine programs together.

- For example, the STDOUT of one program can be sent to the STDIN of another program.

- STDIN and STDOUT streams can be redirected using the following symbols:
  1. `>`
  2. `<`
  3. `|`

Redirecting STDIN and STDOUT

```
cat North_of_Boston.txt | less
```

Here, the STDOUT of `cat` is sent to the STDIN of `less.`

```
cat Pasture.txt > New_Pasture.txt
```

Here, the STDOUT of `cat` is sent to a new file called New_Pasture.txt

```
less < New_Pasture.txt
```

Here the file New_Pasture.txt is sent to the STDIN of less

```
cat Pasture.txt WoodPile.txt >2Poems.txt
```

Capturing STDERR

Errors from running commands are sent to STDERR. By default, STDERR is shown on your monitor.

```
nonsense_command
-bash: nonsense_command: command not found
```

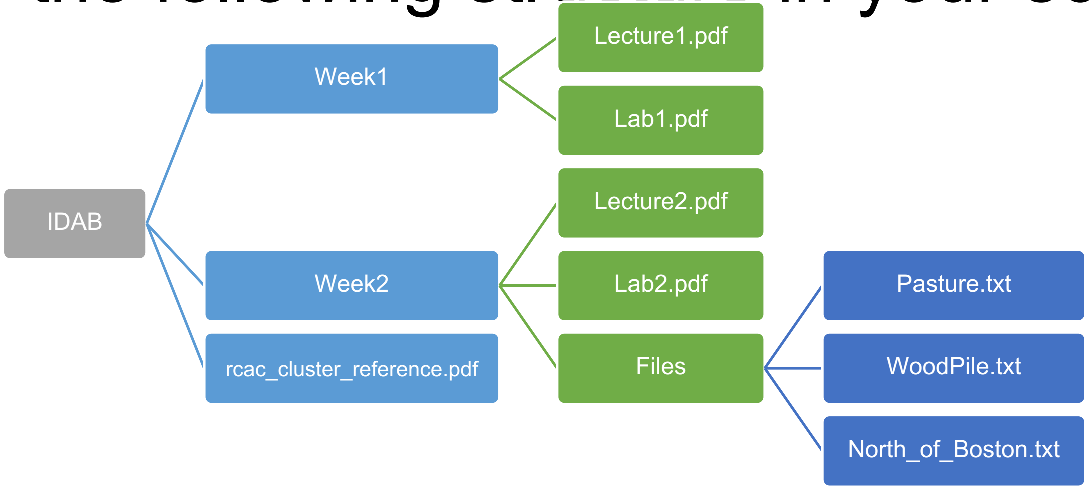
Here, the STDERR of `nonsense_command` is sent to the monitor

```
nonsense_command 2>err.log
```

Here the file STDERR of `nonsense_command` is sent to the file err.log

Try these on your own

- Try moving to these locations with the cd command:
  - /home/<yourID>
  - /scratch/scholar/kvarala/IDAB/
  - /usr/bin/
  - /root/
- Create the following structure in your scratch space:



Learning about UNIX commands

- `which <cmd>`
  - Tells you the location of the command
- `man <cmd>`
  - Displays the manual for the command
- `<cmd> --help/-h`
  - Displays a short list of options for the command