# Introduction to Computing

Sarah Mancina

IceCube Bootcamp

Madison, WI

May 31, 2019

An important thing to remember in computing is, when in doubt, you can always do a google search for your question. There are a lot of resources online including StackExchange, which usually can answer any of your questions. There are also a lot of online tutorials that can teach you python and other languages. This is a basic guide designed to get you from an absolute basic understanding of computers to opening a Jupyter notebook, which can then prepare you to go through our notebook python tutorials. We are trying to make this bootcamp inclusive for all of the incoming levels of our students, so feel free to skip any of this introduction you already are familiar with.

# 1 Computer Basics

## 1.1 Hardware



Figure 1.4

**Hardware organization of a typical system.** CPU: Central Processing Unit, ALU: Arithmetic/Logic Unit, PC: Program counter, USB: Universal Serial Bus.
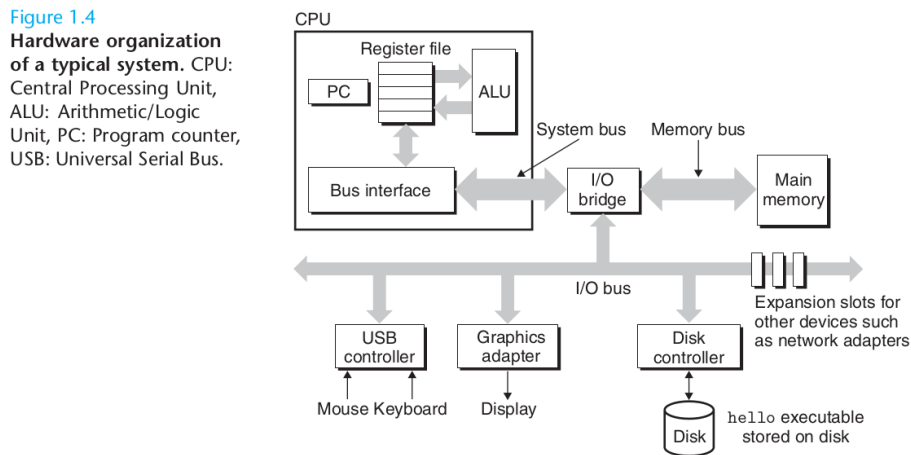
Fig. 1.1. Diagram of computer hardware

If this is your first time coding, it is helpful to know some basics about how a computer works. There are a couple of parts of computer hardware you want to think about when writing basic code: the CPU, the memory, and the disk (figure 1.1).

The CPU is the central processing unit, this is where computations are done under the instruction of a computer program. Most modern CPU have multiple cores, where each core can focus on one task at a time. CPUs have a limited speed that is limited by heat build-up and transmission delays over wires connecting the transistors in the CPU chip.

The memory is storage for quick access by a program. A common type of memory is random-access memory (RAM). When writing a code, it's important to consider the amount of memory your program will use, since it can be easy to use too much memory.

The disk is the computer's storage. Two common types of internal disk are hard drives and solid state drives (SSD). Standard laptop disk drives store about 200GB to 2TB of data. You want to make sure your code is not set up to write more bytes of code to your disk than is available. Also, disk input and output (I/O) is relatively slow compared to accessing memory. Your operating system is usually stored in a partition on your main disk, but you can have multiple internal disks and can plug in external disks (e.g. external hard drives, flash drives, CDs, floppy disks (that save button thing), etc.).
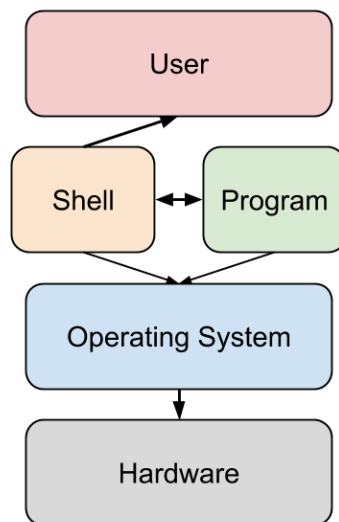
## 1.2   Software



Fig. 1.2. Mapping of computer software

An operating system (OS) creates the first part of a map between the hardware and the user (figure 1.2). The three most common types of OS are Mac, Windows, and Linux. Mac and Linux are in the UNIX family of OSs. Mac and Windows are proprietary software whereas Linux systems are open source. This makes it easy for IceCube software and security to be quickly updated for changes to the Linux OS. A user friendly Linux OS is Ubuntu. The virtual machine sent out to all bootcamp attendies is Ubuntu 18.04.2 which is the newest long term service release.

The user interacts with the operating system and programs through what is called a shell. There are two types of shells, command line interfaces (CLI) and graphical user interfaces (GUI). Windows machines use the Windows GUI, Mac machines use the Aqua GUI, and Ubuntu machines use the Unity GUI as a default.

Terminal is a program which is a terminal emulator which uses a CLI shell that interfaces with the GUI shell (e.g. Unity, Aqua). Mac and Linux machines come with terminal which uses Bash, a Unix CLI shell, as a default. Most Icecubers use the terminal command line to write code, run programs, and connect with the WIPAC Cobalt machine nodes over a secure shell (ssh).

# 2    Command Line Basics

When you open up a terminal, you'll start in your home directory. A directory is a folder which can contain computer files and more directories.

Some useful commands you can run are:

- `ls [directory]` - lists the contents of a directory

- `pwd` - print working directory

- `cd [directory]` - change working directory

- `mkdir [name]` - make a new directory

- `rm [filename]` - remove a file

- `mv [filename] [path][filename]` - move a file (includes renaming a file)

- `cp [filename] [path][filename]` - copy a file

- `man [command]` - open up a manual for a command

- `ssh [username]@[machine]` - create a secure shell conection to a machine

Most of these commands also have addition options which can be view using `[command]` `-h`, `[command] --help`, or `man [command]`. For example, if you wanted to list a directory to view information such as the size of the files, you could run

```
> ls -lh ./
```

which adds the `-l` option, which means use a long output format, and `-h` option, which means puts the output in a human readable format. The `./` tells the `ls` program to list the contents of the current working directory, but it is not necissary to include since `ls` will assume the current directory if the input is left blank. More commands line commands and tips are covered in the first python tutorial.

When writing a code you can use a text editor. Nano, Vim, and Emacs are common text editors which can be run using the command line. Nano is good for beginners, but Emacs and Vim have more features and are more commonly used but have a learning curve. However, to learn python we will be using Jupyter Notebook which will open up a browser window where you can create notebooks which interactively run computer code.
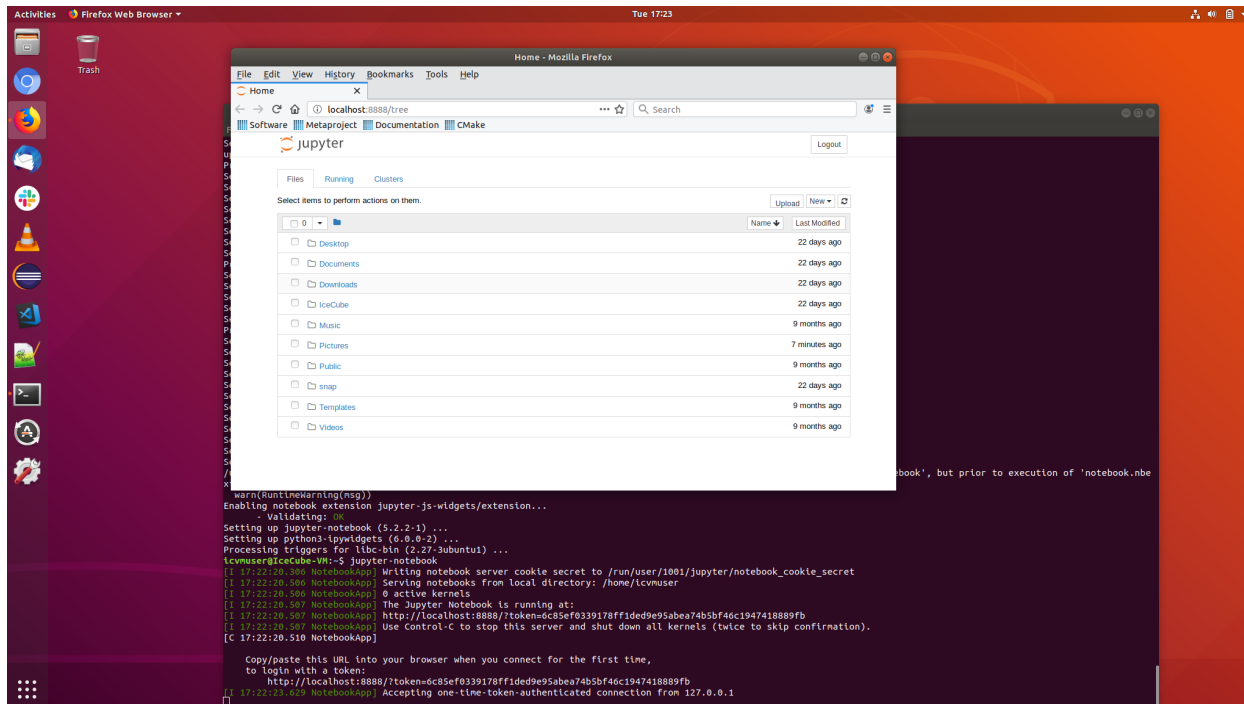
3

Fig. 3.1. Jupyter notebook home browser page

# 3    Installing and Running Jupyter Notebook

To install Jupyter notebook open terminal on your Ubuntu virtual machine. Use the following command to install the

```
> sudo apt install jupyter-notebook
```

The sudo command will require you to enter the password for the user. After the password is entered type Y when prompted to confirm you want to install jupyter-notebook on your machine. Next, you might want to navigate to your IceCube directory before you open up Jupyter notebook.

```
> cd IceCube/
```

Then open Jupyter notebook using the following command.

```
> jupyter-notebook
```

This should cause a browser window to open up as shown in figure 3.1. You can use this browser page to navigate the directory and create new notebooks. Woohoo!