Lecture 1.2 – How Computers Work

Specific Learning Objectives:

1.1.8 – Understand the concept of working directories.

2.1.1 – Understand how file systems are structured and organized.

2.1.2 – Understand how to navigate file systems using a GUI interface.

2.1.3 – Use directories to organize course work.

Tips for Flipped Lectures

- Have a notepad and pen handy so you can write down any questions! Anything that seems confusing, you don't totally understand, anything you might be wondering about. **We'll always start the in-class portion with time for questions from lecture material!**

- Be sure to **pause the video** ⏸️ and work through the Check Your Understanding questions! These are there to help you solidify those concepts. **If you can't do one, ask to go over it at the beginning of the in-class portion!**

Instructions for this Lecture!

Watch the following 5 videos in the lecture notes PDF and then return to the Panopto Video to continue!

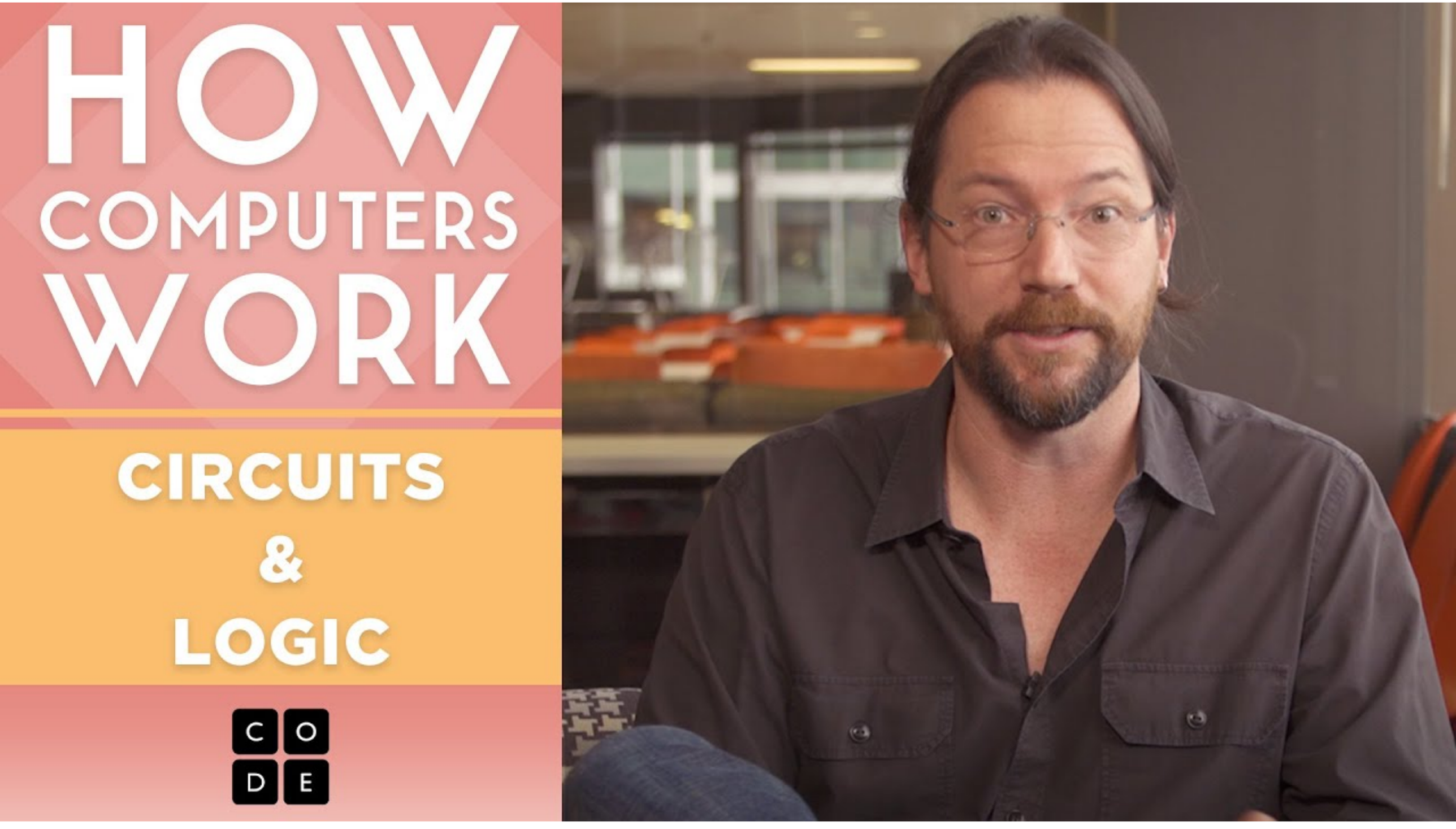
Understanding Computers: What is a computer?

https://code.org/educate/resources/videos

Understanding Computers

https://code.org/educate/resources/videos

Understanding Computers

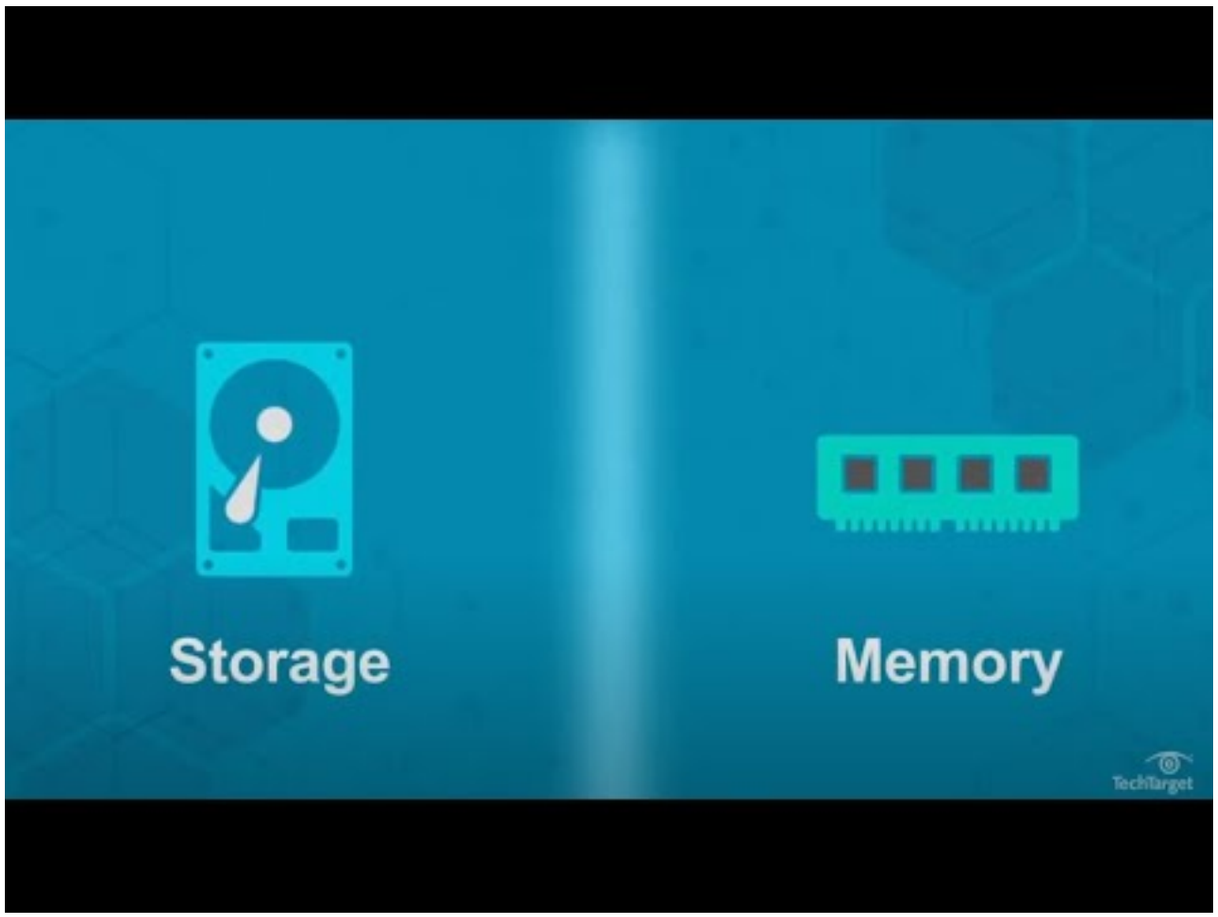


https://code.org/educate/resources/videos

Understanding Computers: Hardware and Software

https://code.org/educate/resources/videos

Understanding Computers: Memory versus Storage



https://www.youtube.com/watch?v=H_M--weEzpA

Check Your Understanding

What are the four things that all computers have in common?

a) input, output, VR, processing

b) input, output, processing, storage

Correct answer

c) hardware, software, circuits, logic

d) CPU, hard drives, memory, buses

Recapping the Videos

- A **computer** is a device that takes input and storage, performs processing, and produces output.

- Computers operate on binary systems (0's and 1's) to create and store information in a variety of forms.

- Computers use basic circuits and logic to perform a variety of calculations in binary.

- No one uses binary to talk to computers because special software converts back and forth between computer binary language and human-readable coding languages (like R), called **compilers**.

- Computers possess both **hardware** (the physical devices and circuits) and **software** (programs, logic) that tells the hardware what to do.

    - Hardware includes the central processing unit, storage, memory, connections, and input and output devices.

    - Software includes the operating system which coordinates all programs running on the computer, as well as CPU, storage, and memory usage.

What does this mean for you?

- When we talk about programming a computer, we have to be mindful of what's going on under the hood to be effective!

- What you should think about:

  - **Inputs and Outputs** – What information are you giving to the computer and what would you like the output to look like?

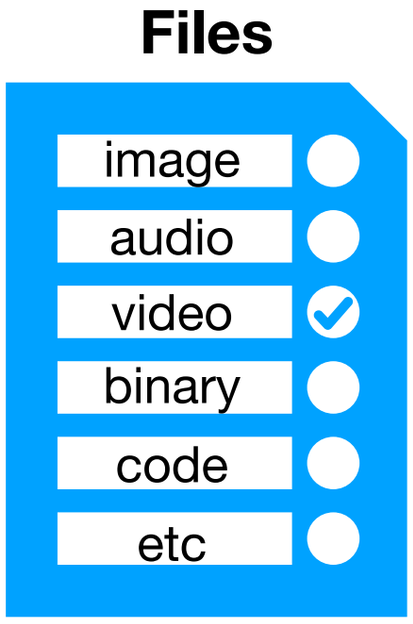
  - **Languages and Circuits** – Ultimately computers only understand TRUE and FALSE binaries, so how can you create instructions that can tell the computer exactly what steps it needs to do to perform the calculations you want?

  - **Storage and Memory** – Be mindful of what is in storage (saved on your hard drive) versus what is in memory! Information only moves from storage to memory when you tell it to!

Filesystems: How computers handle storage

- *Purpose:* Provides a non-volatile storage and organization of information

- *Parts:*
  - Namespace (rules for naming things)
  - Metadata structure (info about files)
  - Application Program Interface (API)
  - Security (Permissions, which users are allowed to access, run, change files)



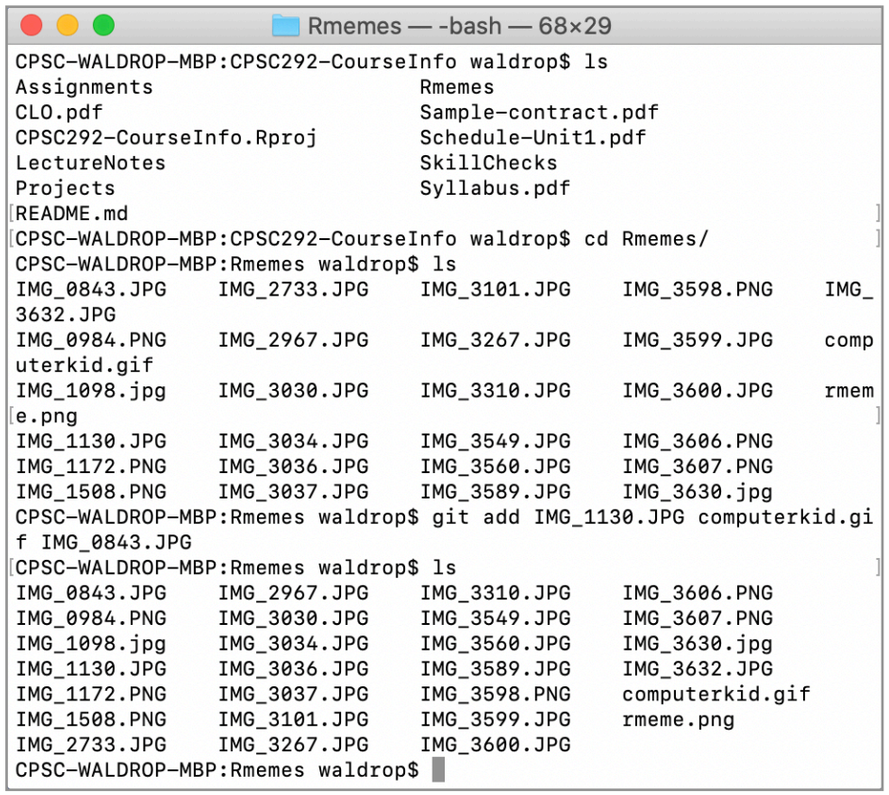
- Operating system (aka the kernel 🌽) interacts with virtual filesystem, which organizes drivers (🚗 software which talks to hardware)

Permissions	Info on files	File names
-rw-r--r--@	1 waldrop staff 11799 Dec 13 2016	Bosque-IBAMR-Install-clang.rtf
-rw-r--r--@	1 waldrop staff 9199 Dec 13 2016	Bosque-IBAMR-Install_Notes.txt
-rw-r--r--@	1 waldrop staff 12445 Dec 21 2016	IBAMR-Bridges.rtf
-rw-r--r--@	1 waldrop staff 10876 Jun 19 2017	IBAMR-Bridges2_gcc.rtf
-rw-r--r--@	1 waldrop staff 16603 Apr 10 16:50	IBAMR-Bridges2_intel.rtf
-rw-r--r--@	1 waldrop staff 14782 Apr 12 17:45	IBAMR-Bridges3.rtf
-rw-r--r--@	1 waldrop staff 10630 Sep 15 2018	IBAMR2016_Dogwood.rtf
-rw-r--r--@	1 waldrop staff 143916 Oct 3 2016	IBAMR_Install_August_2016.pdf

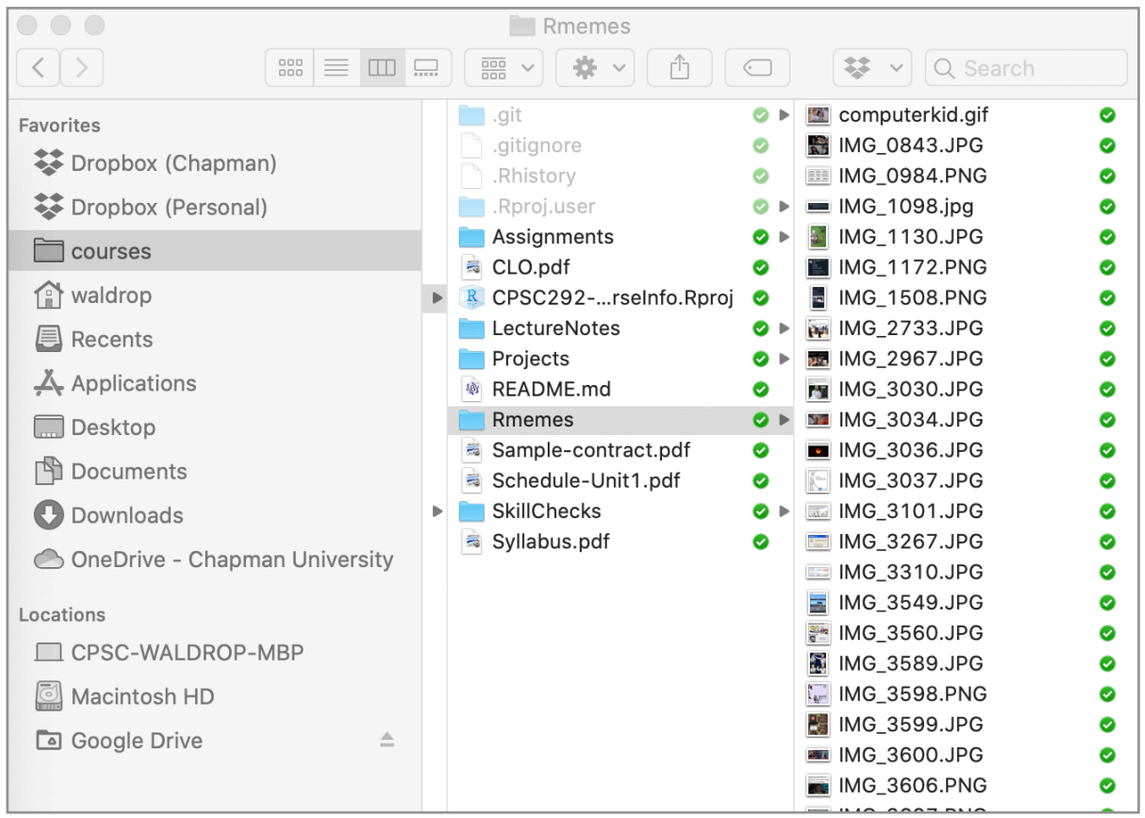
Navigating Filesystems

- **Shells** 🦪**:** special programs help you/the kernel 🌽 navigate and interact with the filesystem.
    - **Command-line shells** – BASh, zsh, and MS-DOS are command-line shells. You navigate by entering specific commands to navigate and interact with items in the filesystem.
    - **Graphical User Interface (GUI) shells** – Finder, Windows Explorer, and Pantheon are GUI shells which you navigate by clicking graphics.

BASh (Linux, Mac OS)

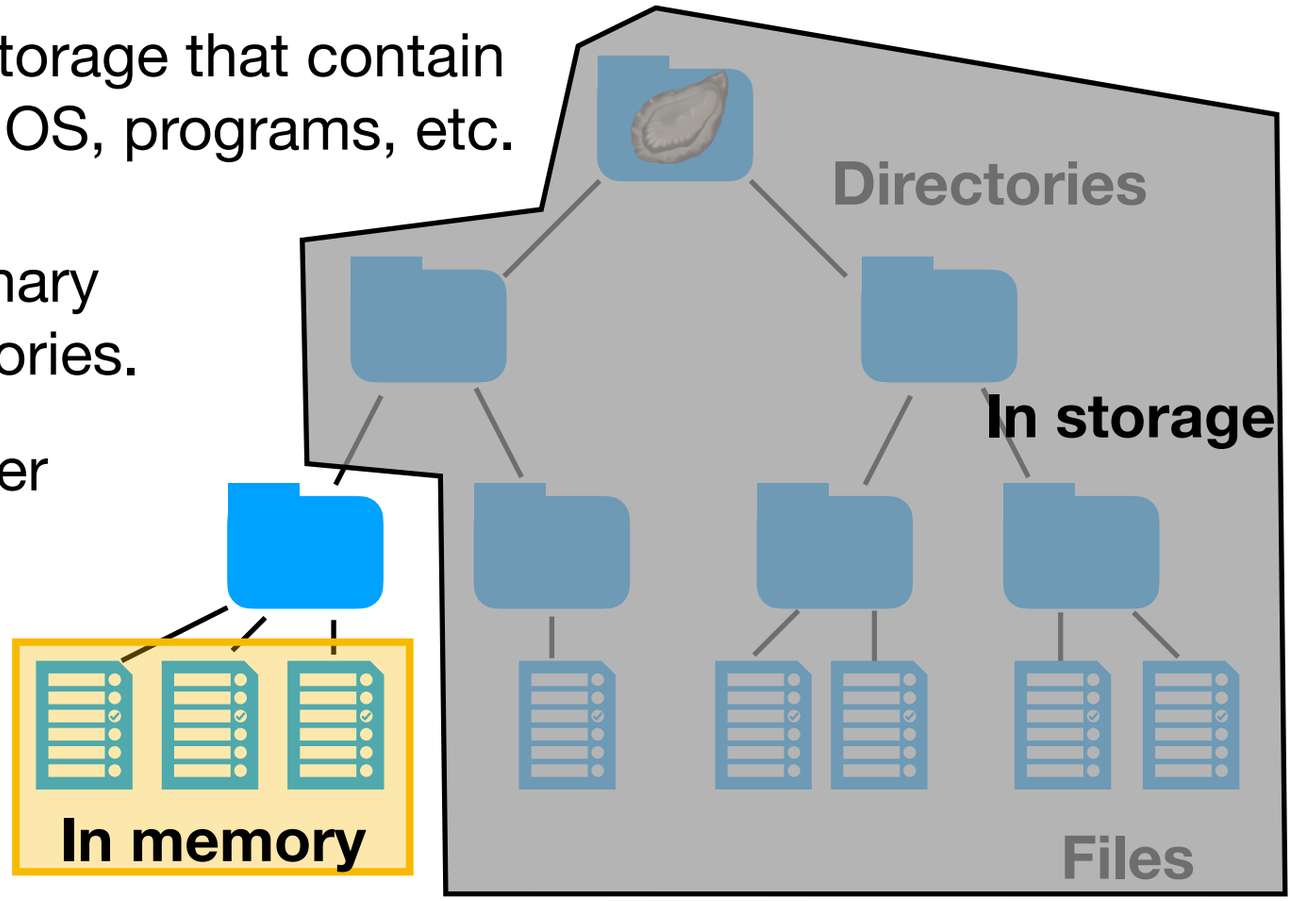


Finder (Mac OS)



Working Directories: how filesystems handle memory!

- **Directories***:* subdivisions of filesystem storage that contain files. Directories are created by the user, OS, programs, etc.
    - Directories can contain items like binary files, computer code, or other directories.
    - Because directories can contain other directories, they have a *hierarchical structure*.

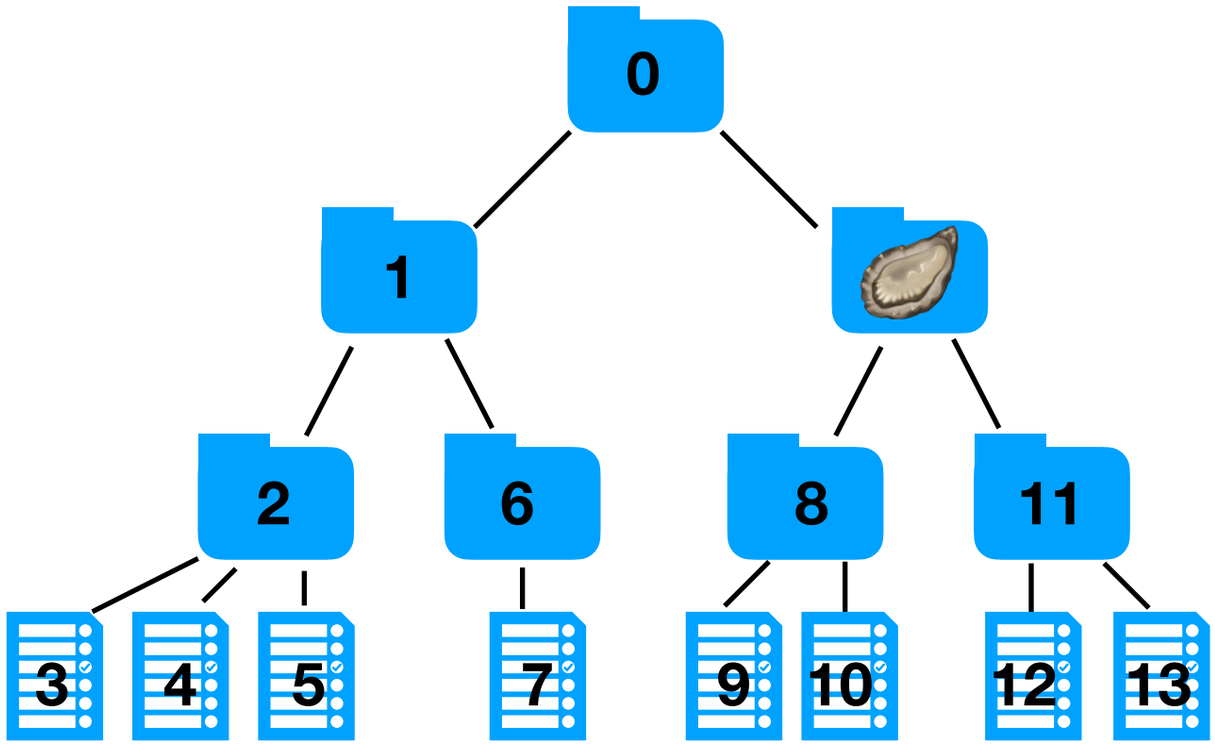


- **Working Directories***:* are the current location of the shell within filesystem storage.
    - When you “enter” a directory, the shell loads into memory the **metadata** about the items within it like the size, date last accessed, permissions, etc.
    - The shell only has information on the items within the working directory! It doesn’t know anything about items in other directories!

Remember where you are (or your working directory)!

Check Your Understanding

If the shell's working directory were in this folder, what items would it see (load metadata for)?



a) 0, 1, 2, 6

c) 8, 11 **Correct answer**

b) 8, 9, 10, 11, 12, 13

d) 0 through 13

Finding the location of an item

- Each item has a location based on its position in the filesystem hierarchy, described by which directories it occupies (in order!)

- **Root directory:** a computer's highest-level directory, this is a `/` in Linux-based OS and `C:\` in Windows.

  Location of file 3: `/1/2/`

  Location of folder 11: `/14/`

- **Home directory:** a specific user's root directory. Shortcut in command-line Bash is `$HOME` or `~`

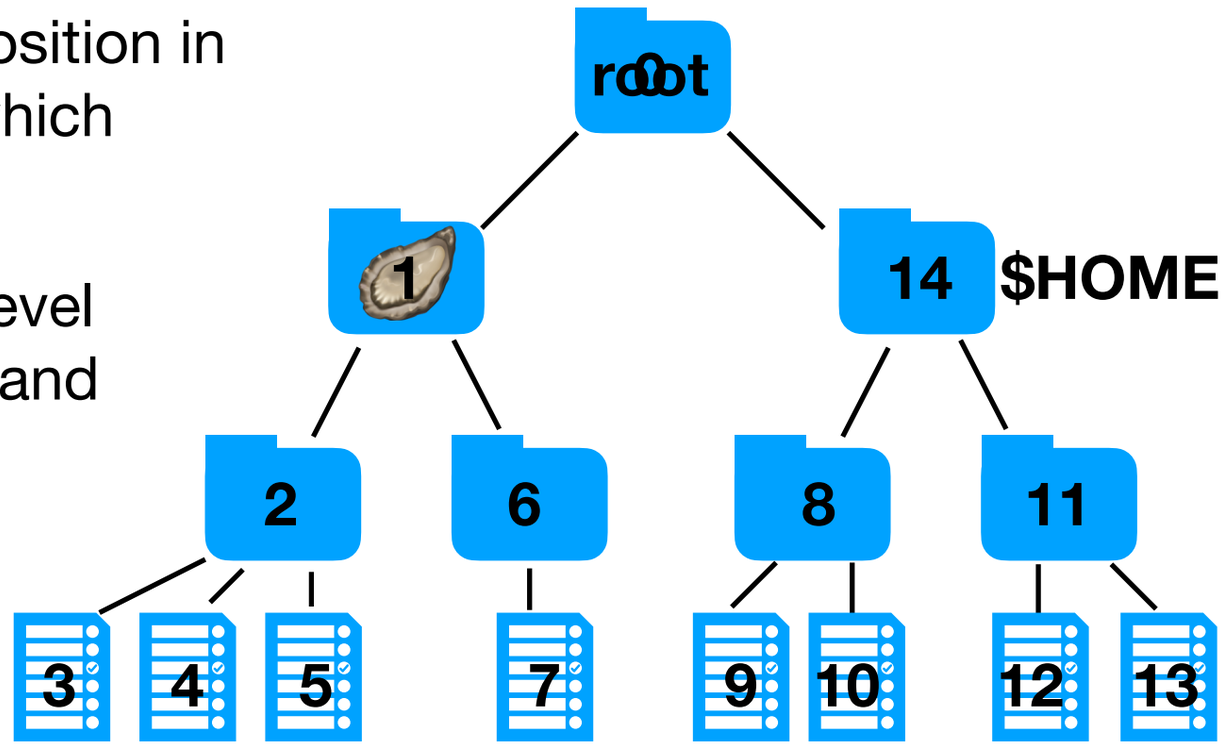
  Location of file 12: `/14/11/` **OR** `$HOME/11/`

- **Working directory:** shortcut is . If you don't include the root or home shortcut, the shell will look in your working directory!

  Location of file 3: `2/`

  Location of file 7: `../6/`

  Two dots indicates the directory one above your current directory!



Check Your Understanding

1) **Write down the location for file 7 from root.**
2) **Write down the location for file 7 if your working directory is 8.**

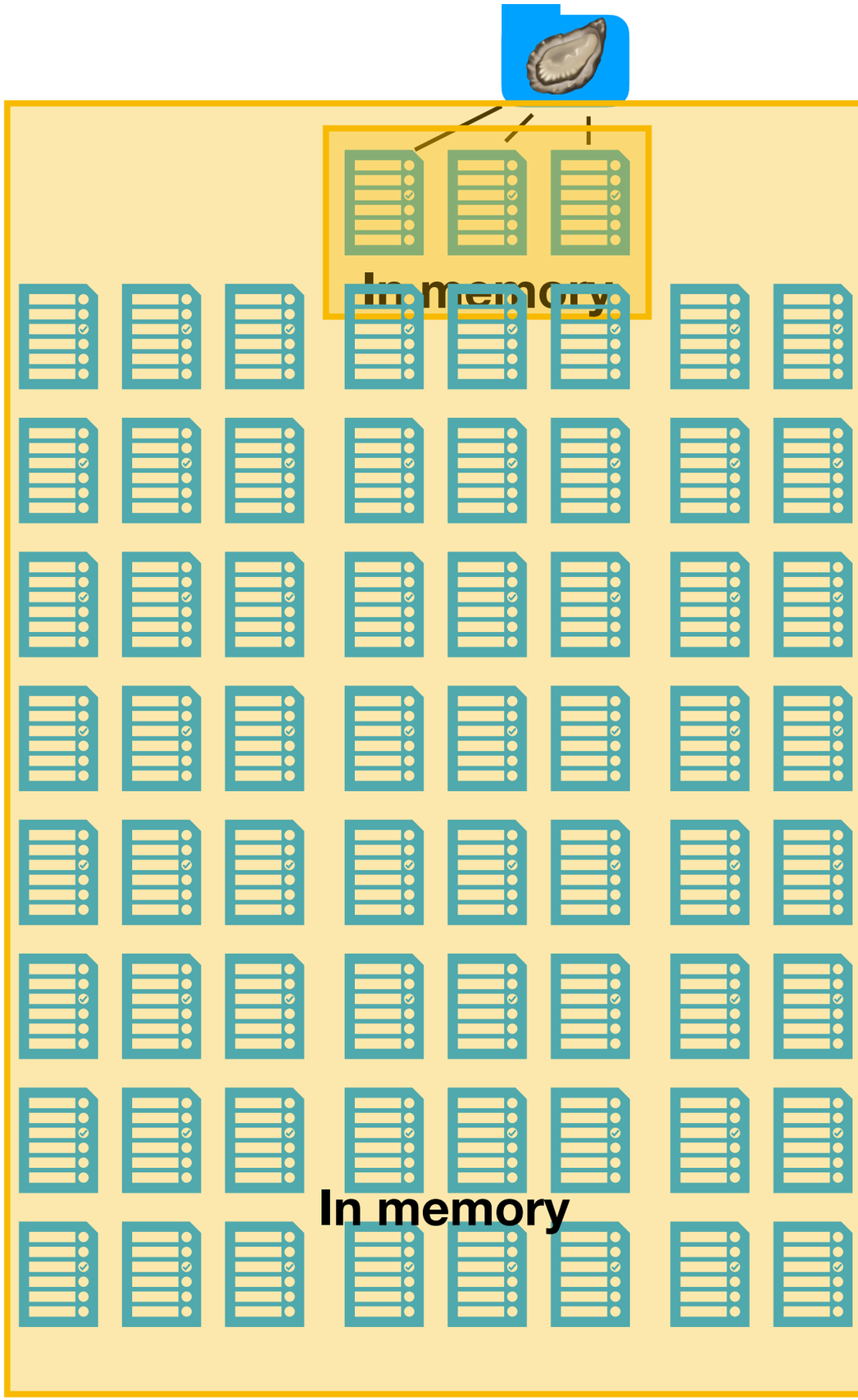
Why bother using directories to organize files?

- Only metadata of files in the working directory is loaded into memory, but what if that directory has a LOT OF FILES??

- By having too many files in one directory/folder, you clutter up your memory with all sorts of stuff you don't need.

- It's also super hard to find stuff, or remember which versions are which, what's related to what, etc.

- **Do yourself a massive favor and organize your files into separate directories!!!!!!!**



DON’T WORK IN YOUR DOWNLOADS FOLDER.

Action Items

1. **Complete assignments 1.2 and 1.3 (due a week from today).**

2. **Review Computer Games for the next class! (No video lecture.)**