

Software

- Software - code that runs on the hardware
- I'm going to simplify things a bit here
- CPU implements "machine code" instructions
 - Each machine code instruction is extremely simple
 - e.g. add 2 numbers
 - e.g. compare 2 numbers
- Code we have seen `pixel.setRed(10)`
 - To run, expanded to about 10 machine code instructions

Machine Code

"Software" is the general category of code which runs on the hardware. If the hardware is a player piano, then the software is the music. The common case is a "program" like Firefox -- software you run on your computer to solve a particular problem. A computer can run multiple programs at the same time, keeping their use of memory, drawing in windows etc. separated so they hopefully do not interfere with each other.

A CPU understands a low level "machine code" language (also known as "native code"). The language of the machine code is hardwired into the design of the CPU hardware; it is not something that can be changed at will. Each family of compatible CPUs (e.g. the very popular Intel x86 family) has its own, idiosyncratic machine code which is not compatible with the machine code of other CPU families.

Instructions and Programs

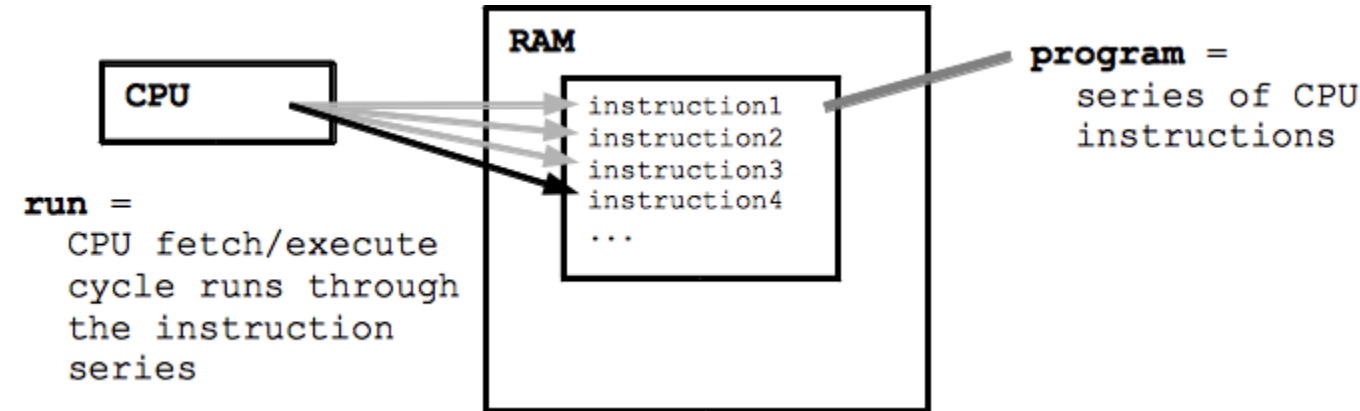
The machine code defines a set of individual instructions. Each machine code instruction is extremely primitive, such as adding two numbers or testing if a number is equal to zero. When stored, each instruction takes up just a few bytes. When we said earlier that a CPU can execute 2 billion operations per second, "operations" there refers to these simple machine code instructions.

A program, such as Firefox, is made up of a sequence of millions of these very simple machine code instructions. It's a little hard to believe that something as rich and complicated as Firefox can be built up out of instructions that just add or compare two numbers, but that is how it works. A sand sculpture can be rich and complicated when viewed from a distance, even though the individual grains of sand are extremely simple.

What is a Program?, What is Running?

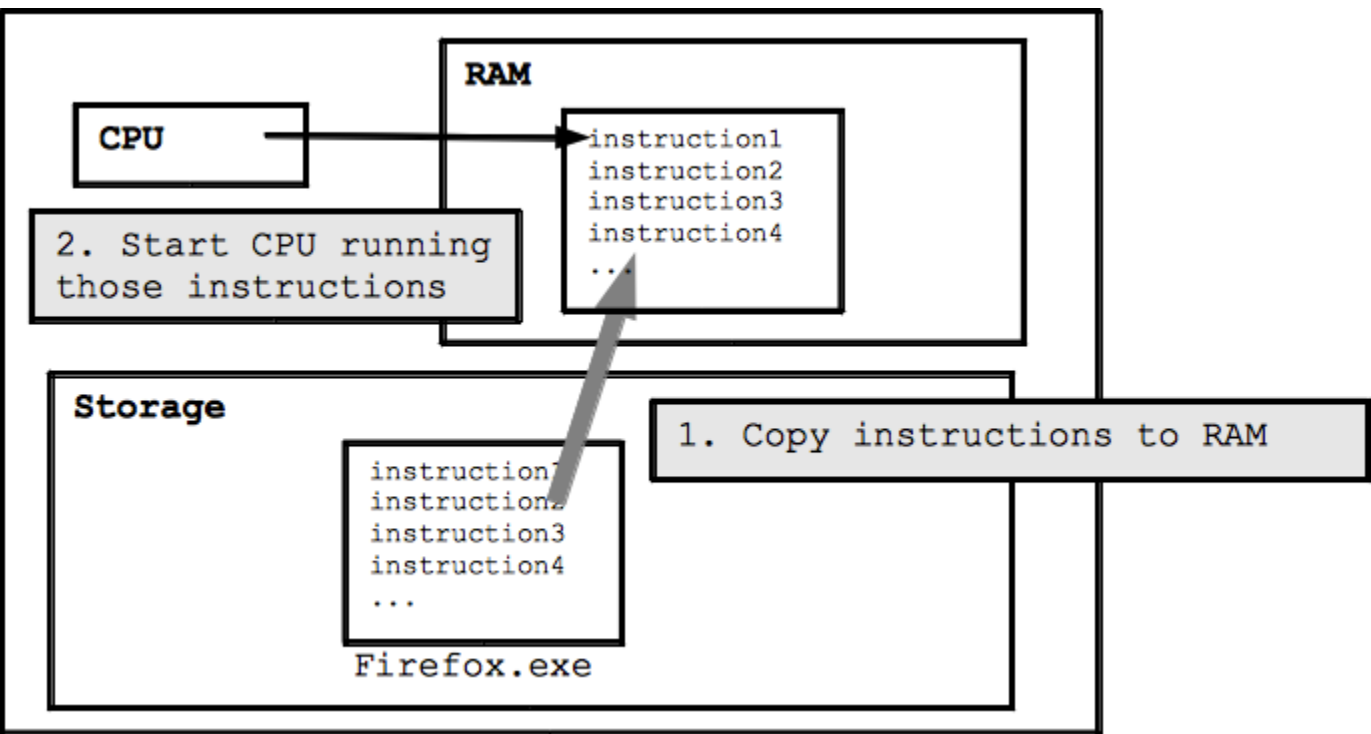
- "Program" e.g. Firefox, millions of simple machine code instructions
 - Instructions, like grains of sand making up sculpture
- CPU runs a "fetch/execute cycle"
 - fetch one instruction in sequence
 - execute (run) that instruction, e.g. do the addition
 - fetch the next instruction, and so on
- "loop" instruction: jump back 10 instructions
 - Loops are implemented this way
- "if" instruction: skip ahead if a certain condition is true
 - If statements are implemented this way

The CPU runs instructions using a "fetch-execute" cycle: the CPU gets the first instruction in the sequence, executes it (adding two numbers or whatever), then fetches the next instruction and executes it, and so on. Some of the instructions affect the order that the CPU takes through the instruction sequence .. for example an instruction might direct the CPU to jump back to an earlier point in the instruction sequence (loops are implemented this way), or to skip over the next instruction if a particular condition is true (if-statements are implemented this way).



Double-Click .. Running?

- What is a program, like Firefox.exe (.exe is a Windows convention)
- The file Firefox.exe is mostly the bytes of millions of instructions
- Double click Firefox.exe to Run
 - The instruction bytes are copied up into RAM
 - The CPU is directed to start running at the first instruction



In the file system, a file like Firefox.exe just contains the bytes of the machine code instructions that make up the program (".exe" is a windows convention to mark a file as a program). Each machine code instruction takes up about 4 bytes, and whole program is just an enormous sequence of instructions.

On my machine, Firefox is 80 MB in size. Assuming all those bytes are instructions, and each instruction is 4 bytes, how many machine code instructions make up Firefox? (Answer: 80 MB is 80 million bytes, so that would be 20 million instructions.)

Double Click to Run

When the user double clicks a program file to run it, essentially the block of bytes of the instructions for the program are copied into RAM, and then the CPU is directed to begin running at the first instruction in that area of RAM.

Operating System

- Who starts Firefox?
- Operating System
- Set of supervisory programs, run when computer first starts
- Administration behind the scenes
- Starting/managing/ending other programs
 - Modern computers can run multiple programs at the same time
 - Operating system keeps each program run isolated
 - Program has its own RAM, its own windows on screen
 - vs. accidental or malicious action between programs
- e.g. Laptop
- e.g. Digital camera

The "operating system" of a computer is like a first, supervisory program that begins running when the computer first starts up ("boots up"). The operating system plays an invisible administrative and bookkeeping role behind the scenes. When a desktop or laptop starts up, the operating system typically gets things organized and then launches a "file explorer" program which displays windows and menus etc. that show the user what file systems are available, allowing the user to navigate and operate on the files.

The operating system keeps things organized in the background so that multiple programs can run at the same time, which is known as "multitasking". The operating system gives each program its own area of memory, so each program only accesses its own resources .. attempting to limit what an erroneous or malicious program can do. Keeping the programs separate is sometimes known as "sandboxing" .. mediating the access of each program so it operates independently, without interfering with other programs or the system as a whole. Similarly, each program has some access to the screen through a window, but this output area is separated from the output of other programs.

Recall that a .exe file or whatever is essentially just a file of machine code instructions. When you double-click the program, it is the operating system that "launches" the program, doing the housekeeping steps of allocating an area of memory within RAM for the program, loading the first section of the program's machine code into that memory, and finally directing the CPU to start running that code.

A digital camera is also a little computer. When it starts up, it does not run a file manager program. Instead, after the basic housekeeping is set up, the camera may just run a single program that draws the menus etc. on the camera's screen and responds to clicks on the camera's buttons and so on.

Boot / Reboot

- Chicken and egg problem.. who runs the operating system?
- When first powered on, computer runs a tiny "get started" program
- That program typically looks for a disk containing an operating system to run
- Etymology: "lift self over a fence by pulling on your bootstraps"
- Boot up -- start
- Reboot -- shutdown/start-fresh cycle