

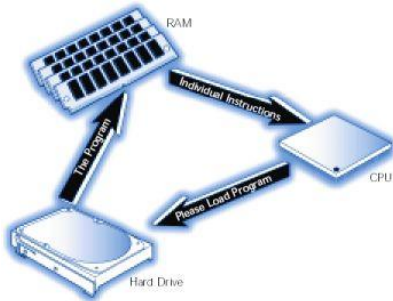
Module 4 – Starting up your Computer and the Operation Systems.

'Start me up'

Lyrics by the Rolling Stones

<http://www.youtube.com/watch?v=ZzlgJ-SfKYE>

As a programmer, and computer user, it is useful to understand the sequence events that are *started up* when you press your Power Button.



For the computer to do something, it needs instructions. Even to *start up*. When the computer is first turned on **the RAM is empty**. There is nothing in it, no start up instructions, nothing.

If you want to use the OS or Application programs, they first must to be *copied* from a HDD or a SSD or a CD into the RAM. When the program is copied into the memory, the CPU can now read the individual lines of code and run them.

Key Concept: The computer's CPU cannot run/execute any programs instructions directly off the computer's HDD or SSD or flash drive, or CD. A program must first be copied from the HDD, SDD, Flash Drive or CD into temporary storage (RAM). The CPU can only read instructions that are in temporary Memory.

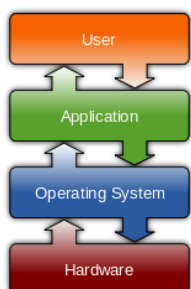
BIOS – Basic Input Output System

Ok... Where is the program stored that instructs the computer what steps to take to start ? The **BIOS**.

What is the BIOS chip ? It is a solid state memory chip mounted on the motherboard.



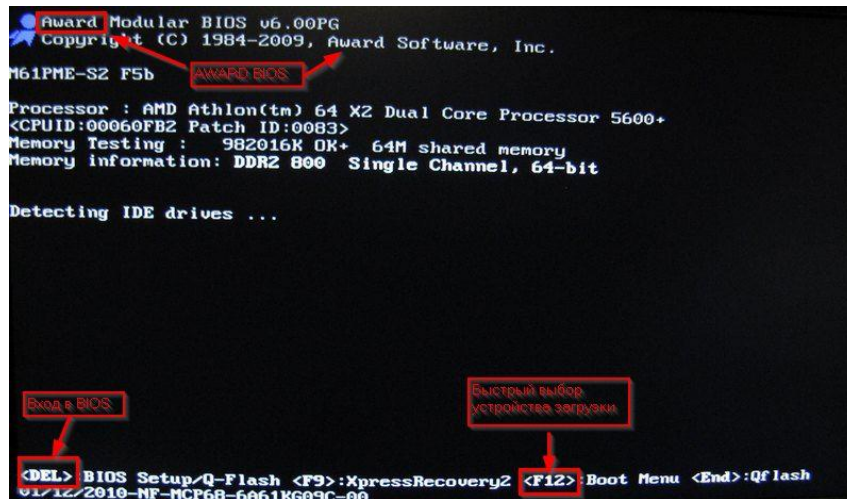
The basic instructions in the BIOS are to check the power supply, memory and other hardware, then load the Operating System (OS) Software into memory (RAM).



The *Operating System* is the program that manages the hardware: *CPU, MEMORY and I/O Devices*, and more. Those OS functions are separated out into their own program and are not part of APPLICATION programs.

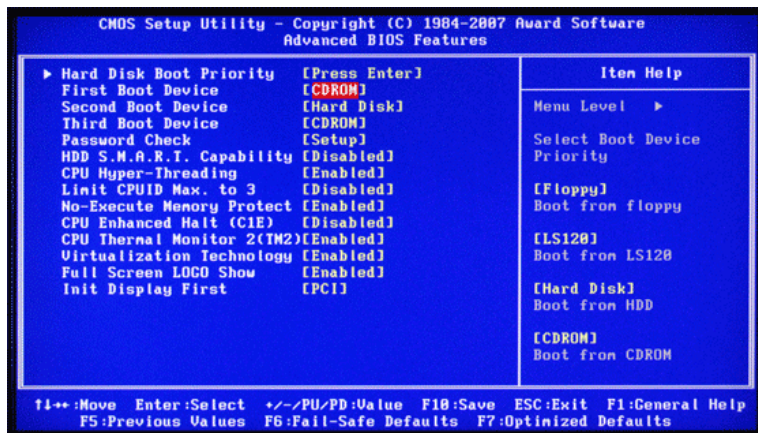
Making a separate program (OS) to act as a software layer about the Hardware, enables Application programmers to use that prewritten code layer of OS code, and concentration on the unique things they want their Application programs to do. (Hierarchy of concepts)

The steps – for Starting up or Booting your Computer



Note: This information may be displayed differently depending on your version of BIOS.
Your BIOS may be set to NOT display this startup information.

- Press the Power Button
- **Power Supply Test**
 - Power supply performs Self test. You need to insure that there is reliable power available for the rest of the computer, before you do anything else.
 - Power Good signal sent to CPU
- CPU loads the address of BIOS boot program into RAM.
- **POST Test.** CPU runs BIOS **P**ower **O**n **S**elf **T**est (POST) code -
 - CPU runs BIOS tests of basic hardware: Beep codes if errors
 - CHECK CPU
 - CHECK BIOS integrity
 - Video Card Test – Runs Video cards own BIOS, test and initializes video card.
 - **Memory Test:** Hex Memory address displayed on screen if memory error
 - BIOS start up Screen Displayed Display BIOS info on Monitor (Depends on Settings)
 - POST and Beep codes: http://en.wikipedia.org/wiki/Power-on_self-test
- **Locates Disk with OS.**
 - BIOS reads CMOS memory (maintained by small *battery* on motherboard).
 - **Locates sequence in which to check list of drives for location of Operation System code**
 - Checks on BIOS list, starting with first drive, and identifies which sequence of disks to check for the OS



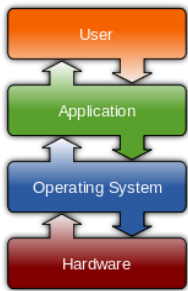
Note: If you go into your BIOS at startup, you can select the search order for the Boot Drive

- BIOS examines disk with OS for Master Boot Record code (MBR).
 - Loads MBR code into RAM.
(MBR consists of Partition Table data which describes layout of disk and instructions how to load it into RAM).
 - BIOS Transfers control to the Partition loader code
 - Detects other Hardware
 - Notes Hardware Configuration
- **Loads OS File System Drivers** so it can load KERNAL OS files.
- Loads the OS into RAM and starts the subprograms:
 - OS Object Manager
 - OS Executive
 - Security Manager
 - OS Memory manager
 - OS Cache Manager
 - I/O manager
 - OS Process manager
- Loads OS System **Device drivers**.
- Launches the Session Manager (SMSS)
 - **Start up the GUI** (Images next page of Windows and Mac PC Login Pages)
 - User LOGIN



- Successful Startup.

Parts of the Operation System



There are many different Operation Systems such as:

Max OS, Android, Windows, Linux and Others

An Operation System consists of several components called **Managers**.

They all have the same basic components: **Process, I/O, File, GUI, Memory, Cache, Security** and others.

OS User Interface Manager - New and Old

GIU. Today, the newer and more commonly used interface is a **GUI**, a *Graphical User Interface*. It is a rich visual environment where you see icon objects and manipulate them. You have been using GUI Interfaces for years. You are familiar with its folders, Files and Icons, look and feel.



Window Gui



Mac GUI

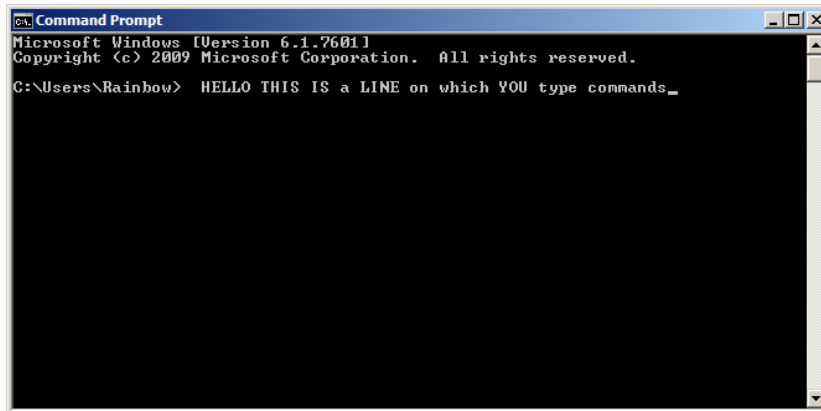


Linux GUI

CUI. Before GUI's, in the old days, there was **CUI**, Character User Interfaces. It is also called **CLI**, a Command Line Interfaces. In a CLI, you had to **type** all OS commands and had a lot of commands to remember correctly. Most Users found this a bit complicated, and were glad when the GUI OS finally became available so they did not have to use the CLI.

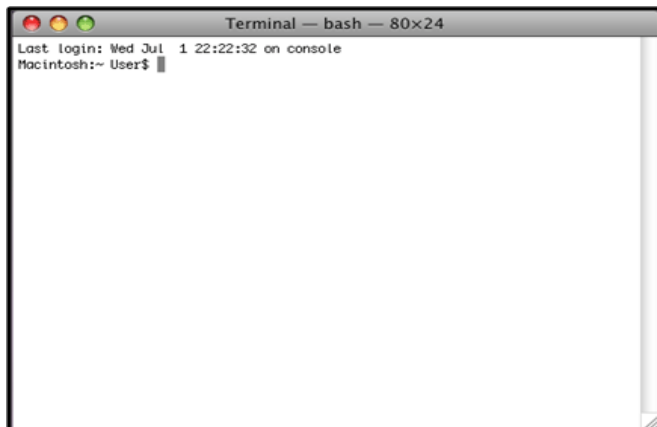
As a programmer, you will find you need to sometimes understand and use a Command Line Interface. You can start up a COMMAND line is a separate window on your computer. Each CLI OS has similar commands. If your take more programming classes, you will learn more about each OS's CLI commands.

Windows OS CLI window



How to use Windows CLI commands: <http://www.youtube.com/watch?v=MNwErTxfkUA>

Mac OS CLI window



How to use MAC CLI commands: http://www.youtube.com/watch?v=ftJoIN_OADc

Linux OS CLI window



How to use Linux CLI commands: <http://www.youtube.com/watch?v=kUEIRcvOiwo>

OS Process Manager

There needs to be some OS code that manages the programs that are currently running on the CPU. Note the word ‘process’ management is related to the Central ‘processing’ unit. The OS process/task manager is a key part of the OS.

Normally the OS AUTOMATICALLY handles the management of processes/programs that are running. If you want to watch what processes/programs your OS is running on a CPU, you can look at a 'Task Manager' Tool. If you want to MANUALLY modify how tasks are run, you can. Watch the following Videos:

Window Task Manager: http://www.youtube.com/watch?v=xaXON_v9jIM

MAC Task Manager: <http://www.youtube.com/watch?v=zXGqCPjsvoM>

Linux Task Manager: <http://www.youtube.com/watch?v=YVDLjnbXrQg>

As a programmer or as a user, and if you have a multi core CPU, you can set which program/job will run on which CPU or you can let the process manager decide. Assigning which program runs on which CPU is called setting the CPU Affinity:

Windows CPU Core Affinity: <http://www.youtube.com/watch?v=bKCcdIfrocQ>

As a programmer or a user, you can assign the priority for a process, who get more CPU use.

Window CPU Program Process Priority:

<http://www.youtube.com/watch?v=wj5NCXTpqgM>

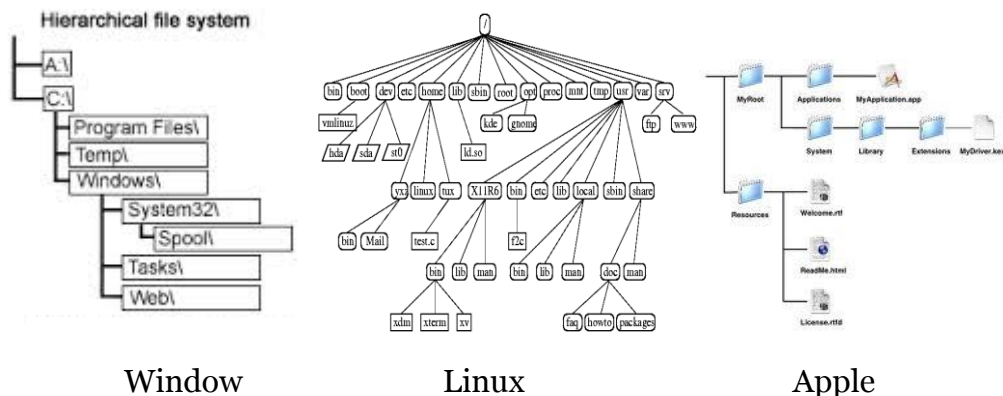
Through your OS, you can schedule the time programs will run.

Task Scheduler program: <http://www.youtube.com/watch?v=8986eNlXjos>

Using your OS, you can decide the priority your program runs on.... Great for playing games... crank up the priority.

OS File Management

An OS File Manager is the program that is used to *automatically* keep track of the folders and what files are in which folders. It lets you create new files, save files, create and delete folders, move files, delete file and more. Folders are arranged in a **HIERARCHY**. It is entirely up to us to think clearly and create a hierarchy to store your data files, so your work is easy to place and retrieve.

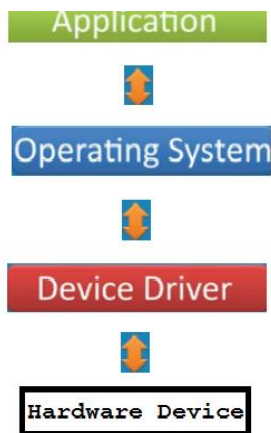


Windows File Manager: http://www.youtube.com/watch?v=8_rJ7NJZNhY&list=PL85DB87B9F604D867

Programmer Alert. When you write programs, you may be saving data to a file, and opening a file to read data. You will be interfacing with the OS File Manager.

Where exactly does a files' data (1's and 0's) reside on a disk or in a flash drive is not important. These details are taken care of automatically by the File Manager, and the device. When you write a program that issue commands to open and close a file, or read or write data, these commands will be interfacing with the OS file manager.

OS Device Managers



The Peripheral hardware devices *each* have their own **Device Driver**. The commands and data codes for each hardware device will be translated back and forth between a hardware device and the OS. The OS communicates with the Applications you use or write.

Device Manager: <http://www.youtube.com/watch?v=l24FNbOEEac>

As a programmer, you do not have to worry about how to communicate with the hardware directly, so your programming is much more simpler.

Normally, you leave device drivers on your computer alone..

OS Security/Login Manager

Operating systems provide password protection to keep unauthorized users out of the system. Operating systems can maintain user accounts, *activity logs*, file and folder security. They also may provide backup and recovery routines for starting over in the event of a system failure.

There are other features of Operating System. If you are interested (optional), here are suggested websites.

About Operating Systems in general:

http://en.wikipedia.org/wiki/Operating_system

Comparison of Operating Systems:

http://en.wikipedia.org/wiki/Comparison_of_operating_systems

Why Operating Systems

In the early days of computers, programmers wrote their own OS. It was the *Tower of Babbble*. It ended up with a tremendous number of programs communicating directly with the computer's hardware. It was quickly recognized that there should be a separate level of programming called the Operating System, and *only it* would communicate directly with the computer's hardware.

KEY IDEA: Having *one* Operation System that can be reused/shared by *any* application *written to run on that OS*, has made programming Applications quicker and easier. Now Application programmers no longer programmed OS functionality, just the Application functionality.

32 or 64 bit Operating System

If your CPU is 32 bits, then you must install a 32 bit version of an Operating System.

- If you try to install a 64 bit version of an OS on to a computer with a 32 bit CPU, most likely the install process will detect the fact you have only 32 bits, and terminate the incorrect install.


If your CPU is 64 bits, then you usually can install a 32 bit or 64 bit version of an OS.


- It is HIGH recommended you install a 64 bit version to take full advantage of your 64 bit CPU.

Recall from the 'Hardware' lecture, that the number of bits in a CPU directly correspond to the amount of RAM the CPU can address.

There is a relationship between the CPU running on your computer and how much RAM that can be used on your PC.

Remember CPU and RAM are binary or base 2 hardware designs... Each byte of RAM has an address. The maximum number of addresses your CPU can access is determined by the number of number of bytes in the CPU.

SO, If you have a  CPU, then the maximum RAM is $2^{32} = 4,294,967,296$ bytes of RAM

AND, If you have a  CPU, then the maximum RAM is $2^{64} = 18,446,744,073,709,551,616$ bytes of RAM

PC: RAM Map Software download: <http://technet.microsoft.com/en-us/sysinternals/ff700229.aspx>

MAC: Activity Monitor <http://support.apple.com/kb/HT1342>

Video Watch Me: **MAX memory Limits** <http://www.youtube.com/watch?v=PFNyyGMZYS0&list=PL8B6962AEF2843CA1>

Assignment OS

- Look up on the Internet, read text book, or lecture notes above..

- 1) What is the percentage of usage for each of the top 6 Operation systems ?
- 2) Which OS will you use for this class ?
- 3) What is a Hierarchical Directory Structure ?
Which is the name and year of the Operating System first to use a 'Hierarchical' Directory Structure ?
- 4) What is the Name of the BIOS in your computer ?
What utility did you run on your computer to get info about your computers BIOS
- 5) Which line command clears the screen in the:
 - a. DOS CLI
 - b. Linux CLI
 - c. Mac CLI
- 6) What does it mean to be a 64 bit Operating System ?
- 7) Do you have 32 or 64 bit CPU ?
What is the exact brand/model of your CPU ?
- 8) Do you use a 32 or 64 bit OS ?
How did you discover this ?
- 9) How many Operating Systems are listed in the website:
http://en.wikipedia.org/wiki/Comparison_of_operating_systems
- 10) Why does understanding how an OS works help you as a programmer ?
Look at what the different OS managers do to answer this question correctly.
- 11) What was the first computer OS ever used ?