1 Introduction

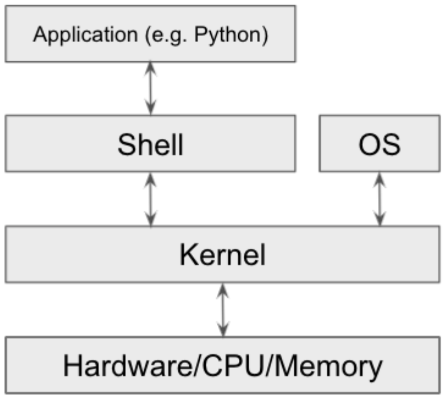
Goals for the ULAB cs curriculum:

1. Familiarity with the terminal/command-line
2. Basics of the Python Language
3. Python packages: numpy and matplotlib

BUT, we only have a few lectures. Lectures + HWs will not be enough to make you proficient.

You need to PRACTICE. One way to practice is by conducting your projects!

2 The Computing Stack

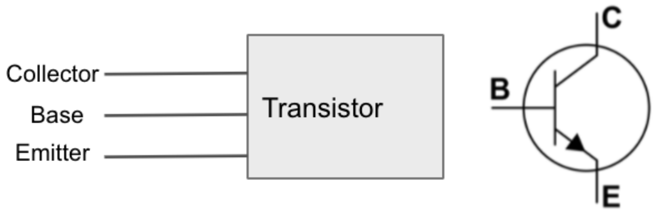


Today's lecture is dedicated to giving you an overview of how a computer works. We will cover a lot to terms and concepts that often pop up when dealing with programming. This will also give you context on where Python is in the mess of things.

It is not necessary for you to understand the technical intricacies of everything that we'll cover. Don't worry if everything doesn't sink in.

3 Hardware

1: Transistor a semiconductor device that acts as a switch.



Base HIGH: current passes from collector to emitter

Base LOW: current can not pass for collector to emitter

2: Logic gate collections of transistors that can compute basic logical operations.

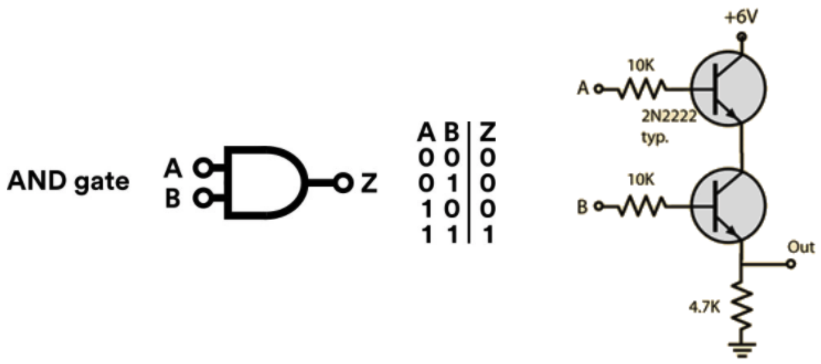


Figure 1: Image source: http://hyperphysics.phy-astr.gsu.edu/

3: CPU (central processing unit) a collection of logic gate that can perform more complex operations: arithmetic, logic, controlling, input/output (i.e. reading from memory)

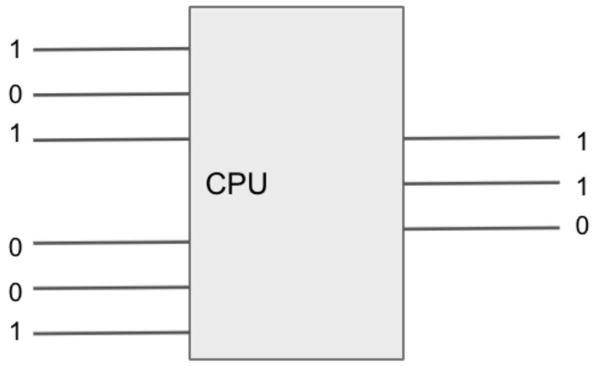


Figure 2: Simple diagram of adding $1 + 5 = 6$

Machine code: a low level programming language, written in binary, that a computer can directly execute.

Q: Are we done?

No. While we have a computer, but how do we program it to do what we want?

Past: physical punch cards completed circuits to program inputs. PAINFUL!

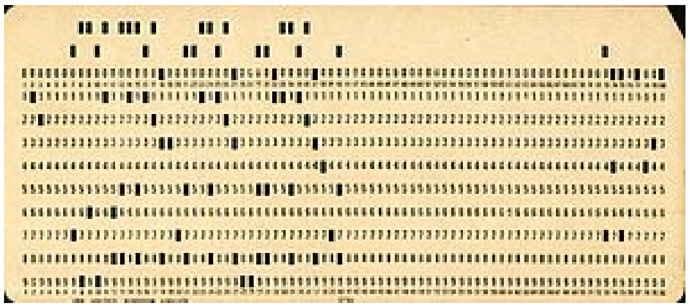


Figure 3: Image source: Flickr, 'Used Punchcard'

Now: we have keyboards, monitors, and operating systems. But there's a bit more to the story...

4 Kernel

- The core program that interfaces between the hardware and software (i.e. applications and programs we want to write).
- On startup, the kernel is the first program to start (...ignoring BIOS/UEFI and bootloader and I'm not a computer engineer...)
- The kernel is always running and has complete control over the computer.
- Kernel is the core of the operating system

5 Operating System

- The OS is composed of a kernel and additional software to allow us to more easily interact with the computer. E.g. graphical user interface (GUI).
- Common OSs as Windows, Mac, and Linux. Each have a different kernel. A kernal/OS is not restricted to a single hardware platform. A PC can run both Windows and Linux.
- Not practical for remote access.

6 Shell

- A program that allows the user to directly interact with the kernel
- The shell program has a text-based interface often called the terminal or command line
- Greater control over OS
- Shell scripting (automate/program shell commands)

On Windows the default shell is DOS (Command Prompt application). On Mac and Linux machines, Bash is the default shell (Terminal application).

Students should download Bash (Git Bash on Windows) before lecture.

Demo: bash commands (mention "directory"="folder")

- pwd
- ls
- cd
- moving files to a folder based on file type

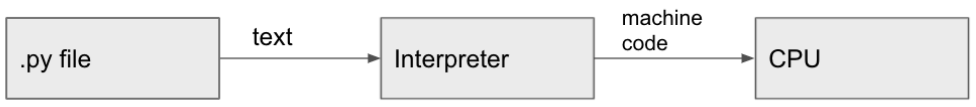
7 Python

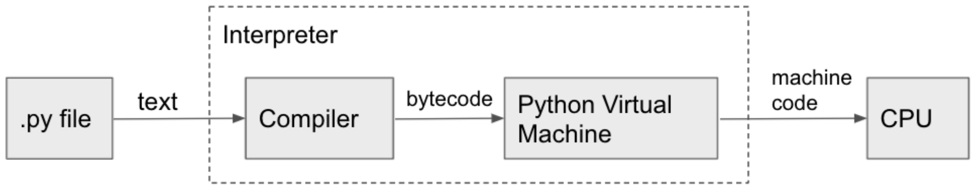
Python is a high-level programming language with human-readable syntax.

- Popular in research (many libraries)
- Fast code development (e.g. dynamic typing)

What happens when you run a Python program? (Interpreter)

- The interpreter *interprets* and executes the python code
- You can think of the interpreter as translating your Python code into machine code for the computer to run.
- When you download Python, you're downloading the interpreter.





- Simple picture: interpreter converts text (Python code) to machine code.
- More complicated picture: the interpreter is composed of a compiler to bytecode and a PVM (C program) that actually runs the Python instructions. BENEFIT: platform independent. Machine code for one processor will not necessarily run on another. As long as we a .py or .pyc file and an appropriate PVM, we can run on any processor![1]

[1]But didn't we just delay the problem of compiling a Python file into the problem of compiling a C file? What language do we use to compile a C file? With C! This is known as compiler bootstrapping!

Text Editors and IDEs:

- **Demo**: writing a Python program in a basic text editor (e.g. notepad) and running on terminal.
- While you can write Python code on any text editor (try using Word and keeping your sanity at the same time...) there are special programs called integrated development environments that act as fancy text editors to write, run, test, and debug code.
- Example of IDE: PyCharm, Atom, Jupyter Notebook etc. **We will be using Jupyter Notebook: a very popular IDE in the lab!**

8 Next Week

- Discussing how to use Jupyter Notebook
- Start learning about the Python language

9 Homework

- Practice Bash commands
- Download Anaconda (for Python and Jupyter Notebook)