Operating Systems

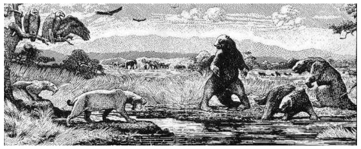
CSCI 4730 / CSCI 6730

Maria Hybinette

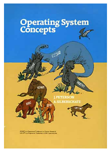
Book: Dinosaurs? & Dinosaur Wars

***Frederick P. Brooks**' Mythical Man-Month (1975). Description of the software crises - likens **large scale programming** to a tar pit*

> *No scene from prehistory is quite so vivid as that of the mortal struggles of great beasts in the **tar** pits. In the mind's eye one sees **dinosaurs**, mammoths, and saber toothed tigers struggling against the grip of the **tar**. The fiercer the struggle, the more entangling the **tar**, and no beast is so strong or so skillful but that he ultimately sinks.*
>
> *Large-scale programming has over the past decade been such a **tar pit**, and many great and powerful beasts have thrashed violently in it. Most have emerged with running systems - few have met goals, schedules, and budgets.*

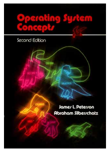


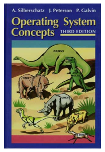
Evolution of Textbook

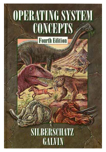


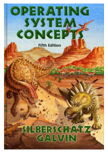
1983

1983

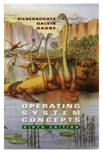


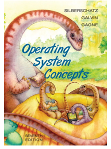




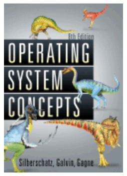


1998

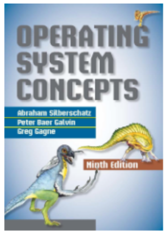




2004



2008



2013

https://galvin.info/history-of-operating-system-concepts-textbook/

Outline & Questions

- What is an Operating Systems (OS)?
- What does an OS do?
- What is an OS and what is it not?
- How do I run an OS?
- How does an Operating System run?
- What is the basic structure?
- Computer System Component Architecture

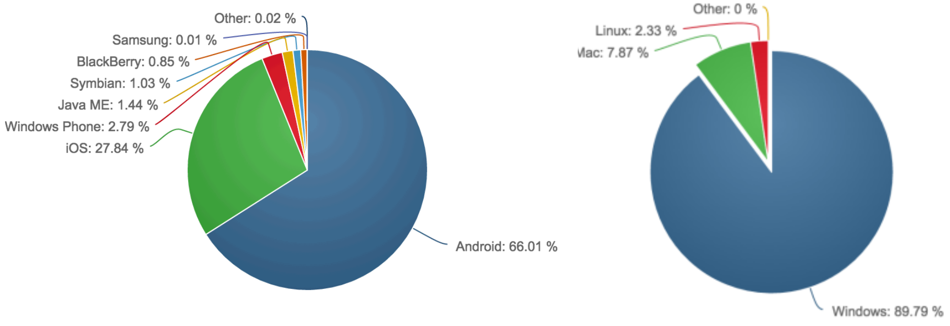
Questions?

- What are the major operating system components?
- What are basic computer system organizations?
- How do you communicate with the operating systems?
- What services are (need to be) provided?

Poll?

- What desktop/laptop OS do you have?
- Which desktop/laptop OSs are you familiar with?
- What do you think the market share (%) is (portion of different OSs)?
  - Windows
  - Mac
  - Linux
  - Other
- What about the mobile OS market share?
  - iOS
  - Android
  - Window
  - Other
- Are they different?

Popularity: OS Market Share Desktop vs. Mobile

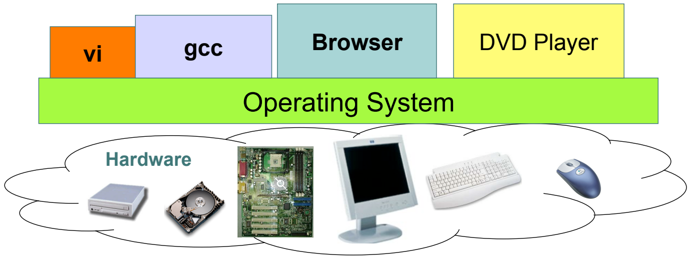


http://www.w3counter.com/

http://netmarketshare.com/

Based on page views : July 2016

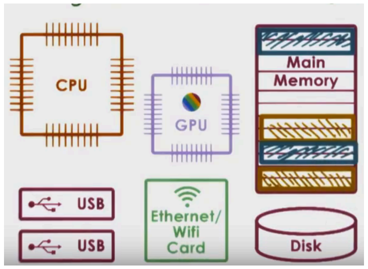
What is an Operating System



- **Software** between applications and hardware
  - Hide hardware complexity
  - Manages resources:
    - Make finite resources infinite
  - Provide protection and isolation

Courtesy: Kai Li, Princeton

What is an Operating System



```
CPU: Scheduling Processes on CPU
Disk: Read And Write Files
Memory: Manage Running Processes Data

Communication: Message Send and Receive
```

- **Software** between applications and hardware
  - Hide hardware complexity
  - Manages resources:
    - Make finite resources infinite
  - Provide protection and isolation

Courtesy: Ada Gavrilovska, G-Tech
Courtesy: Kai Li, Princeton

What is an Operating System?

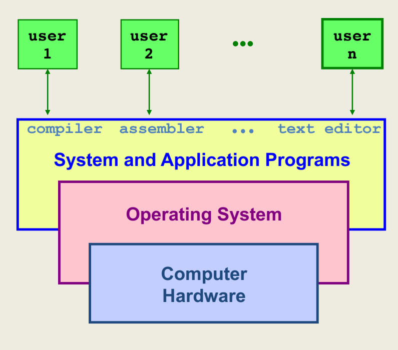
- An intermediary between a user of a computer and the computer hardware.
- A hardware manager
- A Protector

- **Goals:**
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.
  - Combination of the above.
  - Handhelds (convenience), Mainframes/Servers (efficiency)

Where is the OS? Computer System Layers

Computer system can be divided roughly in four components:

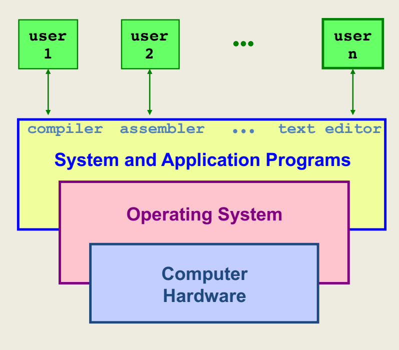
- Hardware
- Operating system
- Application programs
- Users



Computer System Layers

Computer system can be divided roughly in four components:

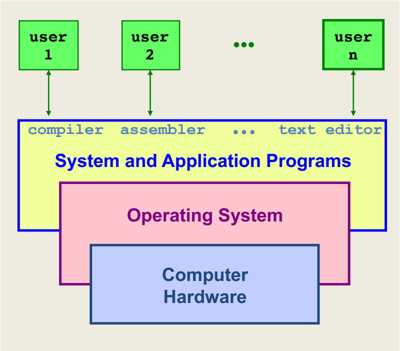
- Hardware:
  - provides basic computer resources:
    - CPU, Memory, I/O Devices



Computer System Layers

Computer system can be divided roughly in four components:

- Hardware
- Operating system:
  - Controls and coordinates use of hardware among various applications and users.



user 1 user 2 ••• user n

compiler assembler ••• text editor

System and Application Programs

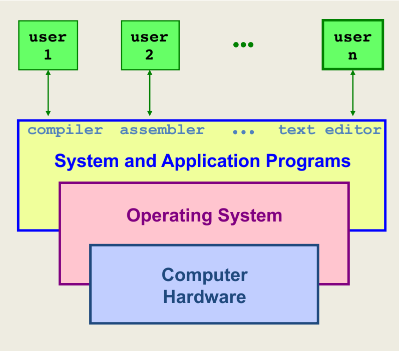
Operating System

Computer Hardware

Computer System Layers

Computer system can be divided roughly in four components:

- Hardware
- Operating system
- Application programs
  - define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games



user 1 user 2 ••• user n

compiler assembler ••• text editor

System and Application Programs

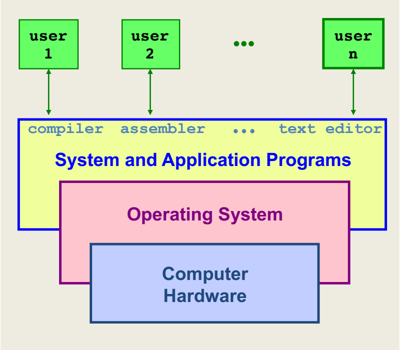
Operating System

Computer Hardware

Computer System Layers

Computer system can be divided roughly in four components:

- Hardware
- Operating system
- Application programs
- Users
  - People, machines, other computers



user 1 user 2 ••• user n

compiler assembler ••• text editor

System and Application Programs

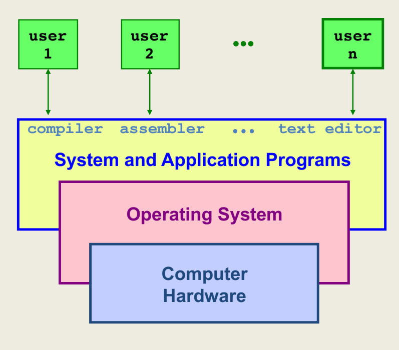
Operating System

Computer Hardware

Computer System Layers

Computer system can be divided roughly in four components:

- Hardware
- Operating system
- Application programs
- Users



user 1 user 2 ••• user n

compiler assembler ••• text editor

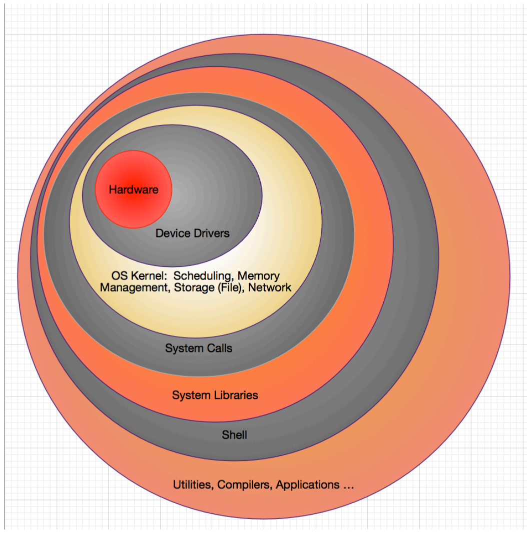
System and Application Programs

Operating System

Computer Hardware

Between The Layers: Interfaces

- Device Drivers
  - Hardware specific software
- System Calls
  - Gateway to ”Kernel Space”
- Shell
  - User Interface



Hardware

Device Drivers

OS Kernel: Scheduling, Memory Management, Storage (File), Network

System Calls

System Libraries

Shell

Utilities, Compilers, Applications …

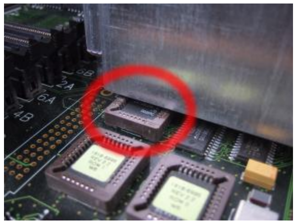
What "Makes up” the Operating System?

- *Hardware → Drivers -- Operating System – Kernel → Applications → Users*
- *Surprise! No universally accepted definition!*
  - *A GRAY border*
- *“Everything a vendor ships when you order an operating system”* used to be a good approximation… But varies wildly
- Operating System is just the “Kernel”
  - the one program that runs at all times on the computer
  - everything else is either a
    - system program (ships with the operating system) or an
    - application program

What Does Operating Systems Do?

- A Space/Time Controller
- [Space] It *allocates resources*
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- [Time] It *controls execution of running programs (processes)*
  - Controls execution of programs to prevent errors and improper use of the computer

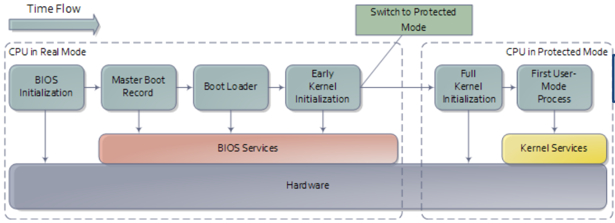
Overview: The Computer Startup Process

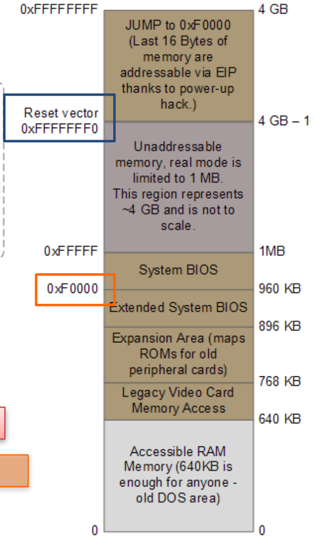


- A *bootstrap program* (initial program) is loaded at power-up or reboot (it itself is called by an instruction at a specific ‘known address’
  - Stored in firmware in ROM/EEROM
  - Stored on a chip on the mother board (‘parent board’)
- Initializes all aspects of system
- At some later point the operating system kernel is loaded (e.g., from disk) and starts execution
- Pentium initial bootstrap program that loads the OS is called the system Basic Input Output System or BIOS.

Visual of the Time-Line of the Booting Sequence (more detailed)

http://duartes.org/gustavo/blog/post/how-computers-boot-up



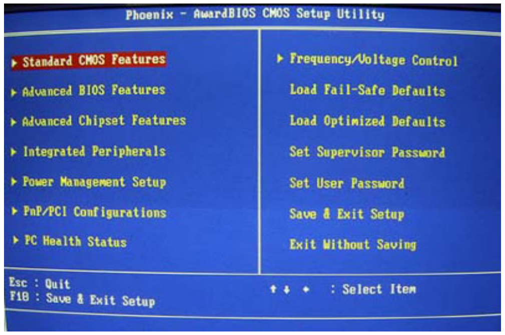

0xFFFFFFFF
4 GB
JUMP to 0xF0000 (Last 16 Bytes of memory are addressable via EIP thanks to power-up hack.)
Reset vector 0xFFFFFFF0
4 GB – 1
Unaddressable memory, real mode is limited to 1 MB. This region represents ~4 GB and is not to scale.
0xFFFFF
1MB
System BIOS
0xF0000
960 KB
Extended System BIOS
896 KB
Expansion Area (maps ROMs for old peripheral cards)
768 KB
Legacy Video Card Memory Access
640 KB
Accessible RAM Memory (640KB is enough for anyone - old DOS area)
0
0

Visualizing the booting sequence - please read the below web page for more detail (HW).

First instruction executed (Intels) : 0xFFFFFFF0 (reset vector)

First instruction is a jump to BIOS entry point

CMOS/BIOS Configuration Utility



Phoenix - AwardBIOS CMOS Setup Utility

- Standard CMOS Features
- Advanced BIOS Features
- Advanced Chipset Features
- Integrated Peripherals
- Power Management Setup
- PnP/PCI Configurations
- PC Health Status

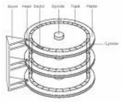
- Frequency/Voltage Control
- Load Fail-Safe Defaults
- Load Optimized Defaults
- Set Supervisor Password
- Set User Password
- Save & Exit Setup
- Exit Without Saving

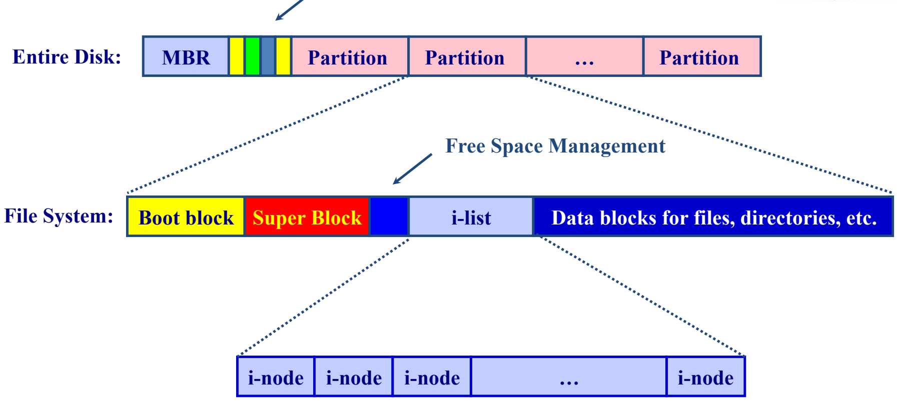
Esc : Quit
F10 : Save & Exit Setup

↑ ↓ ← → : Select Item

Booting from hard disk

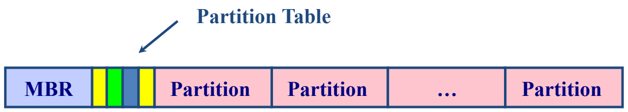
Disk is divided into 1+ partitions: one file system per partition





Entire Disk & Booting Computer

- Master Boot Record (sector 0) - Contains a program, a boot loader) that examines partition table for an 'active' partition that contains the secondary boot loader.
  - used to boot computer
- Partition Table
  - Located at the end of MBR and contains starting and ending address of each partition
- "A program (e.g., the system Basic Input Output System or BIOS for Pentiums)" reads in and executes the MBR
  - searches for first active partition (noted in the partition table)
  - reads in its first block (the boot block) and executes it



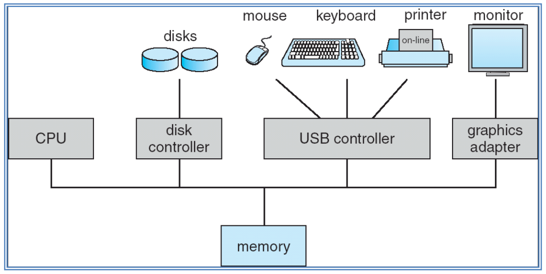
Partition Layout

- Boot block:
  - contains a hardware specific program that is called automatically to load "UNIX" at system startup time
- Super block:
  - file system type, #blocks in file system
- Free space management (two lists):
  - a chain of free data block numbers
  - a chain of free i-node numbers
- i-list/i-node table:
  - administrative information about a file (meta-data: name, type, location, size, protection bits, …) structured into an array: inode table or simply the i-list
  - An i-node number:
    - uniquely identifies a file in a file system
    - is an index to the i-node table

Boot block	**Super Block**		**i-list**	**Data blocks for files, directories, etc.**

Free Space Management

Computer System Organization

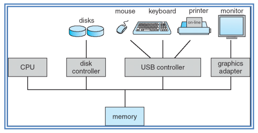


- One or more CPUs, device controllers connect through *common bus* providing access to shared memory
- Concurrent execution of CPUs and devices *compete* for memory cycles

- How do devices communicate to the OS? For example telling the OS to (when) check for user input?

Computer System Operations

- CPU: the processor that perform the actual computation
- I/O controll/ers:
  - take commands in registers, generate flags and interrupts
  - each device controller
    - is in charge of a particular device type
    - has a *local buffer for I/O*
    - Examples: audio – output device, mouse – input, disk – I/O.
- CPU moves data from/to main memory to/from local buffers.
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an *interrupt*.



Communication: Interrupts

An interrupt is a signal to the processor to temporarily suspend execution because some system event needs handling (alert!).

- Occurrence of an event is *signaled* by interrupts either by software or hardware
  - A *trap* is a software-generated interrupt caused either by an error or a user request.
- Modern operating systems are *interrupt driven.*
- *OS Stops what it is doing, preserve the current state and then handles the interrupt (=overhead) .*

Terminology

- Uni-programming
- Multi-programming
- Multiprocessing
- Multithreading
- Multitasking

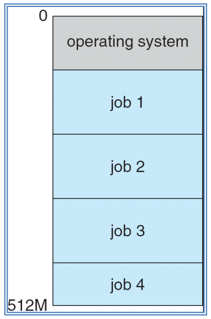
Uni-programming

- One process [in memory] at one time
  - Looking ahead – doesn't need memory protection of other processes. [OS is protected from processes by checking addresses used buy the process]

Multi-programming

Multiprogramming needed for efficiency

- Single user cannot keep CPU and I/O devices busy at all times
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via job scheduling
- When it has to wait (for I/O for example), OS switches to another job



memory layout

User View: *Timesharing* (Multitasking)

Timesharing (multitasking) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing

- Response time should be < 1 second
- Each user has at least one program executing in memory ⇨process
- If several jobs ready to run at the same time ⇨ CPU scheduling
- If processes don't fit in memory, swapping moves them in and out to run
- Virtual memory allows execution of processes not completely in memory

Definition of Terms

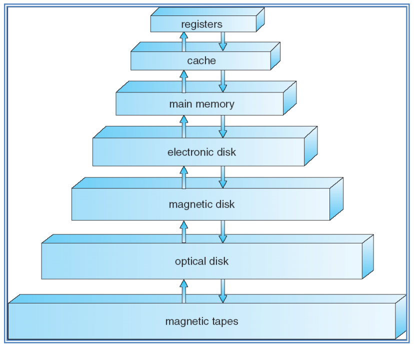
- Uniprogramming
  - *one process at the time.*
- Multiprogramming
  - multiple processes (with separate address spaces) concurrently on a machine (more on this later)
- Multiprocessing
  - running programs on a machine with multiple processors.
- Multithreading
  - multiple threads per *address space (later)*.
- Multitasking
  - a single user can run multiple processes.

Storage Structure

- Main memory – only large storage media that the CPU can access directly.
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity.

Storage Structure & Hierarchy

- Storage systems organized in hierarchy.
  - Speed
  - Cost
  - Volatility
- *Caching* – copying information into faster storage system; main memory can be viewed as a last *cache* for secondary



An Operating System’ s Core Tasks

- Process Managements
- Memory Managements
- File Managements
- I/O System Managements
- Protection System

Process Management

- A process is a program in execution (an active entity, i.e. a running program)
  - Basic unit of work on a computer
  - Examples: compilation process, word processing process
  - A process needs certain resources:
    - e.g. CPU time, memory, files, I/O devices to accomplish its task
- Each user can run many processes at once (e.g. using &)
  - One process:
    - cat file1 file2 &
  - Two processes:
    - ls | wc -l
- A time sharing system (such as UNIX) run several processes by multiplexing between them

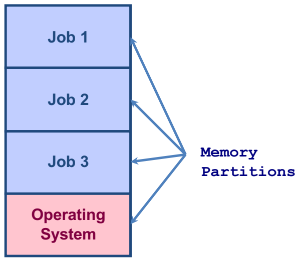
Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

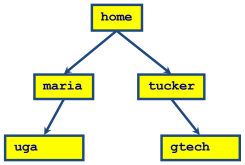
Memory Management

- Programs become processes when they are loaded into memory and start executing.
  - All data in memory before and after processing
  - All instructions in memory in order to execute
- Memory management determines what is in memory when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed



File Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit
    - A File : OS maps logical files to physical devices
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media



Mass-Storage Management

- Main memory is volatile and limited in size
  - Use disks to store ‘overflow’ and data that needs to be persistent.
- Disks are *slower* than main memory and processors
  - Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS mass storage management activities:
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

I/O Subsystem Management

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including
    - buffering (storing data temporarily while it is being transferred),
    - caching (storing parts of data in faster storage for performance),
    - spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware devices

Protection and Security

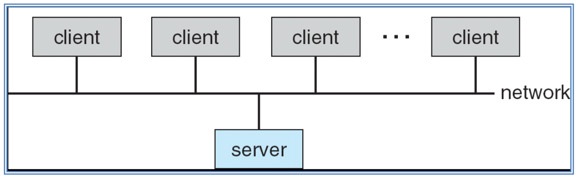
- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
- Security – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (user IDs, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file
  - Privilege escalation allows user to change to effective ID with more rights

Computing Environments

- Traditional computer
  - Blurring over time
  - Office environment
    - PCs connected to a network, terminals attached to mainframe or minicomputers providing batch and timesharing
    - Now portals allowing networked and remote systems access to same resources
  - Home networks
    - Used to be single system, then modems
    - Now firewalled, networked

Computing Environments (Cont.)

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now servers, responding to requests generated by clients
    - Compute-server provides an interface to client to request services (i.e. database)
    - File-server provides interface for clients to store and retrieve files



Peer-to-Peer Computing

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
- Node must join P2P network
  - Registers its service with central lookup service on network, or
  - Broadcast request for service and respond to requests for service via discovery protocol
- Examples include Napster and Gnutella

Web-Based Computing

- Web has become ubiquitous
- PCs most prevalent devices
- More devices becoming networked to allow web access
- New category of devices to manage web traffic among similar servers: load balancers
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

Summary

- An Operating System (from here on OS) is a software (a program) that performs two functions:
  - it extends the “use” of the computer hardware and
  - it manage the computer system resources

Reading Assignment

- Read
  - Chapters 1 & 2 Book
- Read about the boot process:
  - http://www.bbc.co.uk/news/technology-11430069?ref=nf
  - http://en.wikipedia.org/wiki/Booting
  - http://duartes.org/gustavo/blog/post/how-computers-boot-up/
  - https://neosmart.net/wiki/mbr-boot-process/