Linux tutorial on writing a bash script: <https://docs.google.com/a/college.harvard.edu/document/d/1WTTz6f27SL8nVFojtYCExjpmmEY7IjZg5LJD2RvHoFU/edit?usp=sharing>

**Computers and Their Cores**

* Each task has a thread - a program could have 1+ threads
* One core can only process one thread at a time
  + To give the appearance of simultaneously running more than 1 program, one core processor rapidly switches between threads
* One core vs. dual core vs. quad core
  + The more cores you have, the slower your clock time (gHz) per core because there are thermal restrictions on the entire machine
  + High clock time is higher processing speed
  + If your software is designed to only use one core (and can’t use multiple cores), then the software would run faster on a computer with one core than a computer with 4 cores (quad core processor has smaller clock time)

**How to manage space**

To check what’s taking up space

* About this mac > system information > storage

Using a SEAGATE external drive

* Non-mac version is partitioned with NTFS - thus, by default, MacOS is unable to write on it by default
  + \*in contrast, flashdrives are partitioned with FAT32, which MacOS can read/ write on by default
  + See aside below
* The Lab’s Mac Pro
  + macOS Sierra Version 10.12.4
    - macOS Sierra is a version of Apple Inc.’s **desktop and server operating system**
  + Processor: 3GHz 8-Core Intel Xeon E5
  + Memory: 64 GB

An aside: NTFS vs. FAT32

* Both NTFS and FAT32 are **file systems** - which is a set of logic, structure, properties of speed, security, size which organize a set of files. Without a file system, “**information in a storage medium** would be **one large body of data** with no way to tell where one piece of information stops and the next begins“
* New Technology File System (NTFS)
* File Allocation Table (FAT)
* Windows created NTFS to solve a couple problems with FAT32:
  + Theoretical storage limits are 4GB files and 2TB volumes
  + NTFS has larger theoretical storage limits
  + NTFS has security permissions (e.g. read only, write only, read-write)
    - A file system in NFTS CAN BE accessed/ written on by MacOS but not automatically: [here’s a link](http://www.seagate.com/support/downloads/item/ntfs-driver-for-mac-os-master-dl/) to help you write on a Seagate drive with MacOS
* Drives that are more than 2TB are probably formated with NTFS
* MacOS doesn’t have the support for writing on NTFS file systems - MacOS only has support to read NTFS file systems

What is **cache**?

* A CPU has access to a computer’s memory
  + A computer has a hard drive for memory and a RAM system
  + The RAM system is faster to access than the hard drive
  + But the RAM system still isn’t fast enough for a CPU
  + What is fast enough for a CPU? Another layer of memory!
    - **Hard Drive** << **RAM system** << **Cache** (increasing speed)
  + A cache is memory storage inside the CPU itself
  + A cache is only a couple MB large, it takes information from the main RAM system and the CPU can run on only cache memory alone if the cache “hit rate” is high enough
  + Cache operates on **static RAM (sRAM)** while system RAM operates on **dynamic RAM (dRAM)** that takes more time to refresh

Why is my Mac so slow?

* Not enough “disk space” = memory on hard drive not RAM

What is the difference between **computer memory (RAM)** and **hard drive storage**?

* There’s a difference between **memory** and **storage**!
* Memory describes a temporary place to put files; storage is a permanent place to put files
* When a computer is editing a file inside of hard drive storage,
  + The file is copied into RAM
  + The CPU dynamically edits the copy of the file in RAM
  + **\*if the electricity goes out, RAM files are deleted, hard drive storage is preserved**
  + If you save the file, the RAM copy is copied into the hard drive and replaces the original file
  + Analogy: hard drive storage = filing cabinet, RAM = desk space, a power outage = a fan that affects desk space but not the filing cabinet
* When a computer is running a task that involves a chunk of memory

Thinking about the **memory hierarchy**

* Fundamental concept: the faster the access rate, the more expensive memory per unit of capacity
* Vocabulary: **volatile memory** - the data is erased once the computer loses electricity or the computer is turned off
  + Solid state drive (SSD) = non volatile memory that’s between RAM and a hard drive
  + Hard drive > SSD > RAM > cache > register
* Registers are inside a CPU, a coupe bytes large - the CPU

Remaining questions:

* How do you check how many cores a computer is currently using?
* Why is the monster mac so fast sometimes and so slow other times?

**Modems and Routers**

* Modems connect the internet to your personal network
* Routers regulate your personal network by assigning each device a private IP address and a MAC (media access control) address

6/13/17

**Margo Seltzer, Autonomous Scalable Computing - Parallelizing any code**:

* PIN technology, for 1. replicating hardware without having hardware, 2. Used in predicting what points can be parallelized and creating a probability map of where these predicted points are??, 3. Going through the code once, figuring out what parts of the state were changed in the process of going to the next state (what parts of the code were used/ touched)

6/20/17

The following error pops up when I try to open the terminal on the monster mac when he’s slow/ unresponsive:

Login timed out after 300 seconds

[Process completed]