Week 01
Introduction to Operating Systems

Sirak Kaewjamnong

517-312 Operating Systems

Course overview

- Lecturer course
- Programming section
  - Thread programing with Java
  - Zero copy with java
- Linux installation and basic Linux administration
- Virtual machine understanding and practicing

Books and materials

- Presentation from the lecturer
- Books:
  - Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts, 10th edition”, Wiley, 2018.
  - Andrew S. Tanenbaum, Herbert Bos, “Modern Operating 4th edition”, Pearson, 2015.
  - สุจิตรา อดุลย์เกษม, ทฤษฎีระบบปฏิบัติการ, โปรวิชั่น, 2552

Scoring

- Class attending and activities 5%
- Midterm exam 35%
- Final exam 35%
- Assignments 25%

Questions before starting (1)

- What is OS and why we need it?
- Is OS platform extended to the hardware such as CPU?
- Why some OSs like Linux can be used in different CPU architectures e.g. InterX86, ARM ?
- An application program is extended to architecture or OS?
  - Can X86 programs run on all X86 OSs?
  - Can programs on Linux run on different architecture but the same OS?
- In multi-user OS, how the OS switch job for each user?
- How can each user run several program simultaneously? How the OS schedule tasks?

Questions before starting (2)

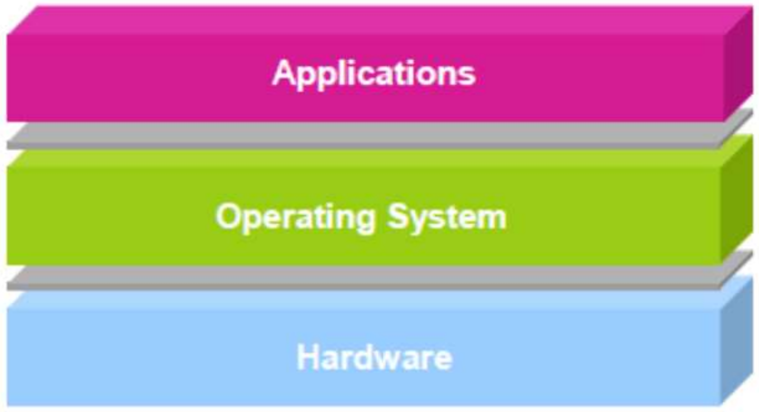
- Each application located in memory, how to make sure this memory location isn’t occupied by other applications?
- Can applications share resources such as libraries together?
- How each application(process) communicate with other applications?
- If some resources e.g. printer has several requests at the same time, how to manage it?
- When user press power-on switch, what is the first program to be executed?
- How can OS been loaded?

How large OS source code is?

- The source code of the heart of an operating system(kernel) like Linux or Windows is on the order of five million lines of code or more
- Think about printing out five million lines in book form, with 50 lines per page and 1000 pages per book. It would take 100 books to list an operating system of this size
- How long will you take to read 100 1K-pages books?
- When essential shared libraries are included, Windows is well over 70 million lines of code

OS is…?

- *Software* layer between hardware and applications



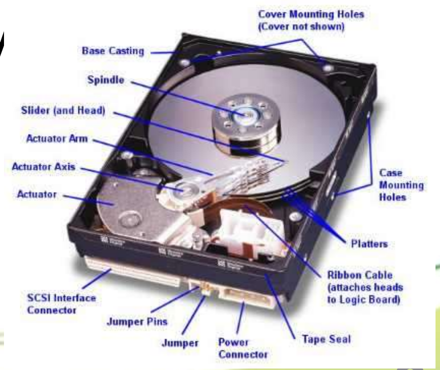
Life with and without OS

Life with an OS

```
file = open ("test.txt",
  O_WRONLY);
write (file, "test", 4);
close (file);
```

- ***Life without an OS***

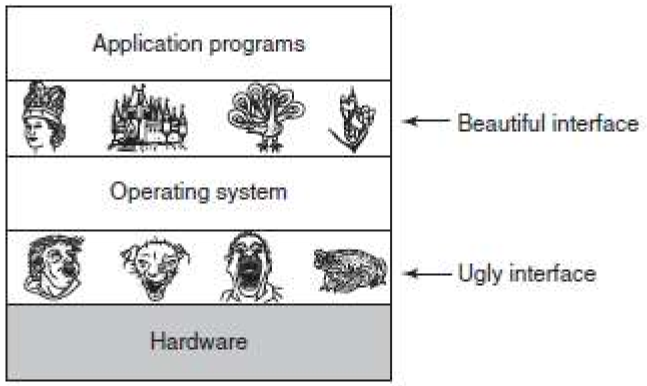
- Blocks, platter, track, and sector
- Where is this file on disk? Which platter, track, and sectors?
- Code needs to change on a different sy



What is an OS?

The OS as an Extended Machine

- OS provides layer of abstraction e.g. drivers for controlling I/O devices or disk and file is an abstraction for using disk



What is an OS?

The OS as a Resource Manager

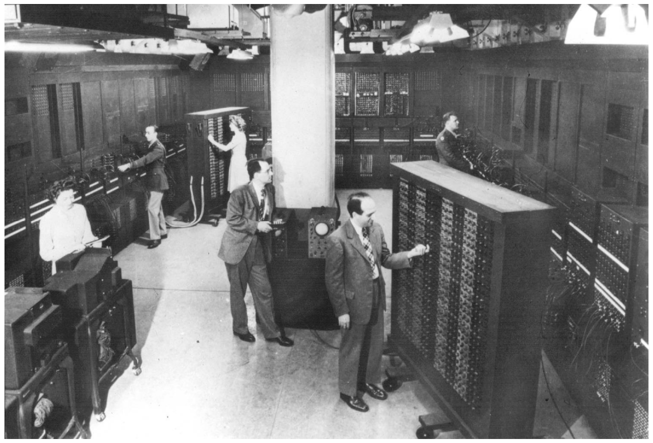
- The OS is to provide for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs wanting them
- In multi-users OS, OS keeps track of which programs are using which resources

History of OSs

- The dark age (1945 – 55): no OS
- Batch systems (1955 – 65)
- Multiprogramming (1965 – 80)
- PC (and mobile) era (1980 – present)

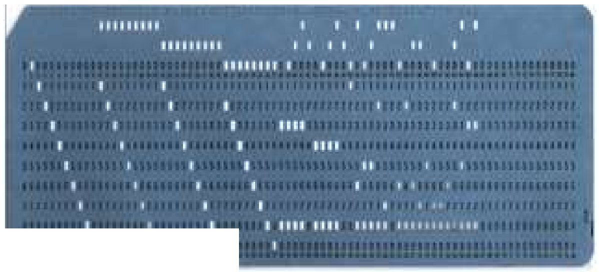
The dark age

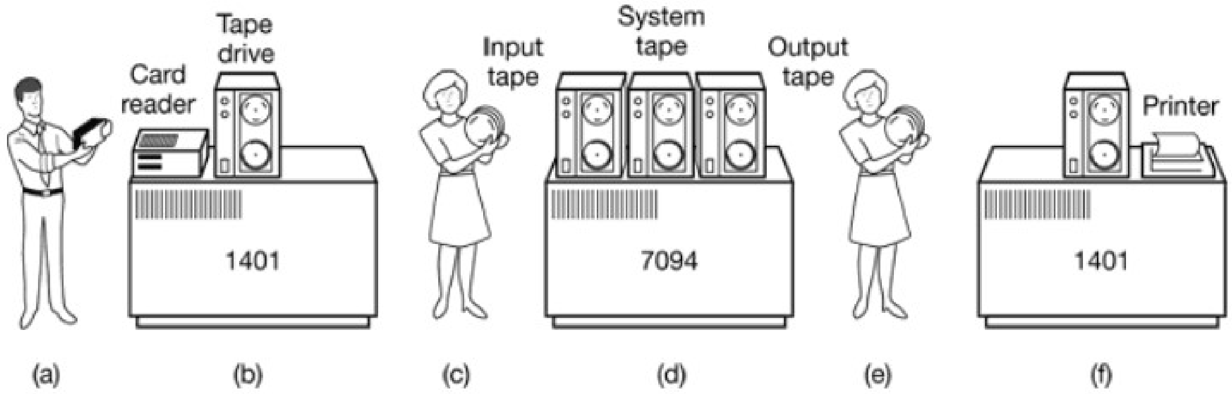
- ENIAC (1946): the first computer
- Only a single group of people designed and used it
  - Protection and virtualization are not needed!
- How to Allocate and Reclaim computation time?
  - Sign-up sheet on the wall!



Batch systems (1955-65)

- Background:
  - At that time, computers are used only to “compute” (instead of entertainments, etc.)
  - Users write program using “punch card”
    - *Punch your program into cards*
    - *Bring the cards to computer operators*
    - *Come back after a day to get result*





Batch systems (1955-65)

- OS:
  - Read the first job from the tape
  - Run it
  - Write the output to another tape
  - Read the next job and repeat the process

  - Similar to the “sign-up sheet on the wall”, only now this process is automated

Multiprogramming (1965-80)

- Problems with Batch systems?
  - Responsiveness
    - Do you want to wait for a day just to find out your program doesn’t compile?
  - Multiple users cannot concurrently access the computer
  - Efficiency
    - CPU is idle while the computer is doing I/O
- Multiprogramming
  - Multiple tasks are performed during the same period of time
  - As if they are executed concurrently
  - Now multiple users can use the same machine simultaneously

MULTICS

- MULTiplexed Information and Computing Service
    - Initiated by MIT, Bell Labs, and General Electric
- Designed to support hundreds of users on a machine far less powerful than iPhone 5
    - People knew how to write small, efficient programs in those days
- Technically successful, but not so much commercially
    - Bell Labs and GE dropped out before it was released

UNIX

- Written by Ken Thompson and Dennis Ritchie from Bell Labs on PDP-11 (1971)
  - Project started because Ken wanted to play the “Space Travel” game without MULTICS
  - Originally named as “Unics” by Brian Kernighan
- In 1973, Ritchie invented C programming language to ease the development of Unix

Linux

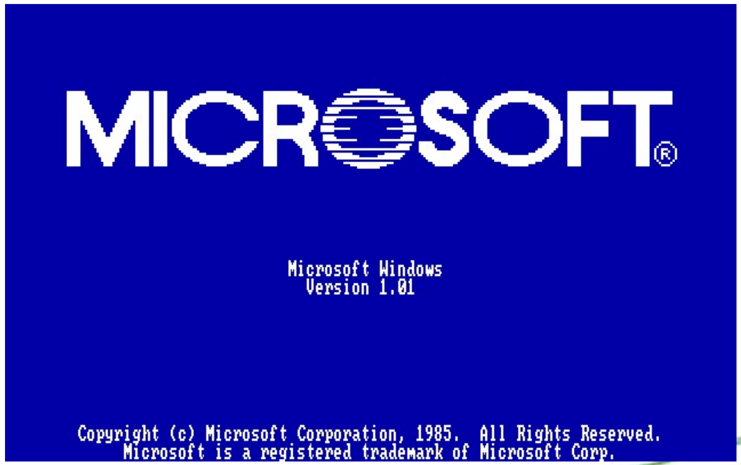
- In 1991, Linus Torvalds, then a student of Univ. of Helsinki, wanted to learn OS
  - But at that time, no free and open-source OS is available
  - Decided to write his own OS and “open-source” it
  - Open-source is the key behind its popularity today

PC-era (1980-present)

- IBM introduces PC in 1981, using Intel processors
  - Looking for an OS for its PC
- At that time, an OS called CP/M (Control Program/Monitor) is already working on Intel CPUs
  - Bill Gates initially suggested IBM to contract CP/M
  - CP/M founder refused to meet with IBM
  - IBM came back to Bill Gates, and he purchased an OS named DOS (Disk Operating System), modified it, and renamed it MS-DOS

Graphical User Interface (GUI)

- The interface of the early OSs is command-line
- Researchers in Xerox-PARC built the first OS with graphic user interface
  - Steve Jobs visited PARC, saw the GUI, and used it in Apple’s Lisa (1983), later Macintosh (1984)
  - Microsoft introduced Windows in 1985



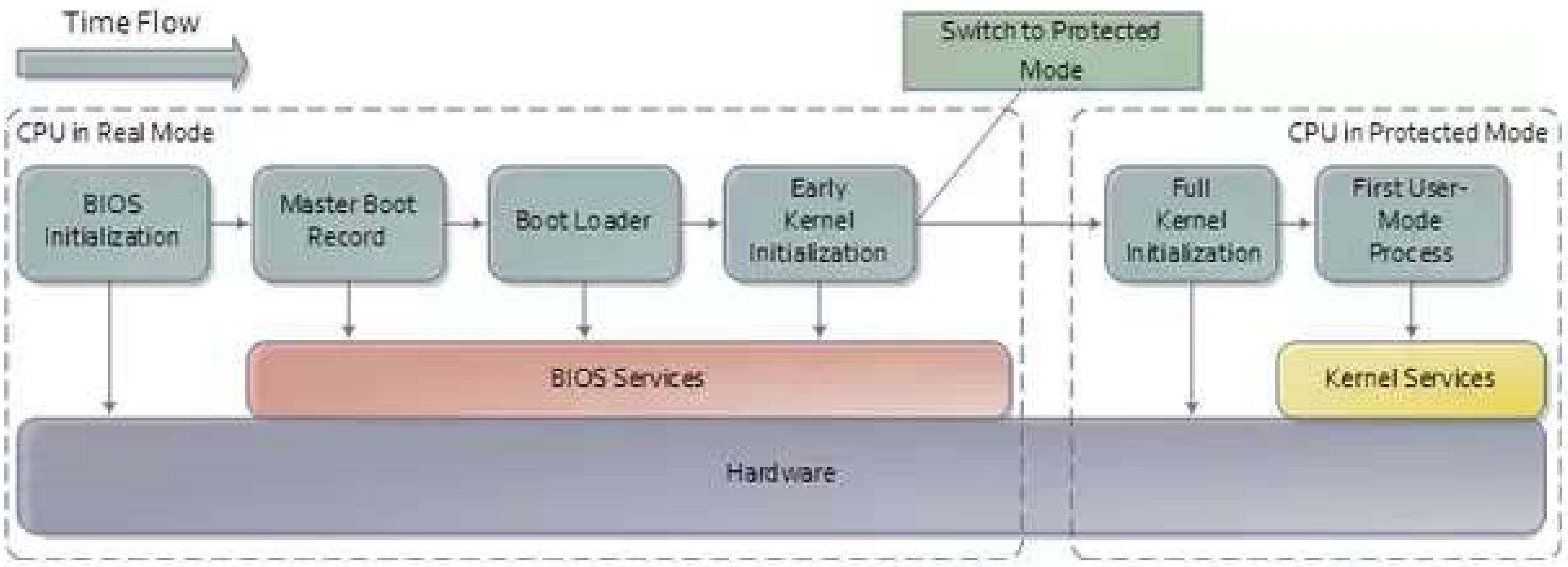
MICROSOFT®

Microsoft Windows
Version 1.01

Copyright (c) Microsoft Corporation, 1985. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

Booting a computer

Booting a computer



Booting a computer

Intel-based (IA–32) startup

- An IA–32-based PC is expected to have a BIOS (Basic Input/Output System, which comprises the bootloader firmware) in non-volatile memory
- When the CPU is reset at startup, the computer starts execution at memory location 0xffff0
  - The location at 0xffff0 is actually at the end of the BIOS ROM and contains a jump instruction to a region of the BIOS that contains start-up code

Booting a computer

Upon start-up, the BIOS goes through the following sequence:

1. Power-on self-test (POST)
2. Detect the video card's (chip's) BIOS and execute its code to initialize the video hardware
3. Detect any other device BIOSes and invoke their initialize functions
4. Display the BIOS start-up screen
5. Perform a brief memory test (identify how much memory is in the system)
6. Set memory and drive parameters
7. Configure Plug & Play devices (traditionally PCI bus devices)
8. Assign resources (DMA channels & IRQs)
9. Identify the boot device

When the BIOS identifies the boot device (typically one of several disks that has been tagged as the bootable disk), it reads block 0 from that device into memory location 0x7c00 and jumps there.

Source : https://www.cs.rutgers.edu/~pxk/416/notes/02-boot.html

Booting a computer

Master Boot Record

- This first disk block, block 0, is called the Master Boot Record (MBR)
  - It contains the first stage boot loader.
  - Since the standard block size is 512 bytes, the entire boot loader has to fit into this space.
  - The contents of the MBR are:
    - First stage boot loader (≤ 440 bytes)
    - Disk signature (4 bytes)
    - Disk partition table, which identifies distinct regions of the disk (16 bytes per partition × 4 partitions)

Booting a computer

Volume Boot Record

- Once the BIOS transfers control to the start of the MBR that was loaded into memory, the MBR code scans through its partition table and loads the Volume Boot Record (VBR) for that partition.
- The VBR is a sequence of consecutive blocks starting at the first disk block of the designated partition.
  - The first block of the VBR identifies the partition type and size and contains an Initial Program Loader (IPL), which is code that will load the additional blocks that comprise the second stage boot loader

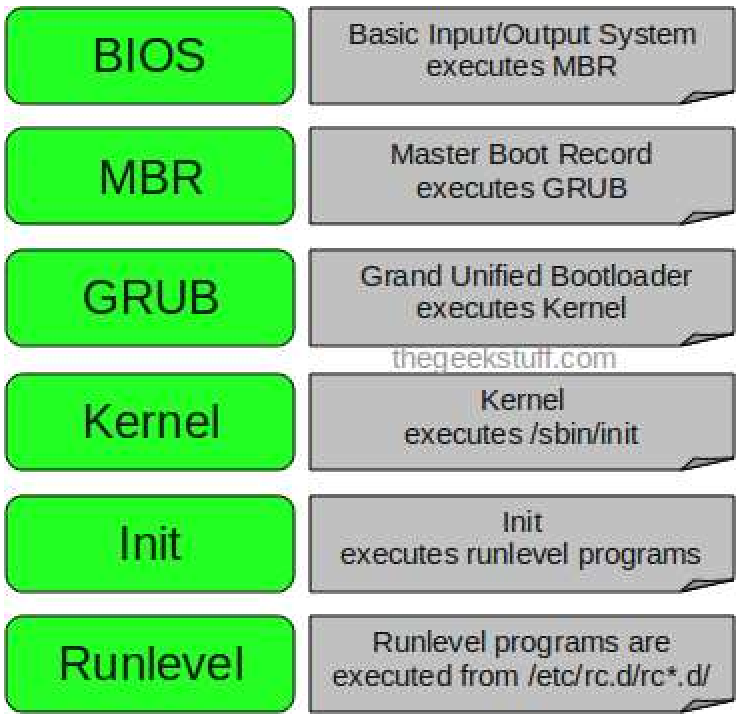
Booting Windows

- On Windows NT-derived systems (e.g., Windows Server 2012, Windows 8, 10), the IPL loads a program called NTLDR, which then loads the Windows operating system.
  - NTOSKRNL.EXE is the kernel image of the Windows NT base OS
  - HAL.DLL is the Hardware Abstraction Layer library. It is a core kernel level driver that allows the OS to interact with the specific computer hardware

https://www.quora.com/What-is-the-step-by-step-booting-process-of-Windows

Booting Linux

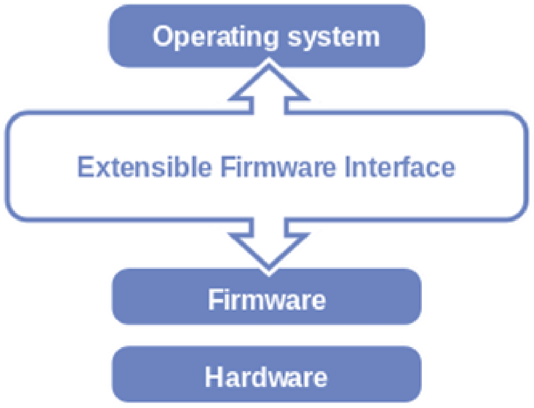
- In Linux boot loader is GRUB (GRand Unified Bootloader)


BIOS
Basic Input/Output System executes MBR
MBR
Master Boot Record executes GRUB
GRUB
Grand Unified Bootloader executes Kernel
thegeekstuff.com
Kernel
Kernel executes /sbin/init
Init
Init executes runlevel programs
Runlevel
Runlevel programs are executed from /etc/rc.d/rc*.d/

https://www.thegeekstuff.com/2011/02/linux-boot-process

Unified Extensible Firmware Interface(UFFI)

- Intel set out to create a specification of a BIOS successor that had no restrictions on having to run the startup code in 16-bit mode with 20-bit addressing
  - It is used by many newer 64-bit systems
  - Also have legacy BIOS support for running Windows



- With UEFI, there is no longer a need for the Master Boot Record to store a stage 1 boot loader

UEFI

- UEFI reads the GUID (Globally Unique IDentifier) Partition Table (GPT), which is located in blocks immediately after block 0
- The GPT describes the layout of the partition table on a disk, from this, the EFI boot loader identifies the EFI System Partition
- This system partition contains boot loaders for all operating systems that are installed on other partitions on the device
  - Windows systems, UEFI loads the *Windows Boot Manager* (bootmgfw.efi)
  - For Linux, an EFI-aware version of GRUB

UEFI features

- BIOS components :preserved some components from the BIOS, including power management (Advanced Configuration & Power Interface, ACPI) and system management components (e.g., reading and setting date).
- Support for larger disks: while the BIOS only supported four partitions per disk, with a capacity of up to 2.2 TB per partition. UEFI supports a maximum partition size of 9.4 ZB (9.4 × 10^21 bytes).
- No need to start up in 16-bit (real) mode: The pre-boot execution environment gives you direct access to all of system memory.
- Device drivers: UEFI includes device drivers, including the ability to interpret architecture-independent EFI Byte Code (EBC) which generally used only for the boot process.

UEFI features

- Boot manager: The old BIOS only had the smarts to load a single block, which necessitates multi-stage booting process. UEFI has its own command interpreter and complete boot manager. A dedicated boot loader is no longer needed. As long as the bootable files is placed into the UEFI boot partition, which is formatted as a FAT file system (the standard file system format in older Windows systems; one that just about every operating system knows how to handle).
- Extensibility: The firmware is extensible. Extensions to UEFI can be loaded into non-volatile memory.

References

- Andrew S. Tanenbaum, Herbert Bos, Modern Operating 4th edition, Pearson, 2015
- Presentations by Ding Yuan, ECE Dept., University of Toronto
- https://www.cs.rutgers.edu/~pxk/416/notes/02-boot.html
- https://www.quora.com/What-is-the-step-by-step-booting-process-of-Windows
- https://www.thegeekstuff.com/2011/02/linux-boot-process
- https://www.howtogeek.com/56958/htg-explains-how-uefi-will-replace-the-bios/