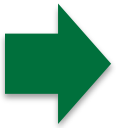
CS 50:
Software Design and Implementation

Introduction

Agenda



1. Course overview

2. Command line

Introductions

- My background
- Learning fellows
  - Something challenging about CS50
  - Something helpful

This course is roughly organized into five main sections

- Shell
- Commands
- Shell programming

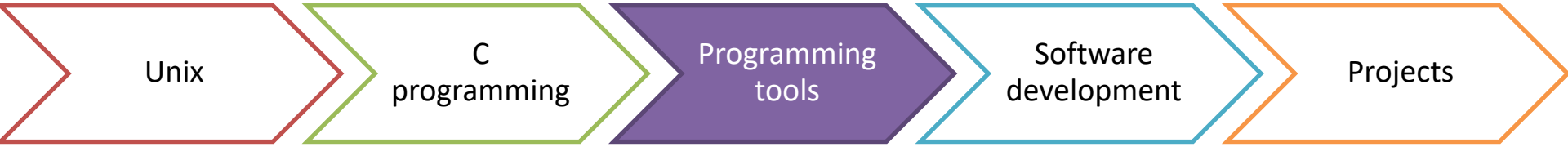
This course is roughly organized into five main sections

- Shell
- Commands
- Shell programming

- Structure
- Arrays
- Pointers
- Dynamic memory
- Files

Why use C when we've got Python?

This course is roughly organized into five main sections



- Shell
- Commands
- Shell programming

- Structure
- Arrays
- Pointers
- Dynamic memory
- Files

- gcc
- make
- gdb
- git
- valgrind

This course is roughly organized into five main sections

- Shell
- Commands
- Shell programming

- Structure
- Arrays
- Pointers
- Dynamic memory
- Files

- gcc
- make
- gdb
- git
- valgrind

- Design documentation
- Implementation
- Unit test and debugging
- Integration
- Maintenance

This course is roughly organized into five main sections

- Shell
- Commands
- Shell programming

- Structure
- Arrays
- Pointers
- Dynamic memory
- Files

- gcc
- make
- gdb
- git
- valgrind

- Design documentation
- Implementation
- Unit test and debugging
- Integration
- Maintenance

- Search engine
  - Crawler
  - Indexer
  - Query engine
- Team project

This course is roughly organized into five main sections

- Shell
- Commands
- Shell programming

- Structure
- Arrays
- Pointers
- Dynamic memory
- Files

- gcc
- make
- gdb
- git
- valgrind

- Design documentation
- Implementation
- Unit test and debugging
- Integration
- Maintenance

- Search engine
  - Crawler
  - Indexer
  - Query engine
- Team project

This course involves a large amount of programming!

Goal is for you to build real systems you'll be proud of

We *will* use x-hours for the first few weeks (but optional at the end)

Come to lecture prepared!

Syllabus: http://www.cs.dartmouth.edu/~tjp/cs50

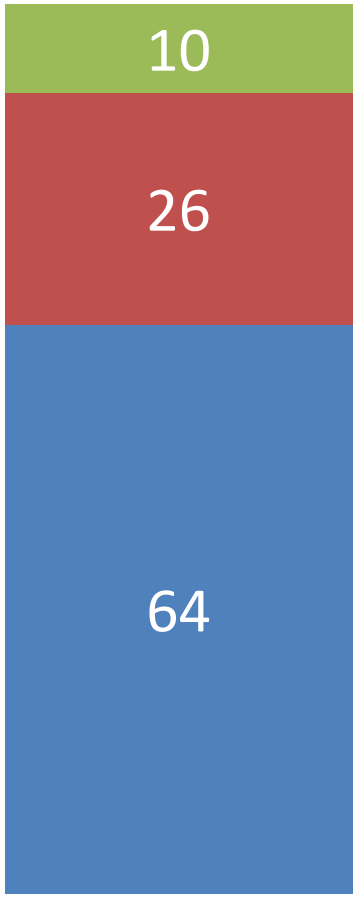
- The Schedule page of the course web has a link to material for that day, **read this material before each class period**
- There are additional reading assignments posted weekly on the Schedule page

Learning Fellows are here to help!

- I plan to spend roughly half of each class doing group exercises
  - I will give a practice problem and time for your group to work on the problem
  - Afterward I will randomly select one student to present their group's solution to the class
  - We will see there are often many ways to efficiently solve a problem, seeing how someone else solved a problem can often be useful

Grading is comprised of labs, a final team project, and class engagement

ASSESSMENT



Engagement

Final project

- Team score: 90%
- Individual score: 10%

Labs

- Six total labs
- Completed individually
- Submit via github
- Must run on Thayer Linux servers!

NOTE:
There are no exams and no textbook in CS 50

We have a large repository of prior solutions

Write your own code!

We will also be using Canvas, Slack, and Git

Canvas

- Course announcements
- Grades

Slack (access via Canvas)

- Q&A forum
- Ask questions, get answers
- Don’t post code!

Git

- Lab and final project submission

Let me know if you don’t have access!

Lab 0 is out today, complete before midnight

Lab 0

- Find it on Canvas
- Take course survey to understand your background
- Read and acknowledge course policies
- **Complete by midnight tonight**

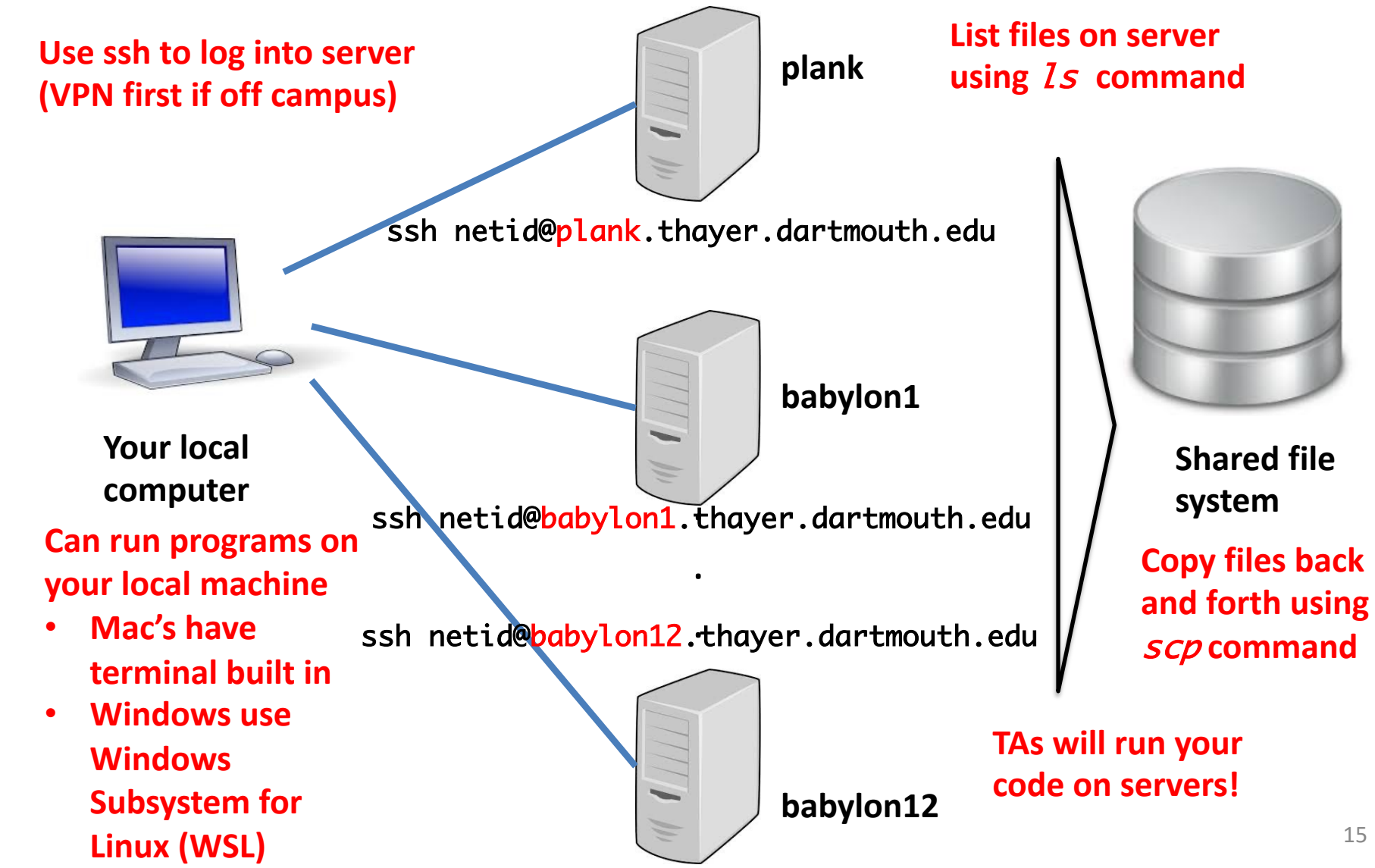
We will use this information to assign you to a group

- Starting tomorrow groups will sit together during lecture and will work on daily problems as a team
- Remember, one of you may be chosen to present your group's solution to the class!

Agenda

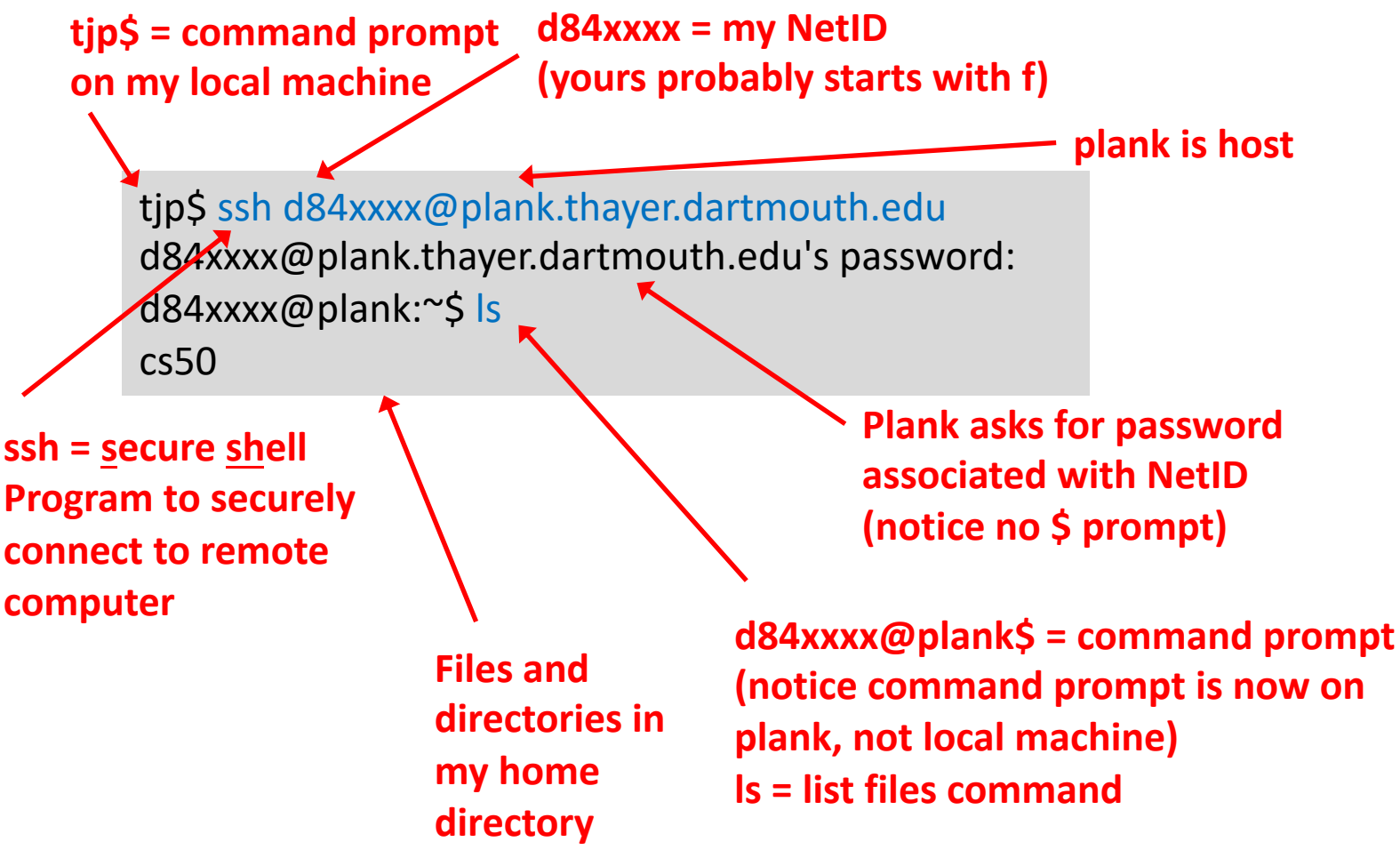
1. Course overview
2. Command line

Dartmouth server overview



DEMO: Secure shell (ssh) to plank server then list files with `ls` command

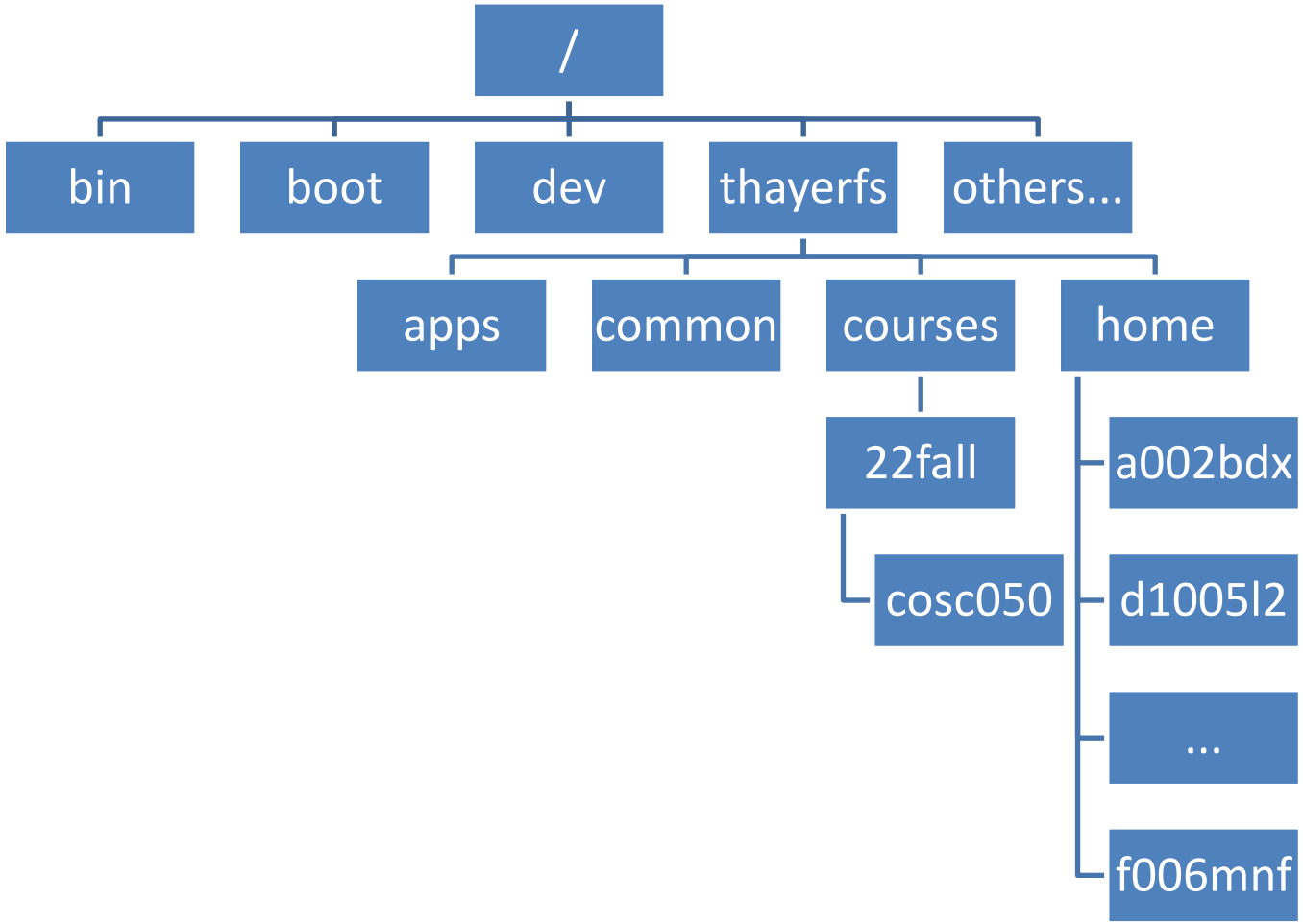
Start terminal program running[1]



[1] I'll be using bash (Bourne-again shell) written by Steve Fox as a replacement for sh written by Steve Bourne

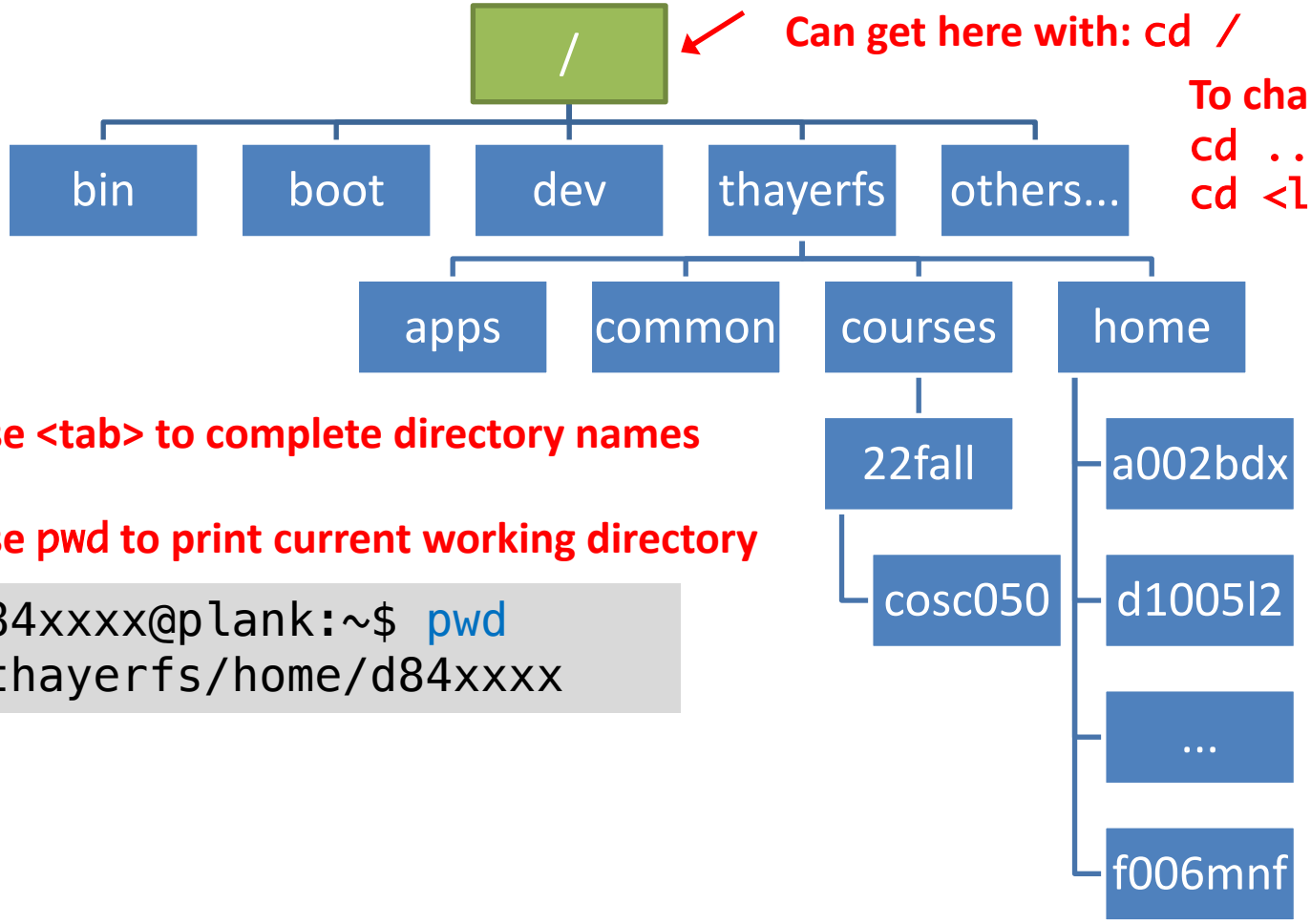
Files are organized like a tree starting at the root

Thayer file system organization



Use cd to change directories and pwd to see where you are

Thayer file system organization



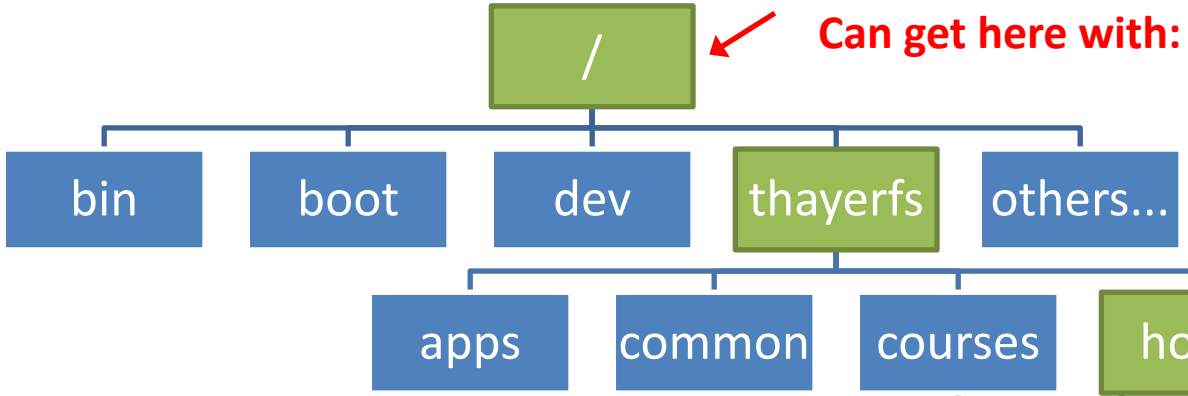
Use <tab> to complete directory names

Use pwd to print current working directory

```
d84xxxx@plank:~$ pwd
/thayerfs/home/d84xxxx
```

Your home directory is located at /thayerfs/home/<netID>

Thayer file system organization



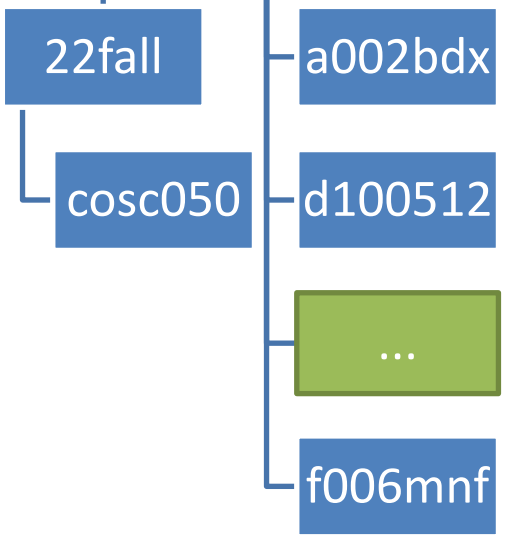
Root
Can get here with: `cd /`

To change directories
`cd .. (up one level)`
`cd <location>`

Use <tab> to complete directory names

Use pwd to print current working directory

```
d84xxxx@plank:~$ pwd
/thayerfs/home/d84xxxx
```

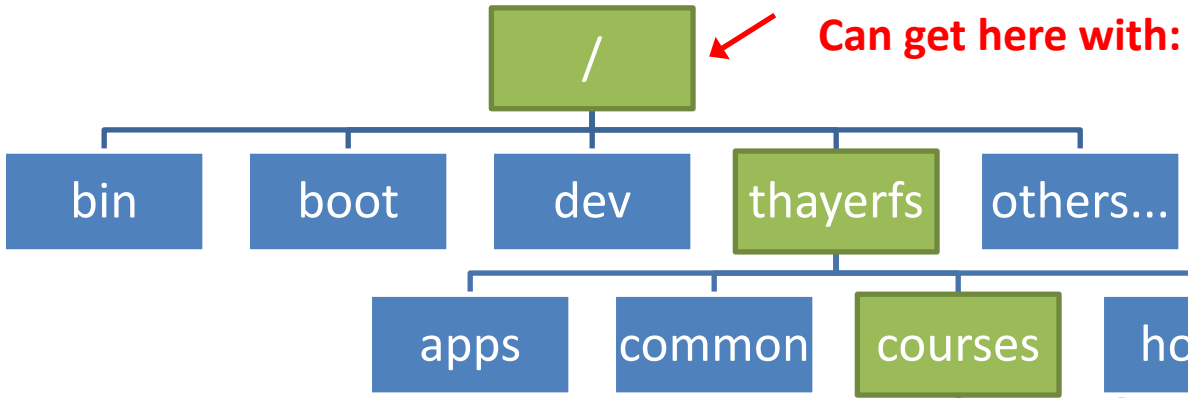


Start at your home directory (netid) after login

Can return to your home directory with:
`cd ~`
or just
`cd`

Course files are located at /thayerfs/courses/22fall/cosc050

Thayer file system organization



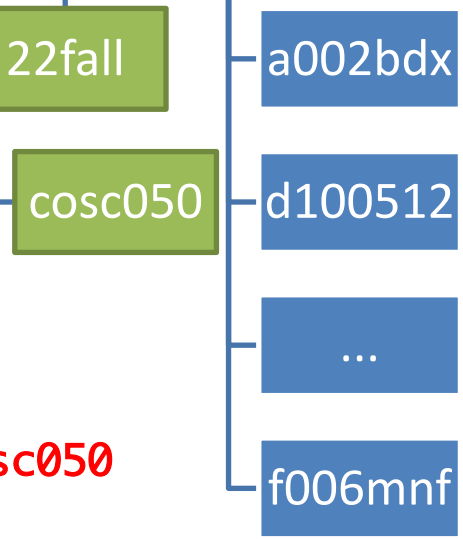
Root
Can get here with: `cd /`

To change directories
`cd .. (up one level)`
`cd <location>`

Use <tab> to complete directory names

Use pwd to print current working directory

```
d84xxxx@plank:~$ pwd
/thayerfs/home/d84xxxx
```



Start at your home directory (netid) after login

CS50 course files are here
`cd /thayerfs/courses/22fall/cosc050`

Can return to your home directory with:
`cd ~`
or just
`cd`

Also nice to know:
Ctrl-a: goes to start of line
Ctrl-e: goes to end of line

Edit *~/.ssh/config* file on your local machine to quickly log in to course servers

Create shortcut to plank

Try to change into directory name .ssh from your local machine's home directory

```
tjp$ cd ~/.ssh
-bash: cd: ~.ssh: No such file or directory
tjp$ mkdir ~/.ssh
tjp$ vi ~/.ssh/config
Type i to go into insert mode
Add lines
     Host plank
       Hostname plank.thayer.dartmouth.edu
       User d84xxx
Type escape then :wq to save (write) and quit vi

Now access plank with
tjp$ ssh plank
```

If no error message, then directory exists so no need to type this, otherwise type it to make a directory called .ssh

Start editor called vi editing file called config and go into insert mode

Add lines, use tab on lines 2 and 3
Use your netid instead of d84xxxx
Indentation not required

Save file (don't forget to type esc first!)

Instead of ssh <netid>@plank.thayer.dartmouth.edu

[1] I'll primarily use vi as an editor, but you can choose the editor you like best

scp securely copies files between computers

Copy from local computer to server

plank

scp FROM TO

Your local computer

scp <file> <netid>@plank.thayer.dartmouth.edu:~<location>

```
tjp$ echo "test test test" > test.txt
tjp$ cat test.txt
test test test
tjp$ scp test.txt d84xxxx@plank.thayer.dartmouth.edu:~/cs50/labs
d84xxxx@plank.thayer.dartmouth.edu's password:
test.txt                                      100%   15     2.0KB/s   00:00
```

echo is like a print command, prints what you type, > sends output to file called test.txt

cat types file contents to screen

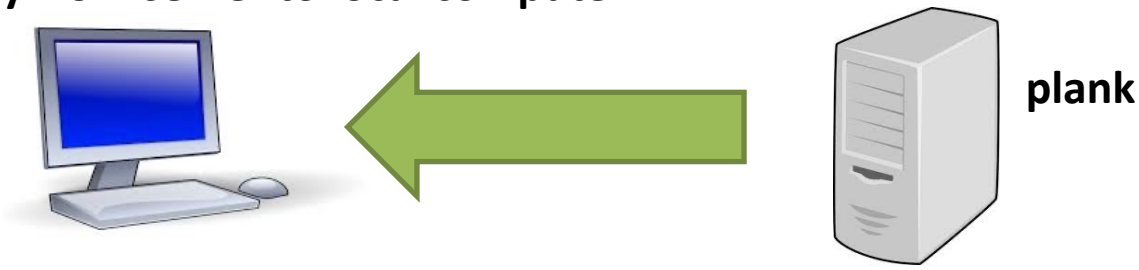
scp securely copies test.txt to directory called labs on plank

cat in labs directory on plank types content of copied file

```
d84xxxx@plank:~/labs$ cat test.txt
test test test
```

scp securely copies files between computers

Copy from server to local computer



Your local computer

scp <netid>@plank.thayer.dartmouth.edu:<file> .

```
tjp$ scp d84xxx@plank.thayer.dartmouth.edu:~/cs50/labs/test.txt .
d84xxxx@plank.thayer.dartmouth.edu's password:
test.txt                                      100%   15     5.7KB/s   00:00
```

From local computer

Give id on name of server and file name/location

Dot (.) means copy to current location on local machine

In both directions the simplest approach is for the copy operation to start on your local machine (the server does not have a way of locating your computer)

If you forget how a command works, use the man (manual) command

tjp$ man scp

```
SCP(1)                                                    BSD General Commands
Manual                                                                SCP(1)


NAME
     scp — OpenSSH secure file copy


SYNOPSIS
     scp [-346ABCOpqRrsTv] [-c cipher] [-D sftp server path] [-F ssh config] [-i identity file] [-J destination]
         [-l limit] [-o ssh option] [-P port] [-S program] source ... target


DESCRIPTION
     scp copies files between hosts on a network.


     It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as a login
ses-
     sion.


     scp will ask for passwords or passphrases if they are needed for authentication.


     The source and target may be specified as a local pathname, a remote host with optional path in the form
     [user@]host:[path], or a URI in the form scp://[user@]host[:port][/path].  Local file names can be made explicit
     using absolute or relative pathnames to avoid scp treating file names containing ‘:’ as host specifiers.


     When copying between two remote hosts, if the URI format is used, a port cannot be specified on the target if
```

Common commands
q to quit
space/f to page forward
b page back
/<text> to search
n next search
N prev search

Use `man –k <term>` to find man pages containing term

Summary of commands

Command	Name	Use
ssh	Secure Shell	Securely connect to a remote computer (replaced insecure, older, command called telnet) `ssh <netid>@plank.thayer.dartmouth.edu`
cd	Change Directory	Move around file system `cd ..` or `cd <location>` or `cd ~` or `cd /`
pwd	Print Working Directory	Print the current directory on screen
mkdir	Make Directory	Create a new directory `mkdir ~/tempdir`
vi	Visual Instrument	Popular text editor; others include emacs, vim, sublime
echo	Echo	Print to screen `echo "hello world"`

cat	Concatenate	View contents of one or more files
		`cat test.txt`
		`cat test.txt test2.txt`

scp	Secure Copy	Securely copy files between computers `scp <file> <netid>@plank.Thayer.Dartmouth.edu:~/labs`
man	manual	Instructions on how to use commands `man scp`

TODO: Before class tomorrow, complete Lab 0 and get access to plank

Lab 0

Make sure you can access plank
Windows users might install WSL

- Find it on Canvas
- Take course survey to understand your background
- Read and acknowledge course policies
- **Complete by midnight tonight**

We will use this information to assign you to a group

- Starting tomorrow groups will sit together during lecture and will work on daily problems as a team
- Remember, one of you may be chosen to present your group's solution to the class!