

IOS103

OPERATING SYSTEM Module 1: Introduction to OS



Objectives

At the end of the course, the student should be able to:

- *Define the operating system;*
- *Demonstrate the abstract view of the components of a computer system;*
- *Give the services provided by the operating system;*
- *Give the role of operating system in computer system;*
- *Give the organization and operation of computer system;*
- *Define interrupt, I/O structure and Storage structure*
- *Give different computer architectures and computer systems.*



Introduction to Operating System

- An ***operating system*** is a program that acts as an interface or an intermediary between a user of a computer and the computer hardware.
- It is a system program which controls all of the system's resources and provides an environment in which a user can execute programs.



Introduction to Operating System

Goals of Operating System:

- Execute user programs and make solving user problems easier.
- Make the computer system convenient to use.
- Use the computer hardware in an efficient manner.



Introduction to Operating System

COMPONENTS OF A COMPUTER SYSTEM

- ***Computer hardware*** – the CPU, memory and I/O devices, provides the basic computing resources.
- ***Application programs*** – such as database systems, word processors, games and business programs, define the ways in which these resources are used to solve the computing problems of the users.



Introduction to Operating System

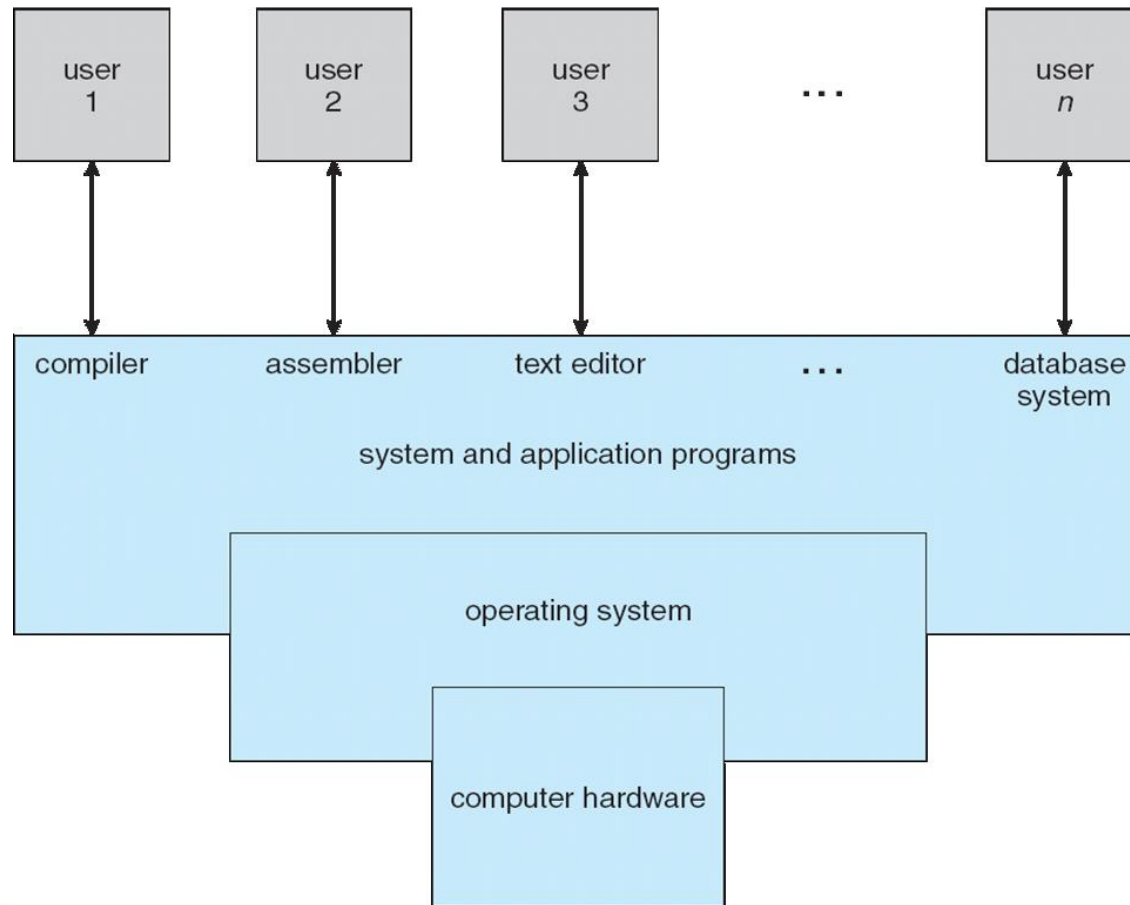
COMPONENTS OF A COMPUTER SYSTEM

- ***Users*** – there are many different users (people, machines, other computers) trying to solve different problems.
- ***Operating Systems*** – the operating system controls and coordinates the use of the hardware among the various application programs for the various users.



Introduction to Operating System

COMPONENTS OF A COMPUTER SYSTEM



Introduction to Operating System

COMPONENTS OF A COMPUTER SYSTEM

- The users of the system issue commands to their respective application programs.
- The application programs in turn communicate the requests to the operating system.
- The operating system then controls the computer hardware to carry out the command or request.

Example:

If a user instructs his word processor to print a particular file, it is actually the operating system, not the word processor, that does the printing.



Introduction to Operating System

Operating System can be thought of as:

- 1. *Resource Allocator.*** The operating system manages the different computer resources such as CPU time, memory space, file storage space, I/O devices, etc. and allocates them to different application programs and users.
- 2. *Control Program.*** The operating system controls the execution of programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of the I/O devices.



Introduction to Operating System

Services Provided by Operating System

- 1. *Program Execution.*** The operating system loads the user program in main memory and runs it.
- 2. *I/O Operations.*** The operating system is responsible for reading and/or writing data from I/O devices such as disks, tapes, printers, keyboards, etc.



Introduction to Operating System

Services Provided by Operating System

- 3. *File System Handling.*** The operating system takes care of organizing the files within the disk or tape. It takes care of the creation and deletion of files for users.

- 4. *Error Detection.*** The operating system should be able to detect errors within the computer system (CPU, memory, I/O, or user program) and take the appropriate action.



Introduction to Operating System

Computer Startup

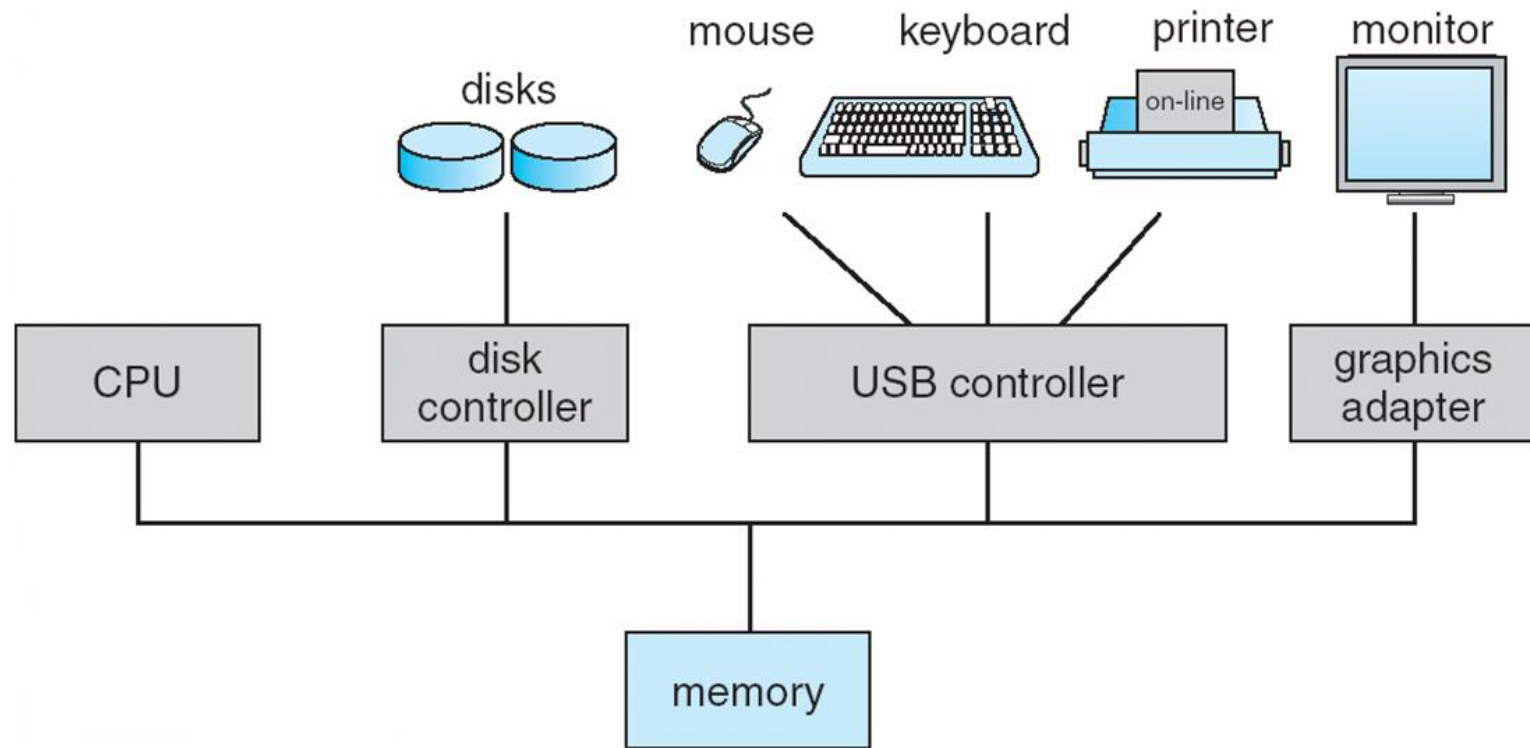
Bootstrap program is loaded at power-up or reboot.

- Typically stored in ROM or EPROM, generally known as firmware.
- Initializes all aspects of system.
- Loads operating system kernel and starts execution.



Introduction to Operating System

Computer System Organization



Introduction to Operating System

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 completing for memory cycles.



Introduction to Operating System

Computer System Organization

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***.



Introduction to Operating System

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**



Introduction to Operating System

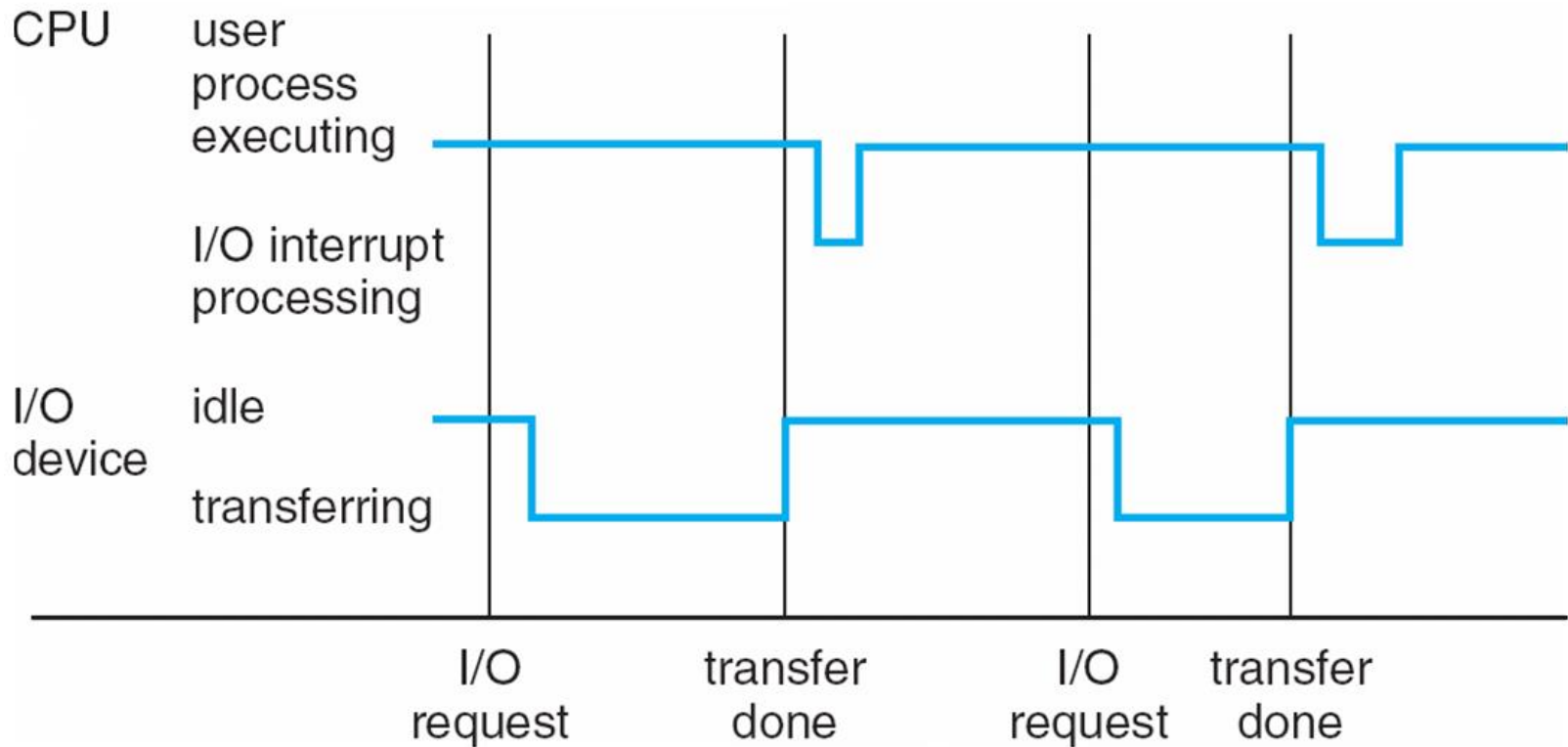
Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
 - **Polling** – operating system sends signal to each devices asking if they have a request
 - **Vectored** interrupt system – requesting device sends interrupt to the operating system.
- Separate segments of code determine what action should be taken for each type of interrupt



Introduction to Operating System

Interrupt Timeline



Introduction to Operating System

I/O Structure

After I/O starts, control returns to user program only upon I/O completion

- Wait instruction idles the CPU until the next interrupt
- Wait loop (contention for memory access)
- At most one I/O request is outstanding at a time, no simultaneous I/O processing



Introduction to Operating System

I/O Structure

After I/O starts, control returns to user program without waiting for I/O completion

- **System call** – request to the operating system to allow user to wait for I/O completion
- **Device-status table** contains entry for each I/O device indicating its type, address, and state
- Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt



Introduction to Operating System

Direct Memory Access Structure

- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte



Introduction to Operating System

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



Introduction to Operating System

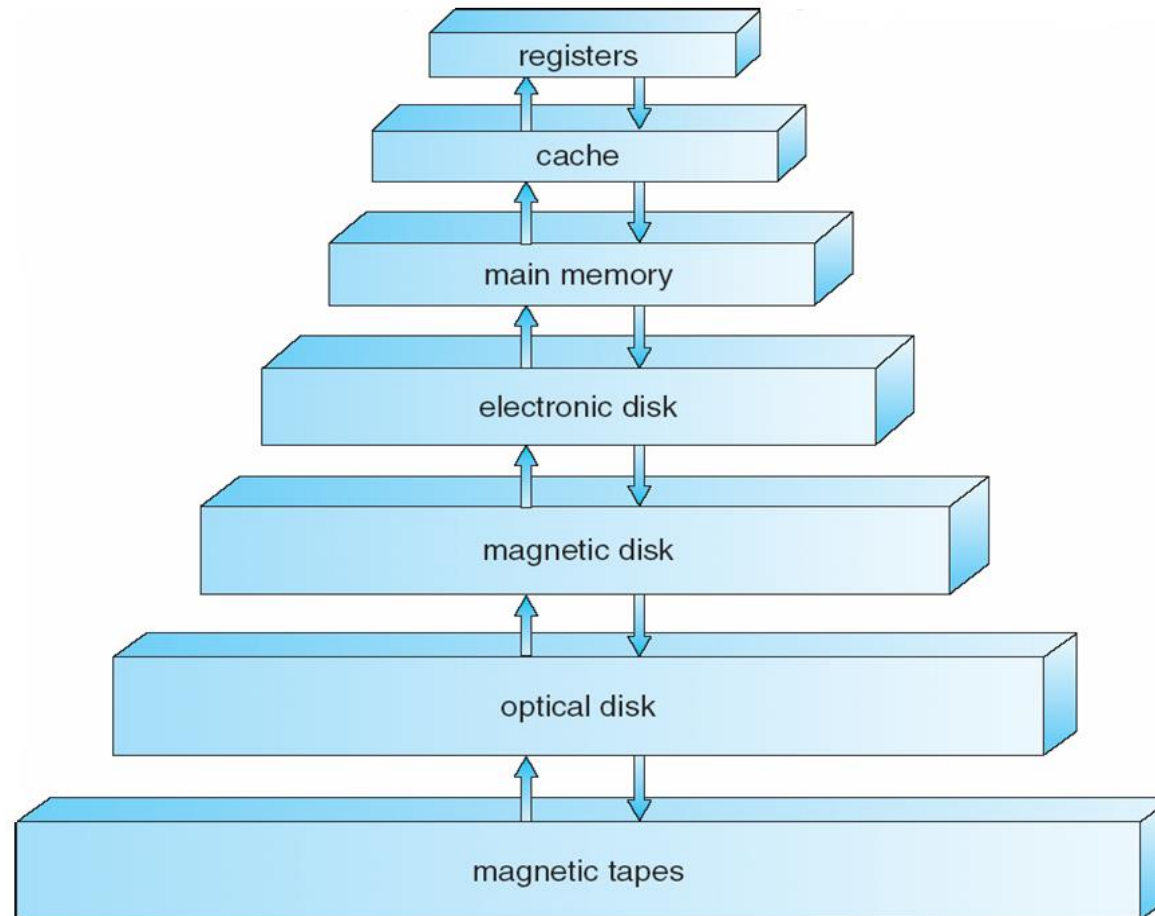
Storage Hierarchy

- 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



Introduction to Operating System

Storage-Device Hierarchy



Introduction to Operating System

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



Introduction to Operating System

Computer System Architecture

Most systems use a single general-purpose processor (PDAs through mainframes)

- Most systems have special-purpose processors as well

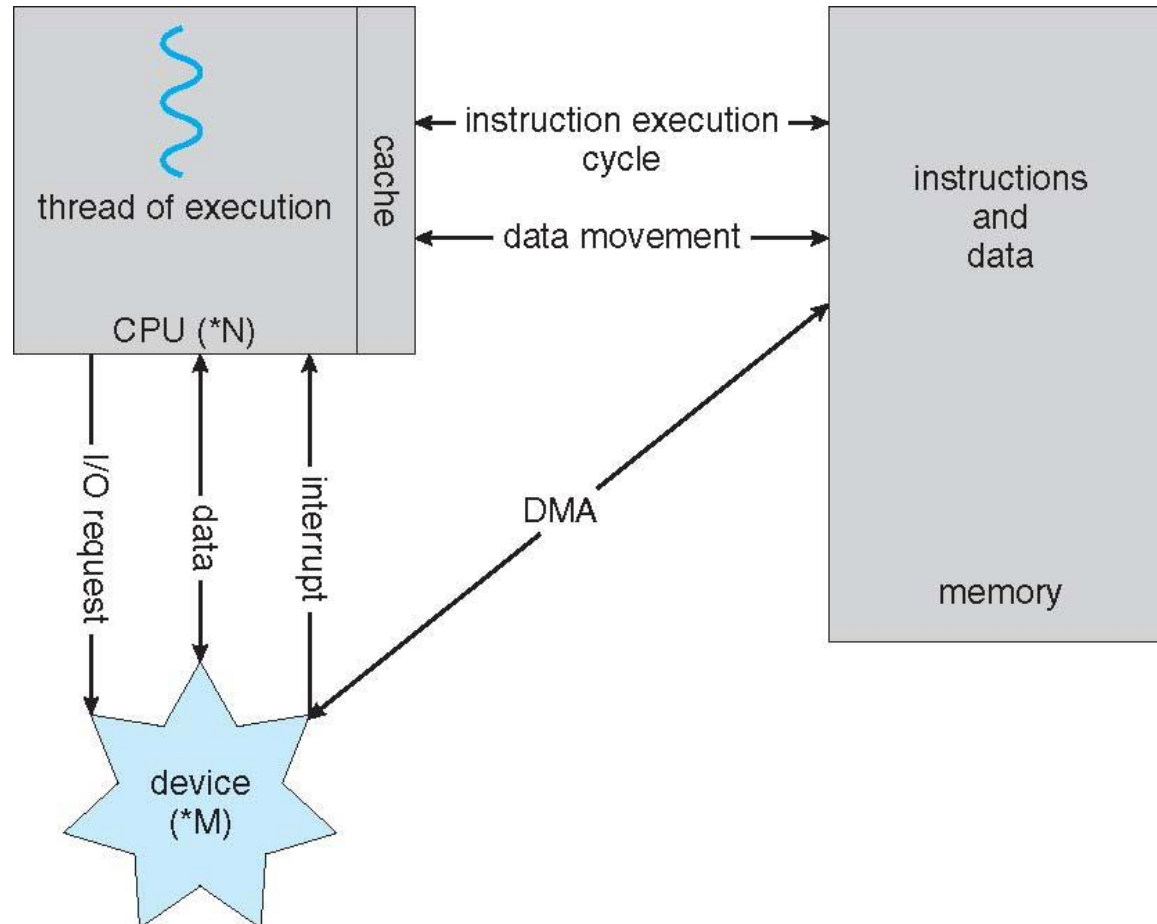
Multiprocessors systems growing in use and importance

- Also known as parallel systems, tightly-coupled systems
- Advantages include
 - ✓ Increased throughput
 - ✓ Economy of scale
 - ✓ Increased reliability – graceful degradation or fault tolerance
- Two types
 - ✓ Asymmetric Multiprocessing
 - ✓ Symmetric Multiprocessing



Introduction to Operating System

How a Modern Computer Works



Introduction to Operating System

Single-Processor System

- There is one main CPU capable of executing a general-purpose instruction set, including instructions from user processes.

Multiprocessor System

- Also known as parallel-system or multicore.
- First appeared in servers and now in smartphones and tablet computers.



Introduction to Operating System

Multiprocessor System

Advantages:

1. **Increased throughput.** Increasing the number of processor, expect to get more work done in less time.
2. **Economy of scale.** It can cost less than equivalent multiple single-processor systems because they can share peripherals, mass storage and power supplies.
3. **Increased reliability.** Functions can be distributed properly among several processors. If one processor fails, the other processor can pick-up the task.



Introduction to Operating System

Multiprocessor System

The multiprocessor systems in use today are of two types.

1. **Asymmetric multiprocessing.** In which each processor is assigned a specific task. A boss processor controls the system and the other processors either look to the boss for instructions or have predefined tasks. Boss-worker relationship.



Introduction to Operating System

Multiprocessor System

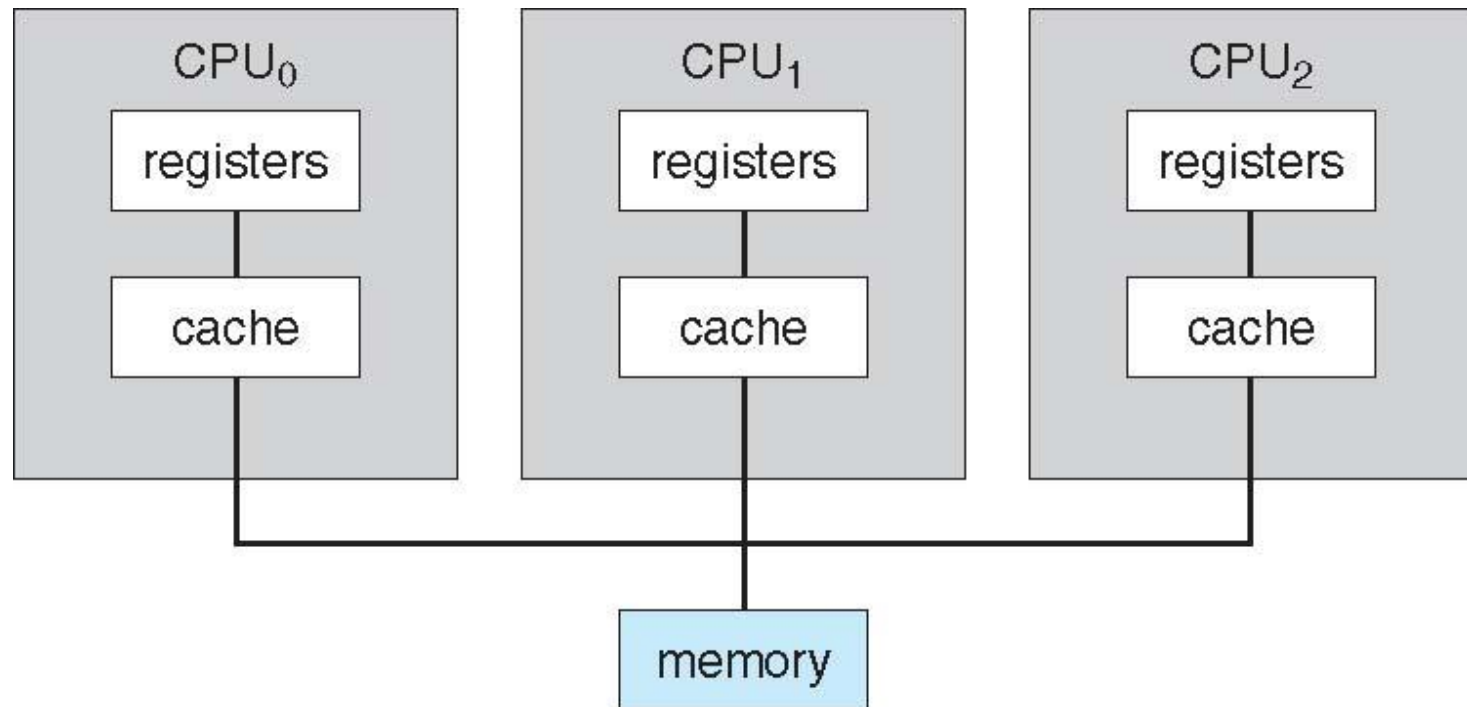
2. Symmetric multiprocessing(SMP). The most commonly used. In which each processor performs all tasks within the operating system. All processors are peers and no boss-worker relationship.

The difference between symmetric and asymmetric may result from either hardware or software.



Introduction to Operating System

Symmetric Multiprocessing Architecture



Introduction to Operating System

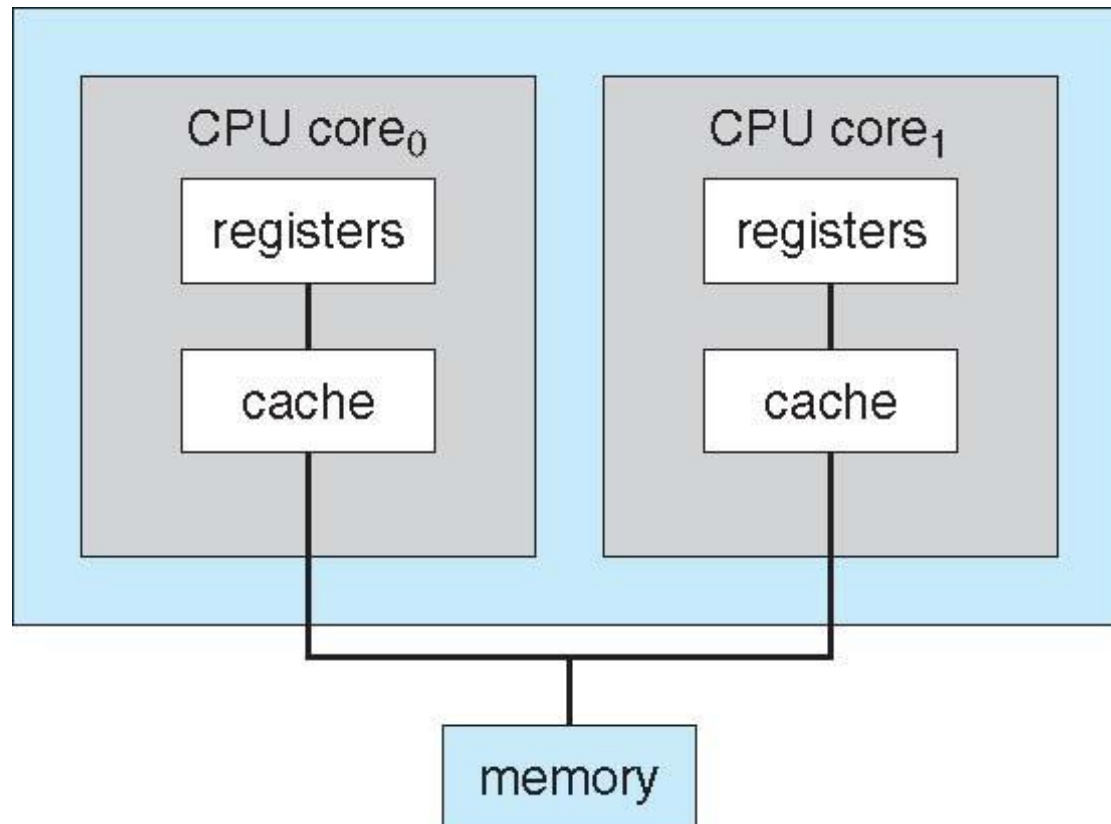
Multiprocessor System

- A recent trend in CPU design is to include multiple computing cores on a single chip. Such multiprocessor systems are termed multicore. They can be more efficient than multiple chips with single core.
- A dual-core design with two cores on the same chip. Each core has its own register set as well as its own local cache.



Introduction to Operating System

Dual-Core Design



Introduction to Operating System

Clustered Systems

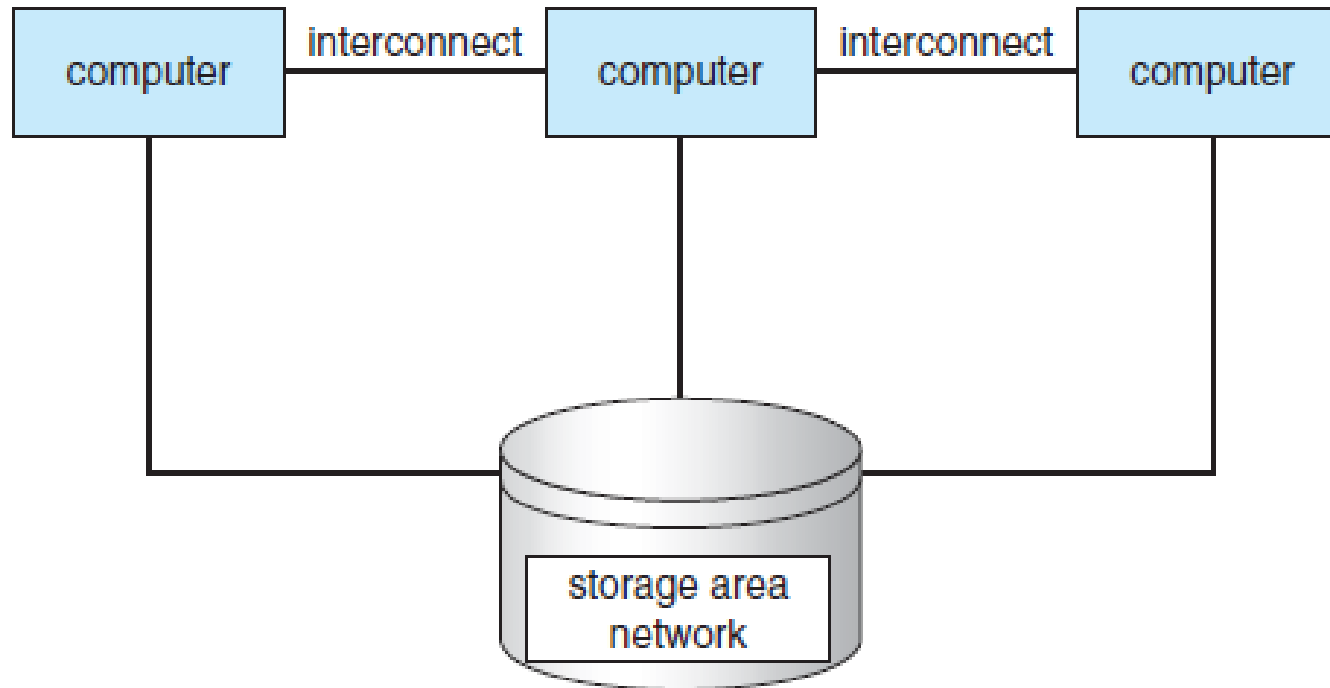
Like multiprocessor systems, but multiple systems working together

- 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
 - ✓ 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



Introduction to Operating System

Clustered System Structured



Introduction to Operating System

Operating System Structure

- An operating system provides the environment within which programs are executed.
- Most important aspect of operating system is the ability to multi-program.
- Multiprogramming increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.
- The operating system keeps several jobs in memory simultaneously. Since main memory is too small, jobs are kept initially on the disk in the job pool.



Introduction to Operating System

Multiprogramming System Memory Layout



Introduction to Operating System

Operating System Structure

- The set of jobs in memory can be a subset of the jobs kept in the job pool. The operating system picks and begins to execute one of the jobs in memory.
- Eventually, the job may have to wait for some task, such as an I/O operation, to complete. When that job needs to wait, the CPU switches to another job and so on.
- As long as at least one job needs to execute, the CPU is never idle.



Introduction to Operating System

Operating System Structure

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 \Rightarrow **process**
- If several jobs ready to run at the same time \Rightarrow **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



Introduction to Operating System

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.



Introduction to Operating System

Operating System Operations

- **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(1) or kernel code(0)
 - ✓ Some instructions designated as **privileged**, only executable in kernel mode
 - ✓ System call changes mode to kernel, return from call resets it to user



Introduction to Operating System

Operating System Operations

Transition from User to Kernel

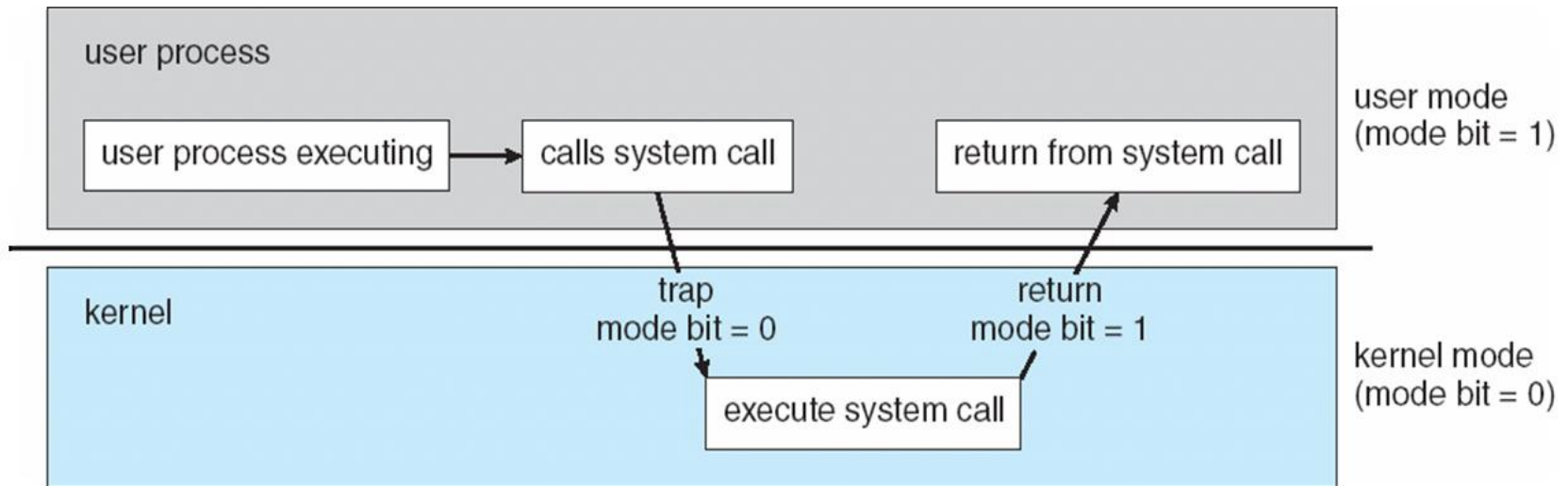
- Timer to prevent infinite loop/process hogging resources.
 - Set interrupt after specific period.
 - Operating system decrements counter.
 - When counter zero generate an interrupt.
 - Set up before scheduling process to regain control or terminate program that exceeds allotted time.



Introduction to Operating System

Operating System Operations

Transition from User to Kernel



Introduction to Operating System

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



Introduction to Operating System

Process Management

- 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



Introduction to Operating System

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



Introduction to Operating System

Memory Management

- The main memory is central to the operation of a modern computer system. It is a repository of quickly accessible data shared by the CPU and I/O devices.
- The central processor reads instructions from main memory during the instruction-fetch cycle and both reads and writes data from main memory.
- All data in memory before and after processing
- All instructions in memory in order to execute



Introduction to Operating System

Memory Management

- 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



Introduction to Operating System

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)



Introduction to Operating System

Storage Management

- 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



Introduction to Operating System

Mass-Storage Management

- Main memory is too small to accommodate all data and programs, the computer system must provide secondary storage to backup main memory.
- 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



Introduction to Operating System

Mass-Storage Management

- 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)



Introduction to Operating System

Caching

- Information is copied into a faster storage system, the cache on a temporary basis.
- Internal programmable registers, provide a high-speed cache for main memory.
- Other caches are implemented in hardware.
- Main memory can be viewed as a fast cache for secondary storage.
- Operating system may maintain a cache of file-system data in main memory.



Introduction to Operating System

Levels of Storage

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape



Introduction to Operating System

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



Introduction to Operating System

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



Introduction to Operating System

Protection and Security

- 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



Introduction to Operating System

Computing Environment

- 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



Introduction to Operating System

Computing Environment

- Mobile computing
 - Refers to computing on handheld smartphones and tablet computers.
- Distributed system
 - It is a collection of physically separate, possibly heterogeneous computer systems that are networked to provide users with access to the various resources that the system maintains.
 - Network operating system is an operating system that provides services across the network.



Introduction to Operating System

Computing Environment

- 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



Introduction to Operating System

Computing Environment

- 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*



Introduction to Operating System

Computing Environment

- Virtualization
 - It is a technology that allows operating systems to run as applications within other operating system.
 - It is one member of the class software that also includes emulation. Emulation is used when the source CPU type is different from the target CPU type.
 - Example: virtual machine



Introduction to Operating System

Cloud Computing

- It is a type of computing that delivers computing, storage and even applications as a service across a network.
- It is a logical extension of virtualization.
 - Public cloud – cloud available via the Internet
 - Private cloud – cloud run by a company for that company's own use
 - Hybrid cloud – cloud that includes both public and private
 - Software as a service(SaaS) – one or more applications available via the Internet
 - Platform as a service(PaaS) – software stack ready for application use via the Internet
 - Infrastructure as a service(IaaS) – servers or storage available over the Internet



Introduction to Operating System

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

