Introduction to Linux

M Tech CS – I
2015-16

Arijit Bishnu
Debapriyo Majumdar
Sourav Sengupta
Mandar Mitra

Login, Logout, Change password

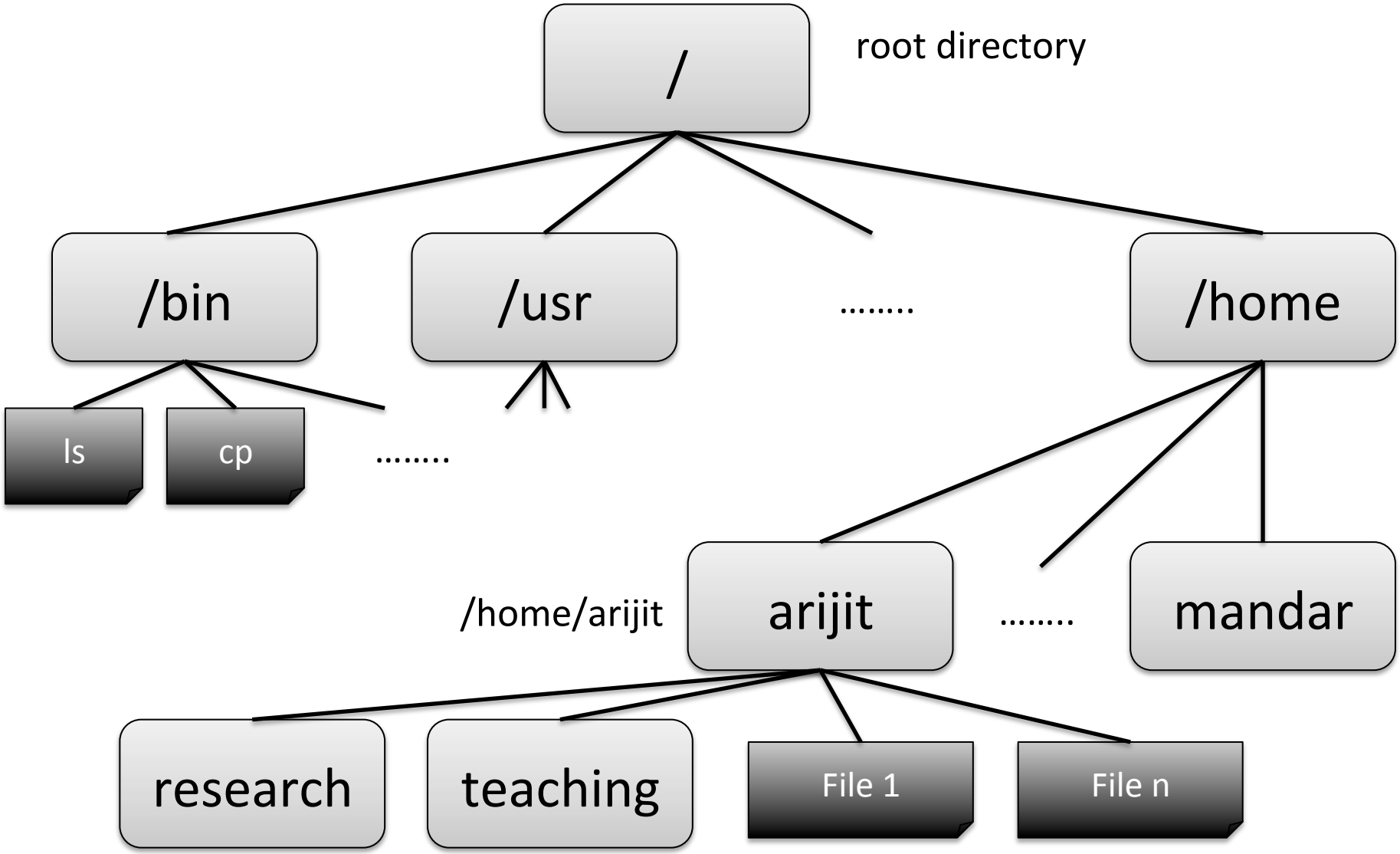
```
$ ssh, ssh –X
```

secure shell

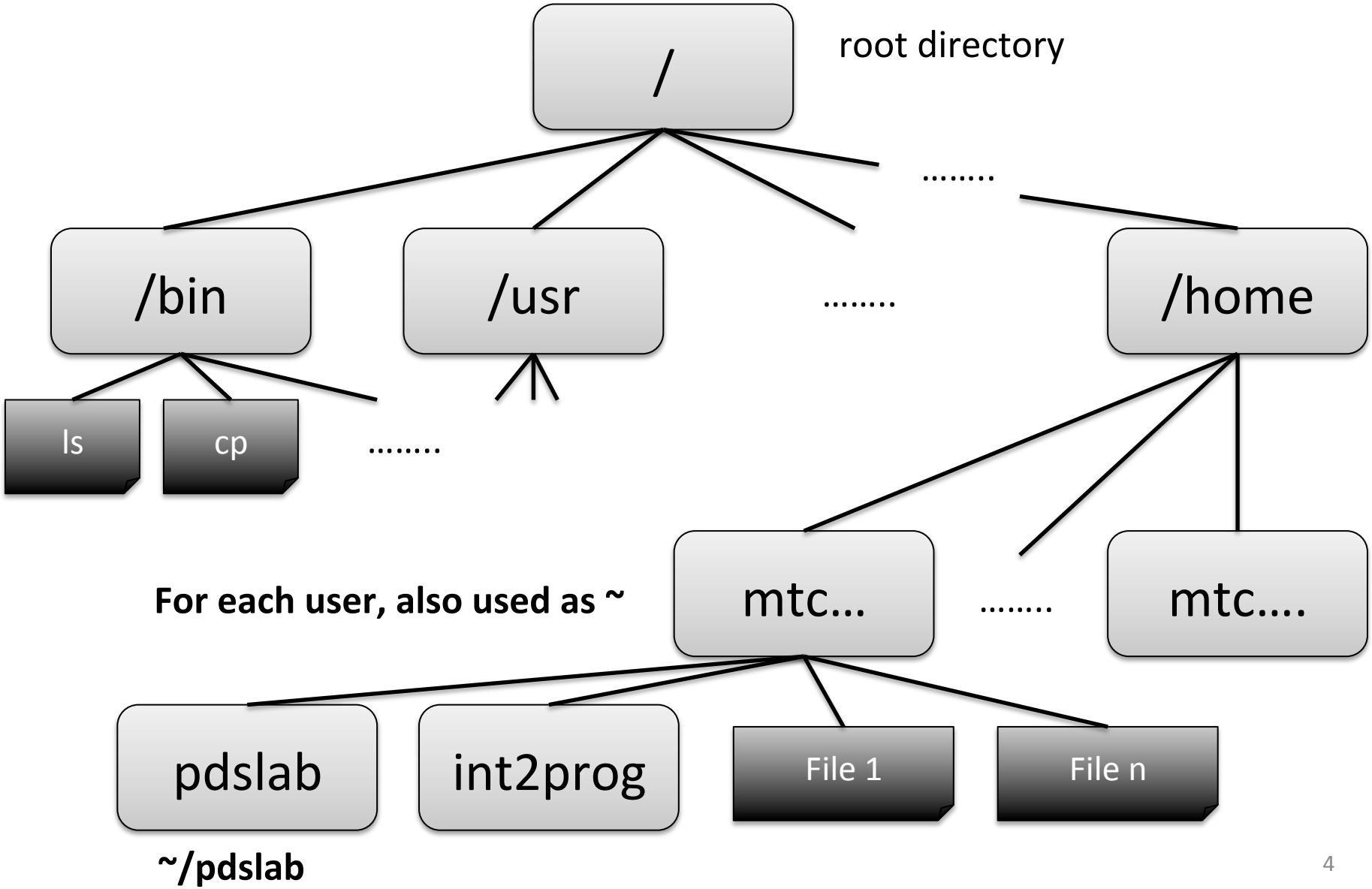
```
$ ssh www.isical.ac.in
$ ssh 192.168…

$ logout, exit, ^d

$ passwd
```

change password

Files and directories (folders)



Files and directories (folders)



Files and directories

```
$ pwd
```

Present working directory

```
~, pdslab/day3/…
```

Relative path

```
/home/username, /matlab/…/…
```

Absolute path

Files and directories

`.`

The current directory

`..`

The parent directory

Use of relative path

```
$ cd ~/pdslab/day2
```

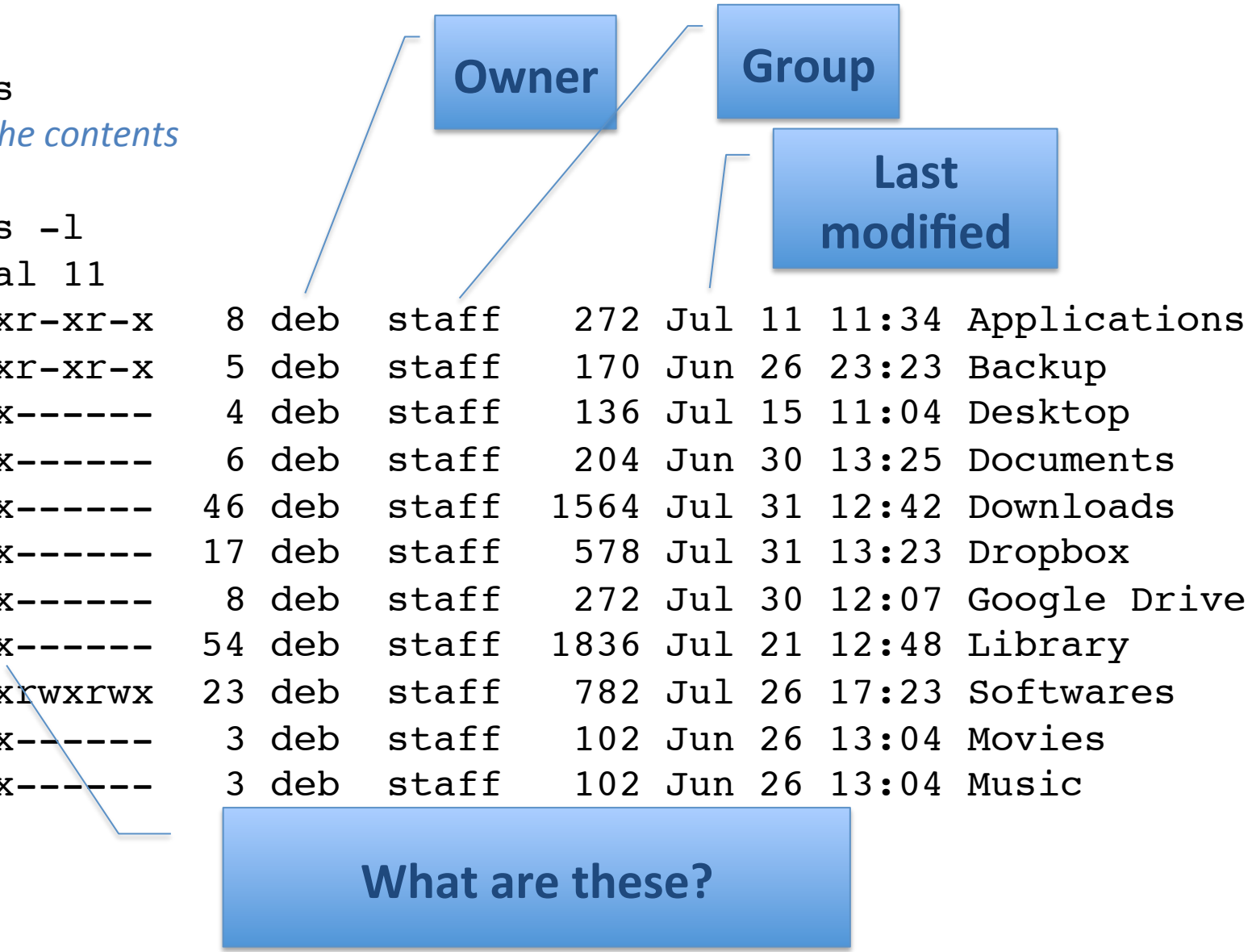
What will happen if we type --

```
$ cd day3
$ cd ./day3
$ cd ../day3
```

Files and directories

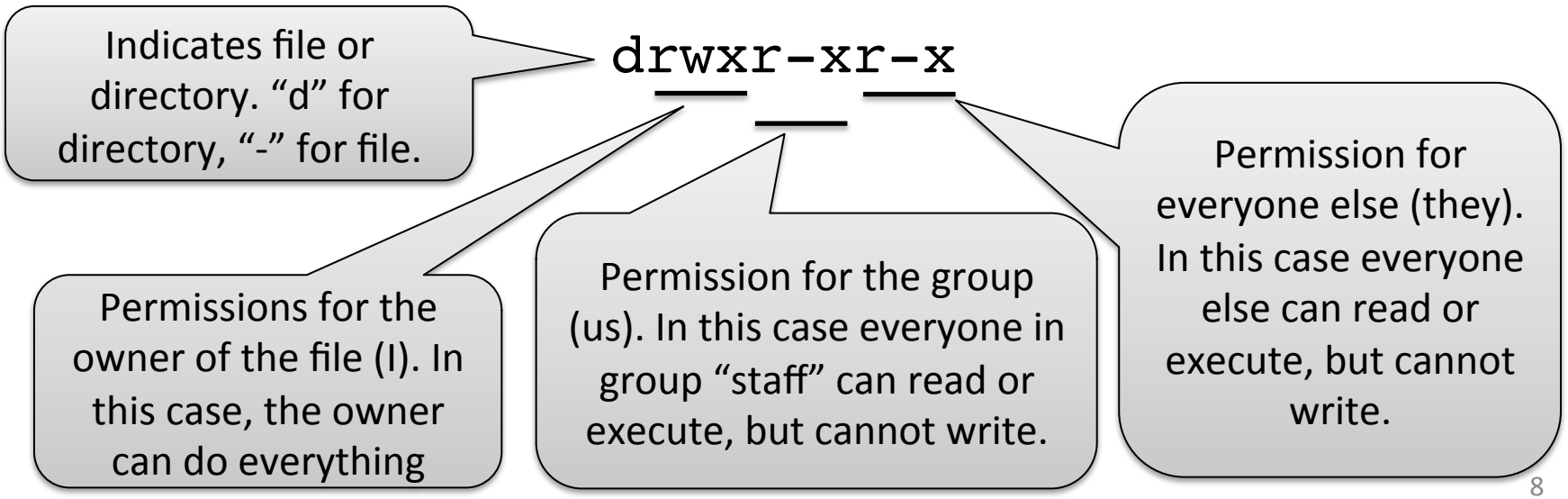
```
$ ls
```
List the contents

```
$ ls -l
Total 11
drwxr-xr-x    8 deb   staff    272 Jul 11 11:34 Applications
drwxr-xr-x    5 deb   staff    170 Jun 26 23:23 Backup
drwx------    4 deb   staff    136 Jul 15 11:04 Desktop
drwx------    6 deb   staff    204 Jun 30 13:25 Documents
drwx------   46 deb   staff   1564 Jul 31 12:42 Downloads
drwx------   17 deb   staff    578 Jul 31 13:23 Dropbox
drwx------    8 deb   staff    272 Jul 30 12:07 Google Drive
drwx------   54 deb   staff   1836 Jul 21 12:48 Library
drwxrwxrwx   23 deb   staff    782 Jul 26 17:23 Softwares
drwx------    3 deb   staff    102 Jun 26 13:04 Movies
drwx------    3 deb   staff    102 Jun 26 13:04 Music
```



Permissions

- Three types of permissions
  - Read (r): Can read the contents of a file or see the contents (list of files) in a directory
  - Write (w): Can edit a file, can put a file in the directory or delete a file in that directory
  - Execute (x): Can execute an executable file. Can go inside (cd) a directory.
- Groups: I, we and they



Change permission (chmod)

```
$ ls -l
Total 4
drwxr-xr-x   8 deb  staff   272 Jul 11 11:34 Applications
drwxr-xr-x   5 deb  staff   170 Jun 26 23:23 Backup
drwx------   4 deb  staff   136 Jul 15 11:04 Desktop
-rwx------   6 deb  staff   204 Jun 30 13:25 Notes.txt
```

```
$ chmod g+w filename
```

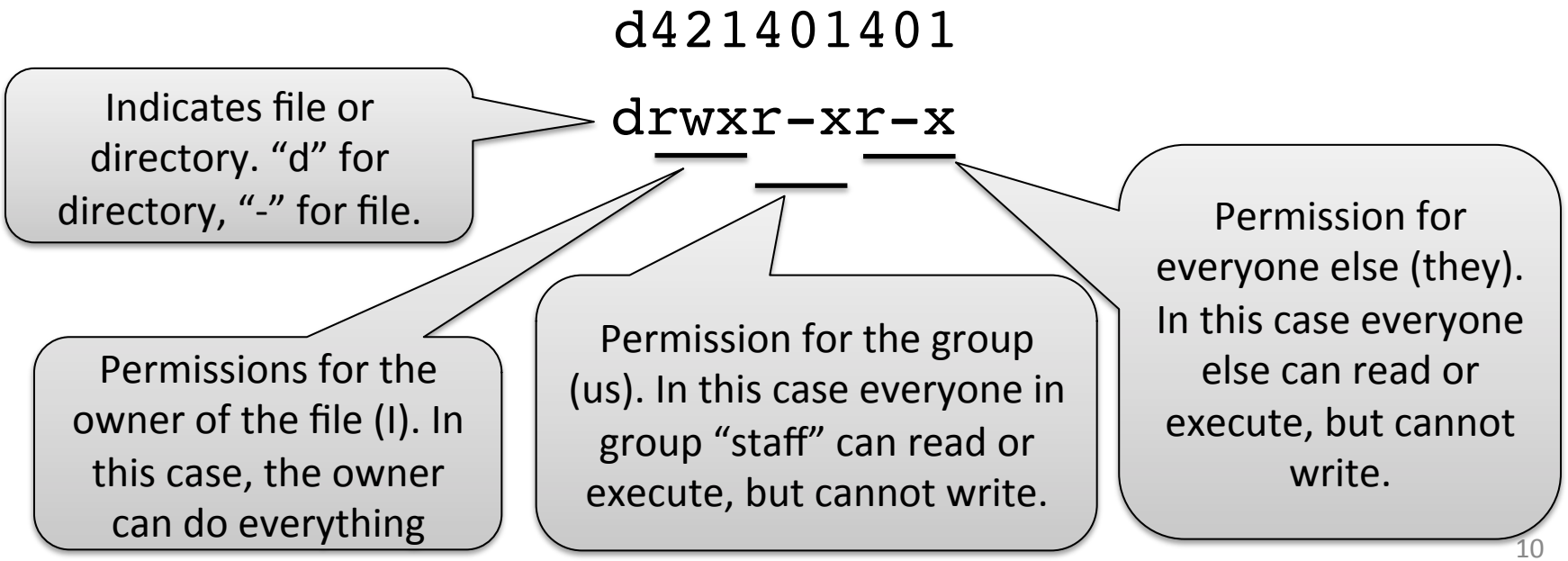
Give the group "w" (write) permission

```
$ chmod o-rwx filename
```

Remove read, write and execute permission from "others"

Permissions: make things easier

- Three types of permissions: assign bit values to each
  - Read (r): 3rd bit = 4
  - Write (w): 2nd bit = 2
  - Execute (x): 1st bit = 1
- Example: rwx = 7, rw- = 6, …



Change permission (chmod)

```
$ ls -l
Total 4
drwxr-xr-x   8 deb  staff   272 Jul 11 11:34 Applications
drwxr-xr-x   5 deb  staff   170 Jun 26 23:23 Backup
drwx------   4 deb  staff   136 Jul 15 11:04 Desktop
-rwx------   6 deb  staff   204 Jun 30 13:25 Notes.txt
```

```
$ chmod g+w filename
```

Give the group "w" (write) permission

```
$ chmod o-rwx filename
```

Remove read, write and execute permission from "others"

```
$ chmod 700 filename
```

All permissions to the user, nothing to the group and others

Create and remove (delete)

```
$ mkdir directoryName
```
MAKE the directory

```
$ rmdir directoryName
```
REMOVE the directory

You cannot REMOVE a non-empty directory

```
$ rm fileName
```
REMOVE the file

```
$ rm –i, rm –f
```
REMOVE interactively, or by FORCE!!

Contents of a file

```
$ more filename
$ less filename
```

less is less work. ☺

```
$ cat filename
```

From the word concatenate – prints the whole content onto the screen.

```
$ head filename
```

Head of the file. Also try head -10, head -20, …

```
$ tail filename
```

Tail of the file. Also try tail -10, tail -30, …

Copy, move

```
$ cp file copy_of_file
```

Copy the content of the file

```
$ mv file different_name
```

Rename / move the file.

Which one should take less time for big files?

```
$ cp file1 file2
$ rm file1
```

mv does not copy and delete the older file, simply changes the pointer

What am I executing?

```
$ ls
```

Where is this ls?

```
$ which ls
/usr/bin/ls
```

/usr/bin is in the path

```
$ prog1.o
$ ./prog1.o
```

If . (dot, the current directory) is in the path, then the first command will work. To ensure that you are executing the program in your current directory, and not something else, specify the location

Input, output

```
$ echo “Hi this is me!”
```
Prints “Hi this is me!”

```
$ echo “Hi this is me!” > test.txt
```
Redirects the output to the file. If the file exists, it is overwritten.

```
$ ./prog1.o > output.txt 2> error.txt
```
Redirect STDOUT and STDERR

```
$ ./prog1.o < in.txt > out.txt 2> err.txt
```
Take input from in.txt

```
$ echo “1 2 3 4 5” | ./prog1.o
```
Print “1 2 3 4 5” and use that as the input for the program

Exercise

- Suppose there is a file <file> containing 100 lines. Using **head** and **tail**, show the lines 91-95 only.

- You have a directory named “pdslab” and under that you have a file named “sharedNotes.txt”. You want to allow your friends (same group) to read the notes, but do not want them to see what other files are present in the pdslab directory. What permissions should you set for your home directory, pdslab directory and the file sharedNotes.txt?

Summary of commands

Command	Use
pwd	Prints the present working directory
cd <dir>	Change directory (go to) to <dir>
mkdir <dir>	Create a directory <dir> under the present working directory
man <command>	Manual page for the command <command>. Press “q” to quit.
ls	List contents of the present working directory
ls -la, ls -l, …	List contents, with details. By default hidden files (starting with a .) are not shown. The “-a” option shows all files including hidden files.
ls -l <dir>	List contents of the <dir>. Default is present working directory.
cat <file>	Show contents of the file within the terminal

Summary of commands

Command	Use
passwd	Change password
cat <file>	Print contents of the file within the terminal, all at one go.
more <file>, less <file>	Print contents of the file on the terminal, allows you to go back and forward. Learn the difference between more and less yourself. Press “q” to quit.
cp <source_file> <destination_file>	Creates a copy of <source_file> to <destination_file>
rm <file>	Remove (delete) <file>. Use -i option for interactive delete.
mv <file> <destination_file>	Move file to another name.
mv <file> <destination_directory>	Move <file> from the current position to the <destination_directory>

Summary of commands

Command	Use
chmod <arg> <filename>	Change permissions for the file (or directory)
which <command>	Show the full path of the command
head <file>	Show the top (default 10) lines of <file>. Use head -20 <file> for showing 20 lines, for example
tail <file>	Show the last (default 10) lines of <file>. Similar to head, use tail -N for any number N.
wc <file>	Show the number of lines, words of a file. Use options -l, -w etc for specific outputs.
sort <file>	Sort the lines of the file by lexicographic order (default). Sort is a very powerful tool with many options, learn later.

For another day

```
$ grep
$ find

$ diff
$ sort

$ ps
$ top

$ sed
$ awk
```