Operating Systems

Lecture 1 : Introduction to Operating Systems

Gaurav Somani
gaurav@lnmiit.ac.in

Structure

- OS Concepts

  Process, File, I/O and Memory Management

- OS Case Studies

  Windows, UNIX, UNIX derivatives and some handheld and Embedded Operating Systems.

- Practical Implementation

  Processes, Threads, Virtual Memory, File Systems and Shell Experiments (Linux Platform)

Course Evaluation

Course Evaluation*

1. Mid Term Examinations (40)
2. Final Examination (45)
3. Home Assignments (15)

Lab Evaluation*

1. Lab Assignments (60)
2. Project (40)

*Random Attendances can effect the total.

Recommended Books

1. [Galvin] Operating System Principals, Seventh Edition, Silberschatz, Galvin and Gagne. http://www.os-book.com/
2. [Tanenbaum] Modern Operating Systems. Second Edition, Andrew. S. Tanenbaum. http://www.cs.vu.nl/~ast/books/mos2/
3. [Stallings] Operating Systems- Internals and Design Principals, Fifth Edition, William Stallings. http://williamstallings.com/OS/OS5e.html

Group Communication

Group name: Operating Systems

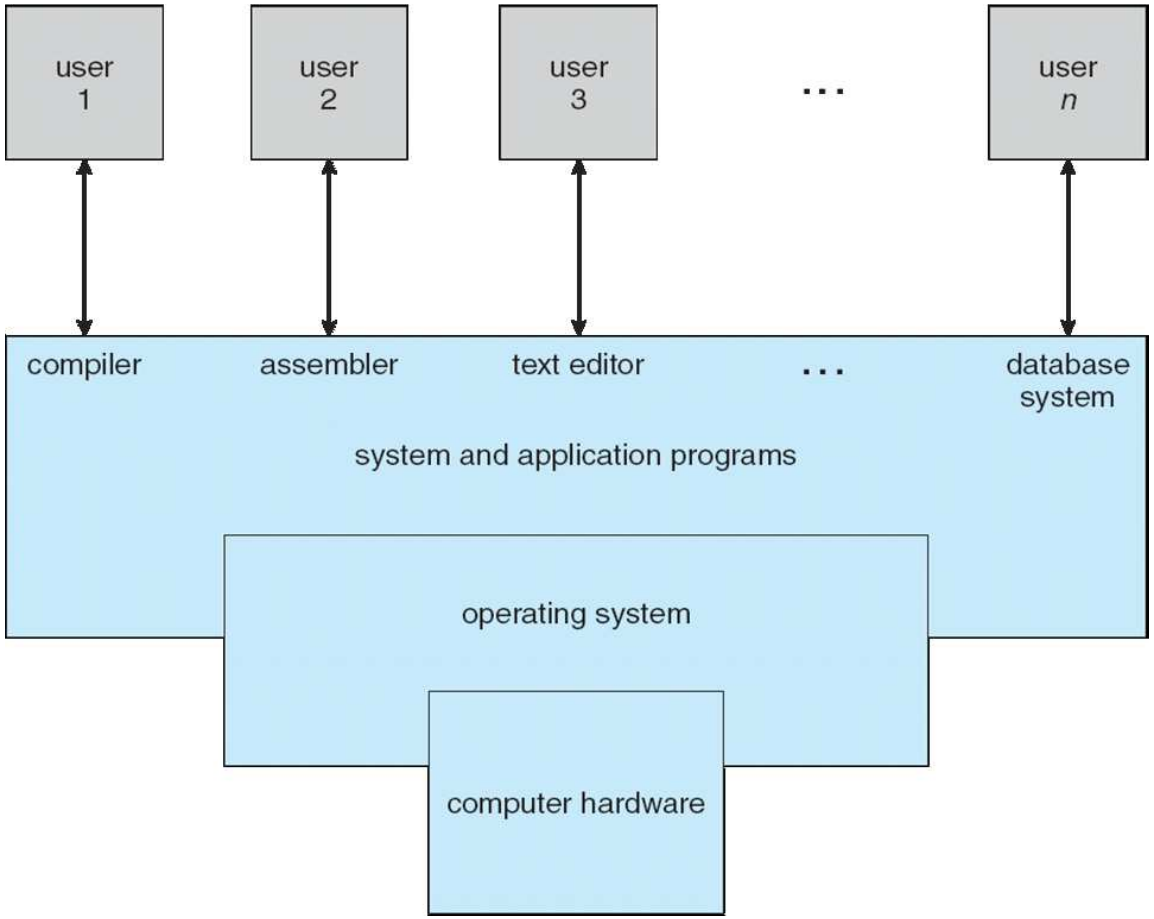
Group home page: http://groups.google.co.in/group/os-lnmiit

Group email address os-lnmiit@googlegroups.com

Computer System Structure

- Computer system can be divided into four components
  - Hardware - provides basic computing resources
    - CPU, memory and I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application/System 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 and video games
  - Users
    - People, machines and other computers

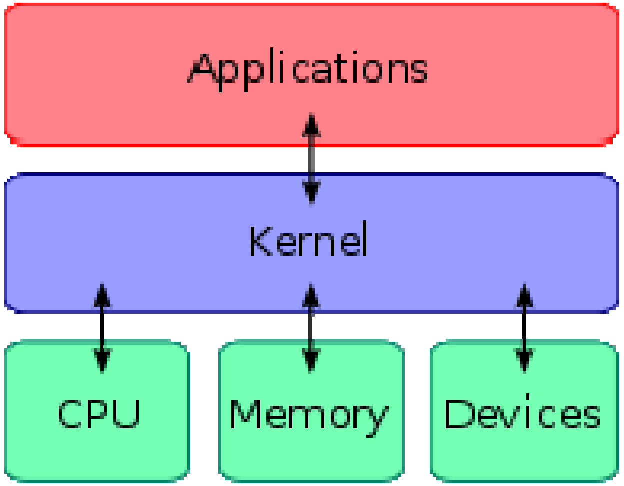
Computer System Structure



Definition of an OS

- No single Definition
- An Interface between computer hardware and user
- It is an extended machine
  - Hides the messy details which must be performed.
  - Presents user with a *virtual machine*, easier to use.
- It is a resource manager
  - Each program gets time with the resource.
  - Each program gets space on the resource.

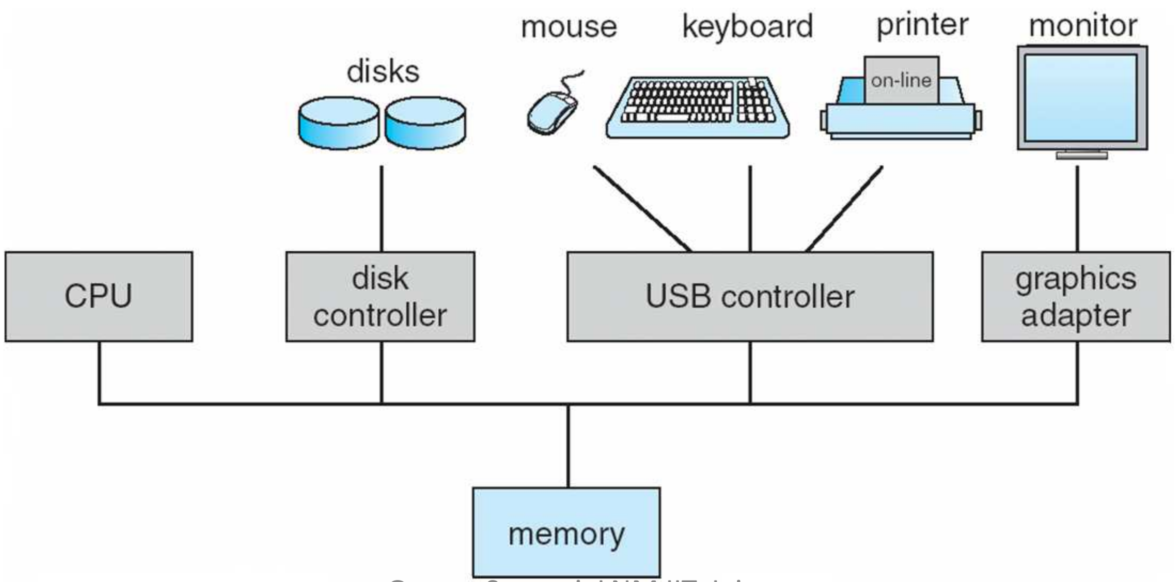
Kernel



“Kernel is traditionally used to denote the part of the operating system that is mandatory and common to all other software.” - Jochen Liedtke (Developer of L4 Microkernel)

Computer System Organization

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



Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- 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*

Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*
- A *trap* is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

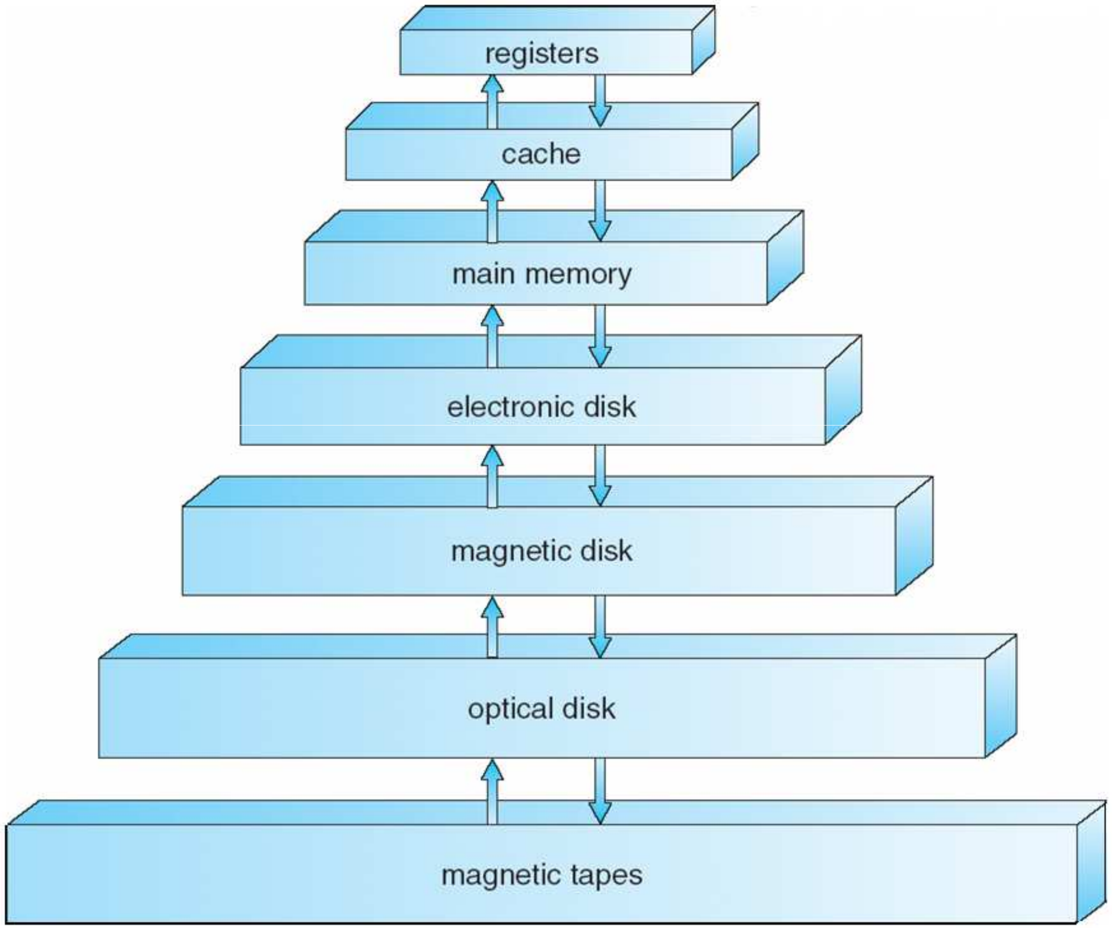
Storage Structure

- Main memory – only large storage media that the CPU can access directly
- Secondary storage – extension of main memory that provides large nonvolatile storage capacity
- Magnetic disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer

Storage 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 storage

Storage-Device Hierarchy



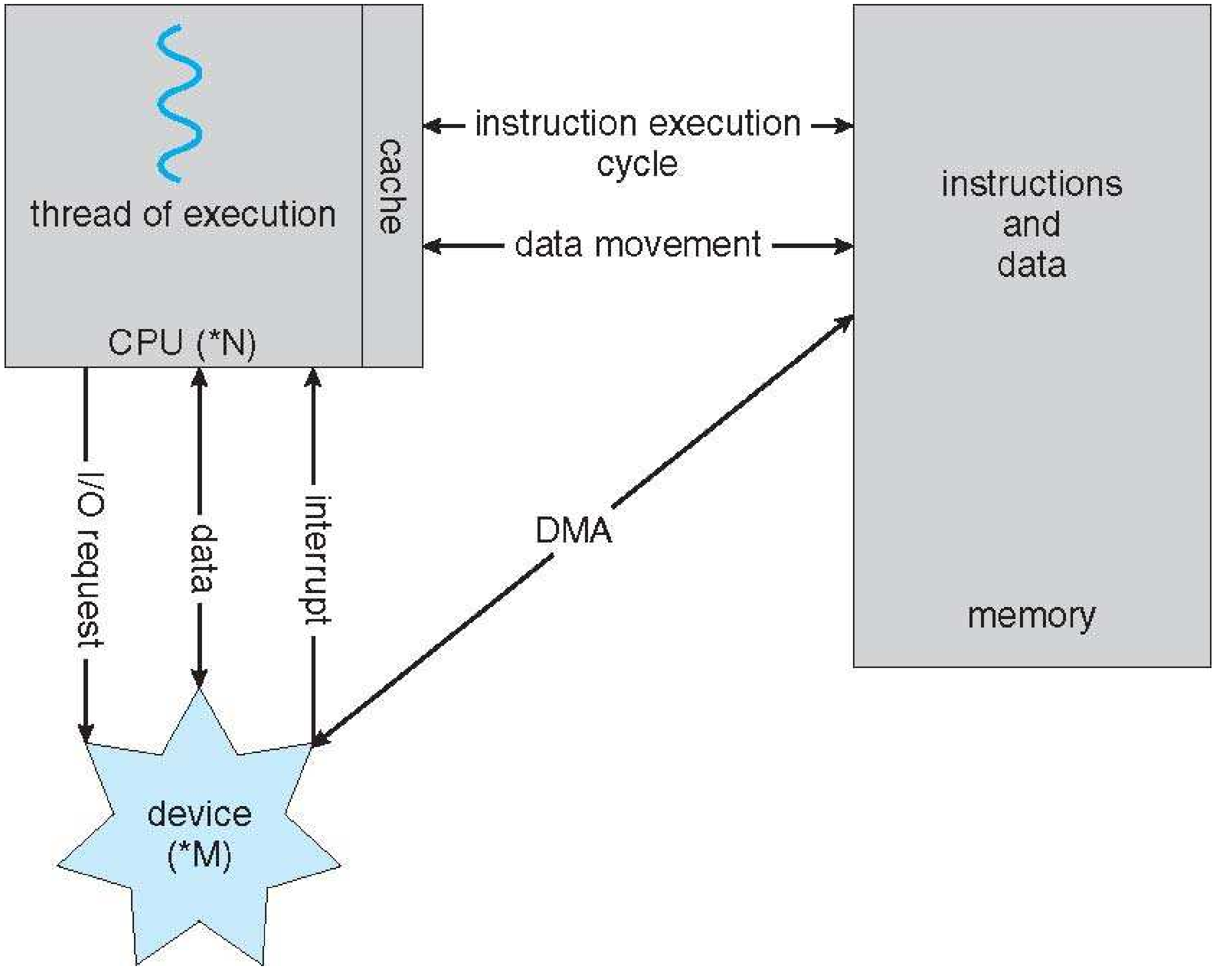
Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management - important design problem
  - Cache size and replacement policy

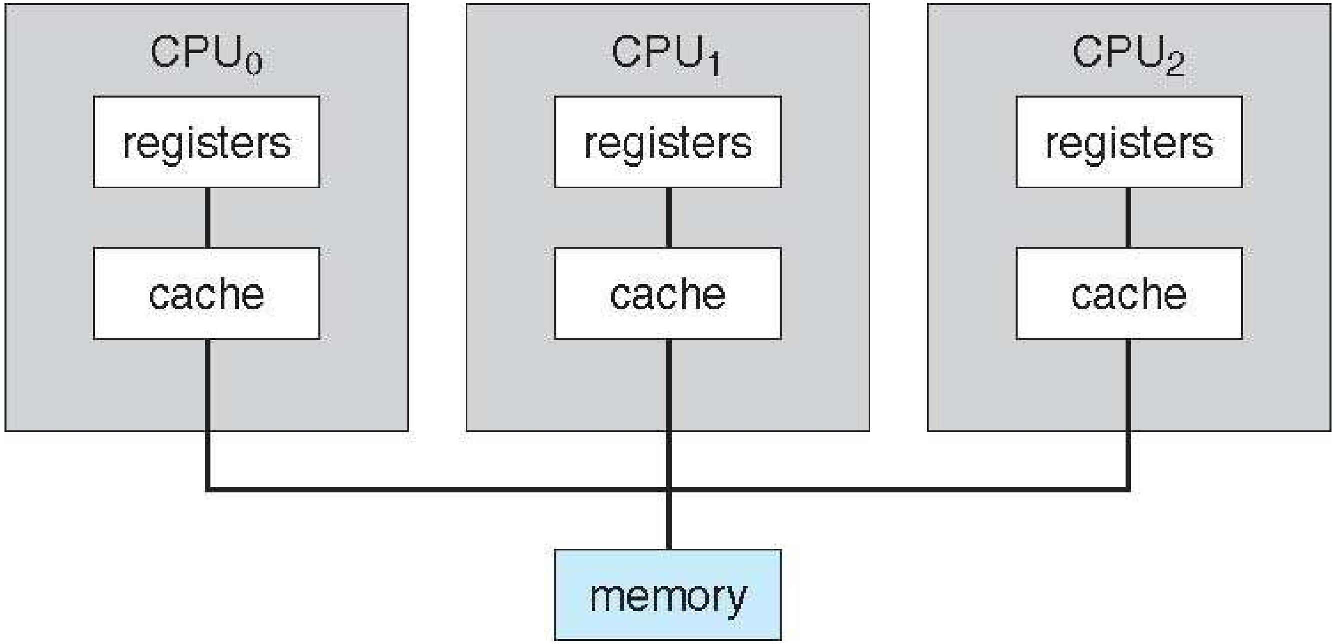
Computer-System Architecture

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
  - Also known as parallel systems, tightly-coupled systems
  - Advantages include
    1. Increased throughput
    2. Economy of scale
    3. Increased reliability – graceful degradation or fault tolerance
  - Two types
    1. Asymmetric Multiprocessing
    2. Symmetric Multiprocessing

How a Modern Computer Works



Symmetric Multiprocessing (SMP) Architecture

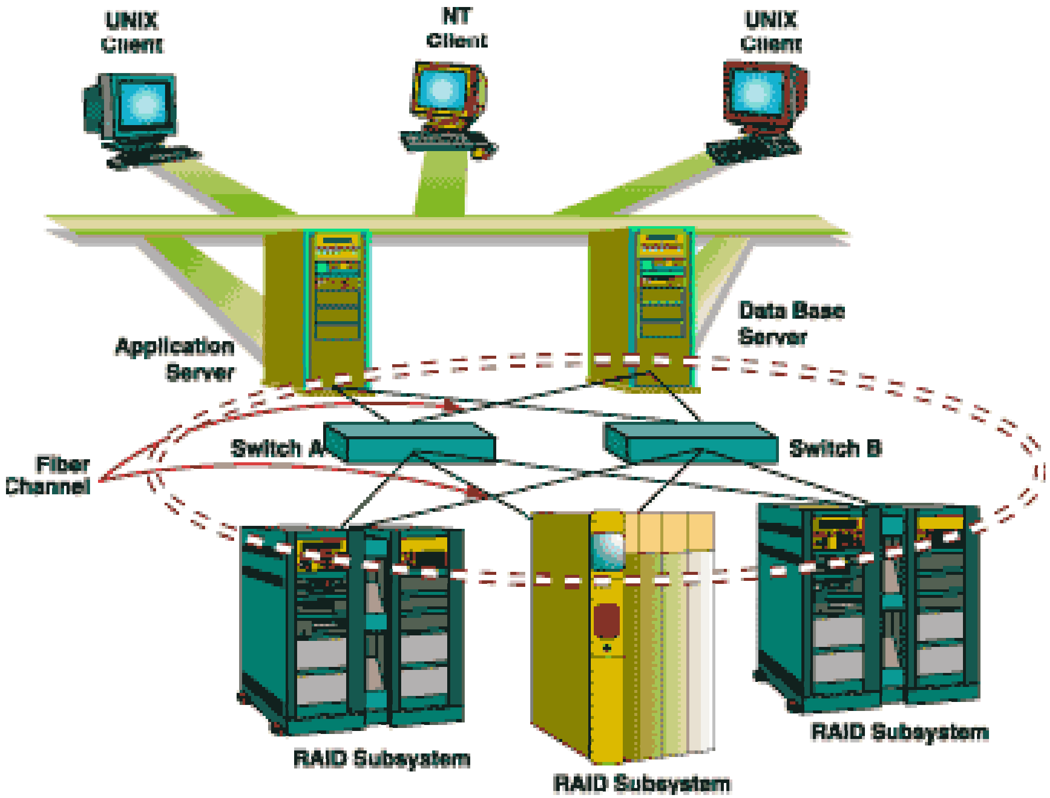


Systems that treat all CPUs equally are called symmetric multiprocessing (SMP) systems. In systems where all CPUs are not equal, system resources may be divided in a number of ways, including **asymmetric multiprocessing** (ASMP)

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a storage-area network (SAN)
  - Provides a high-availability service which survives failures
    - Asymmetric clustering has one machine in *hot-standby mode (hot-standby mode machine does nothing but monitor the active server. If that server fails than it becomes the new active server)*
    - Symmetric clustering has multiple nodes running applications, monitoring each other.
  - Some clusters are for high-performance computing (HPC)
    - Applications must be written to use parallelization
      - MPI, OpenMP and Clik++

Storage Area Network(SAN)



RAID : Redundant Array of Inexpensive Disks

Image Source : http://www.storagesearch.com/auspexart.html

Operating System Structure

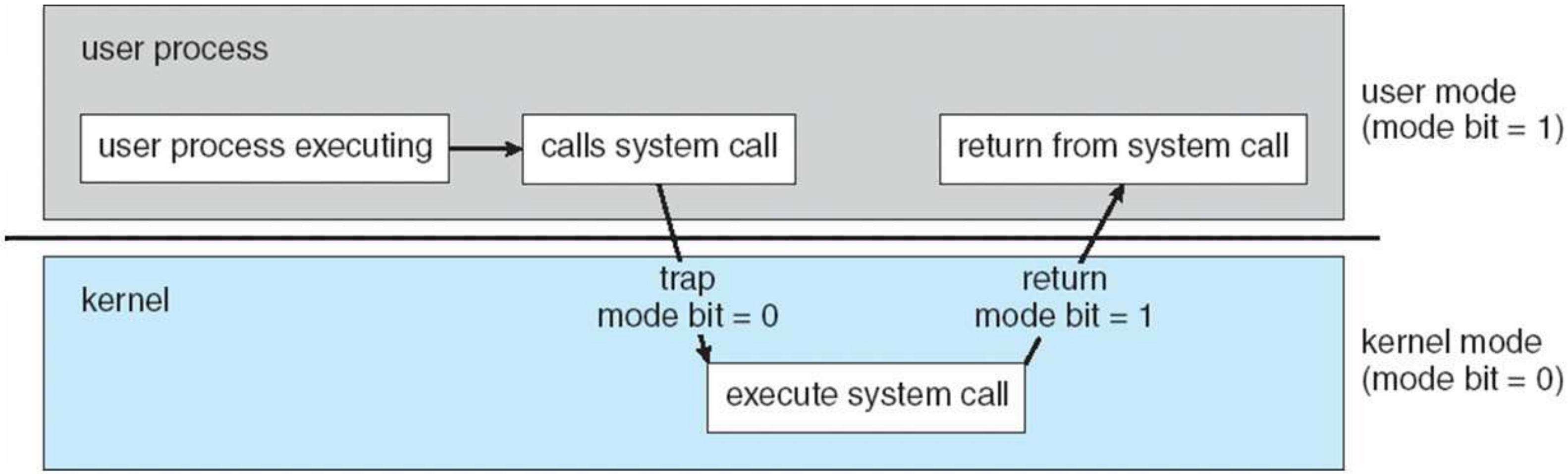
- **Multiprogramming** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**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

Operating-System Operations

- Interrupt driven by hardware
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

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

- 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

Storage Management

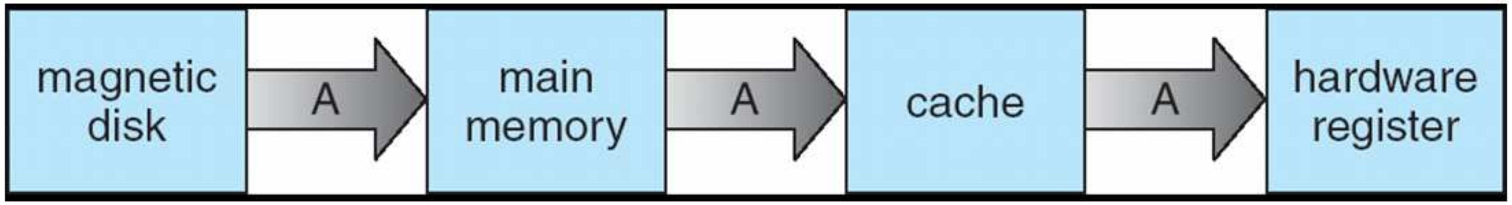
- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential 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 directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS 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)

Migration from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy



- Multiprocessor environment must provide cache coherency in hardware such that all CPUs have the most recent value in their cache
- Distributed environment situation even more complex
  - Several copies can exist
  - Various solutions

I/O Subsystem

- 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

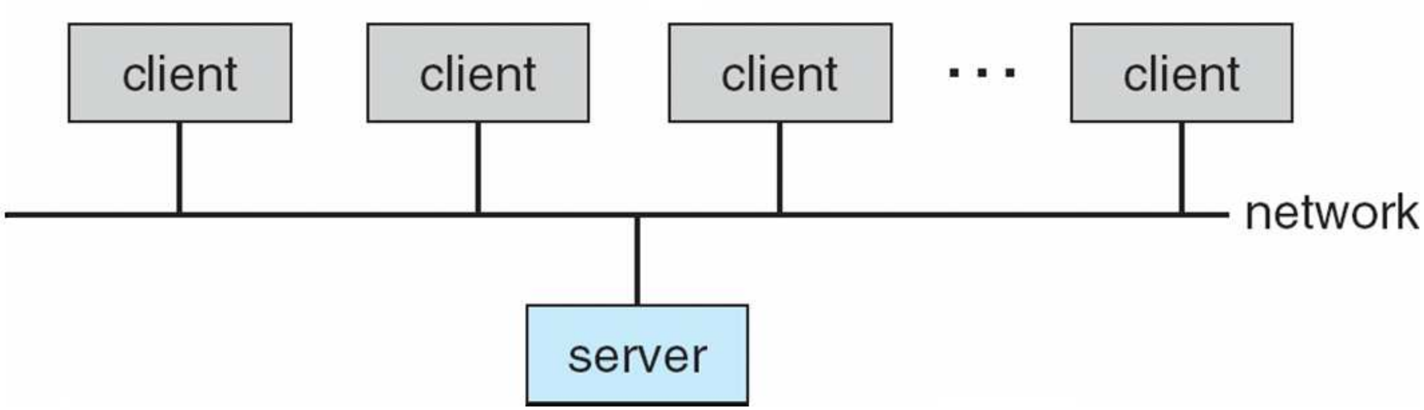
- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first 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 Linux and Windows XP as clients and servers.
- Cloud Computing or On-demand Computing.

Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary closed-source
- Counter to the copy protection and Digital Rights Management (DRM) movement
- Started by Free Software Foundation (FSF), which has "copyleft" GNU Public License (GPL)
- Examples include GNU/Linux, BSD UNIX (including core of Mac OS X), and Sun Solaris

Modern Operating Systems

Usage share of web client operating systems: November 2009

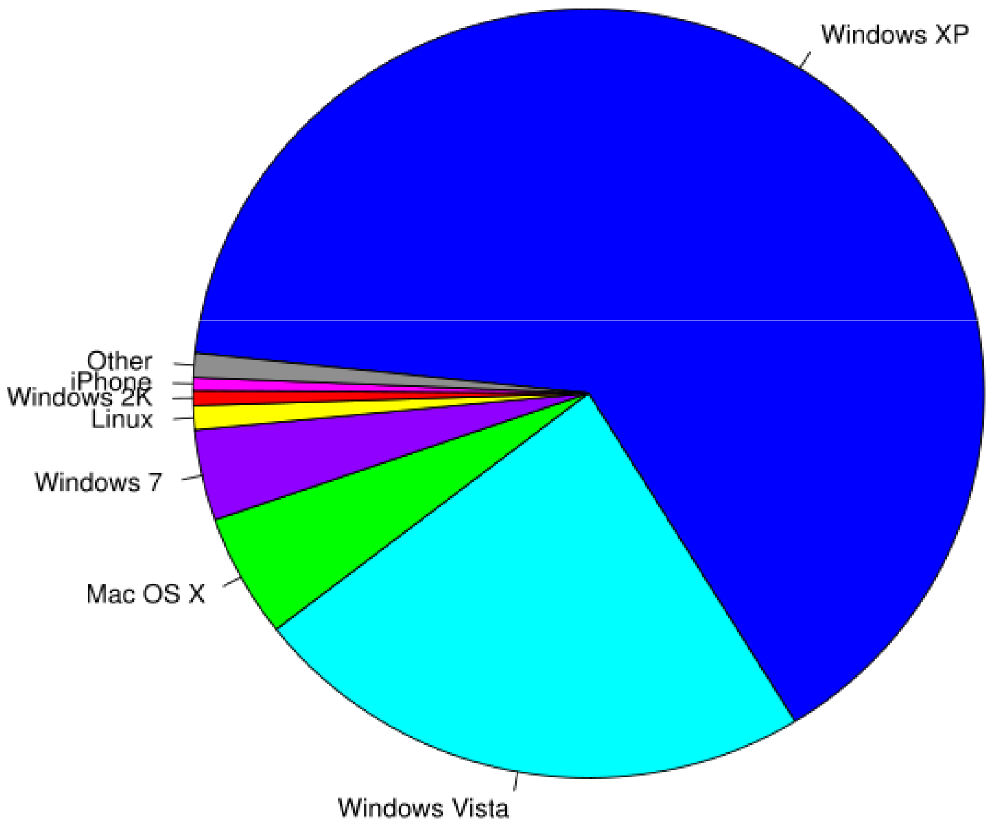


Image Source - Canalys: iPhone outsold all Windows Mobile phones in Q2 2009

Modern Operating Systems

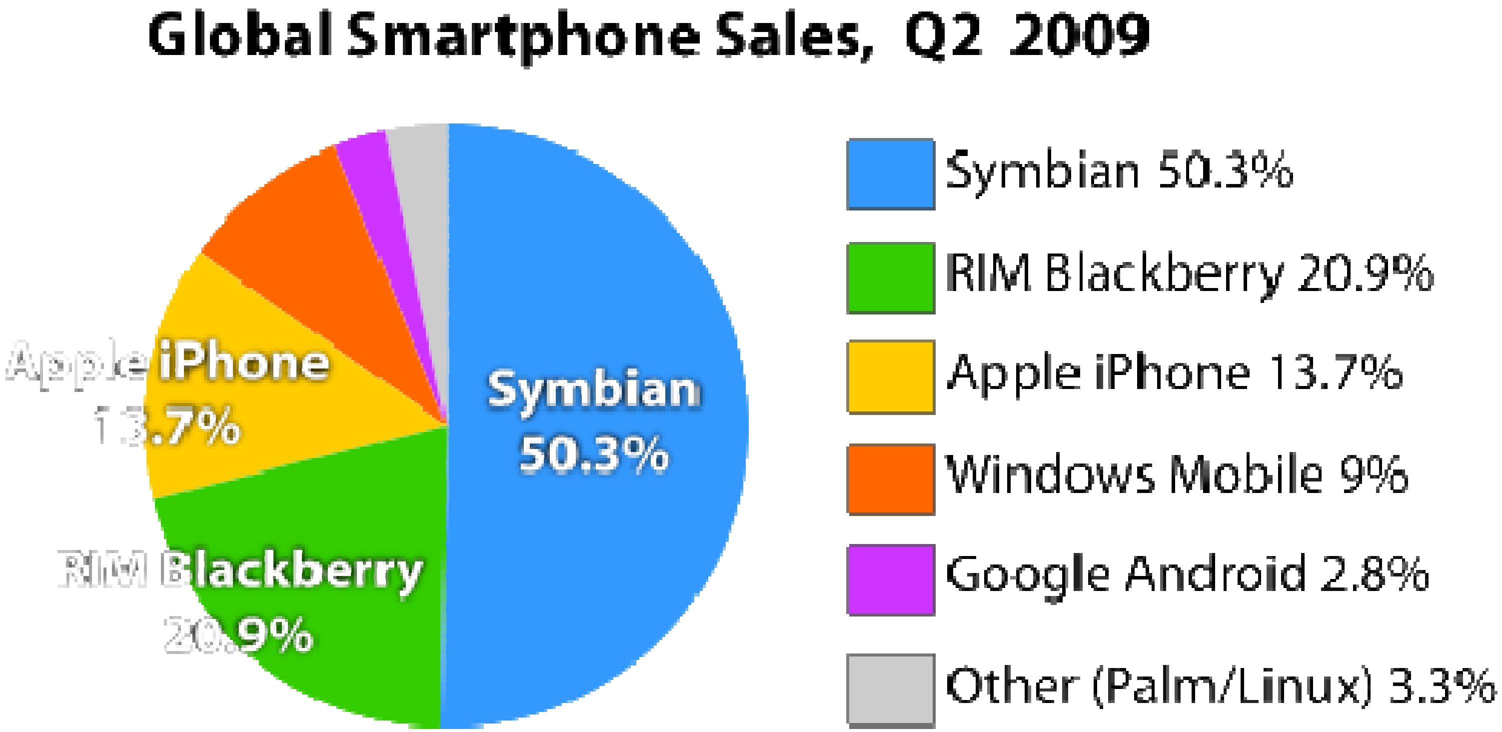


Image Source - Canalys: iPhone outsold all Windows Mobile phones in Q2 2009

Supercomputers

OS	Count	Share	Rmax (GF)	Rpeak (GF)	Processor
Linux	426	85.20%	4897046	7956758	970790
Windows	6	1.20%	47495	86797	12112
Unix	30	6.00%	408378	519178	73532
BSD	2	0.40%	44783	50176	5696
Mixed	34	6.80%	1540037	1900361	580693
MacOS	2	0.40%	28430	44816	5272
Totals	500	100%	6966169	10558086	1648095

http://www.top500.org/stats/list/30/osfam Nov 2007

Rmax – The highest score measured using the LINPACK benchmark suite. This is the number which is used to rank the computers. Measured in Giga-Flops.
Rpeak – This is the theoretical peak performance of the system. Measured in Gflops.

Modern Operating Systems

See some web operating systems

1. Glide OS – https://desktop.glidesociety.com
2. EyeOS - eyeos.org
3. Ghost Operating System

http://g.ho.st/vc.html?ghostId=guest

4. gOS -http://www.thinkgos.com/gos/download.html

Thanks