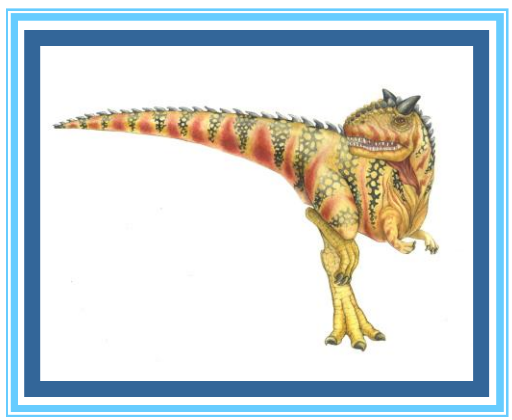
Chapter 1: Introduction



Chapter 1: Introduction

- What Operating Systems Do
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security

Objectives

- To describe the basic organization of computer systems
- To provide a grand tour of the major components of operating systems
- To give an overview of the many types of computing environments

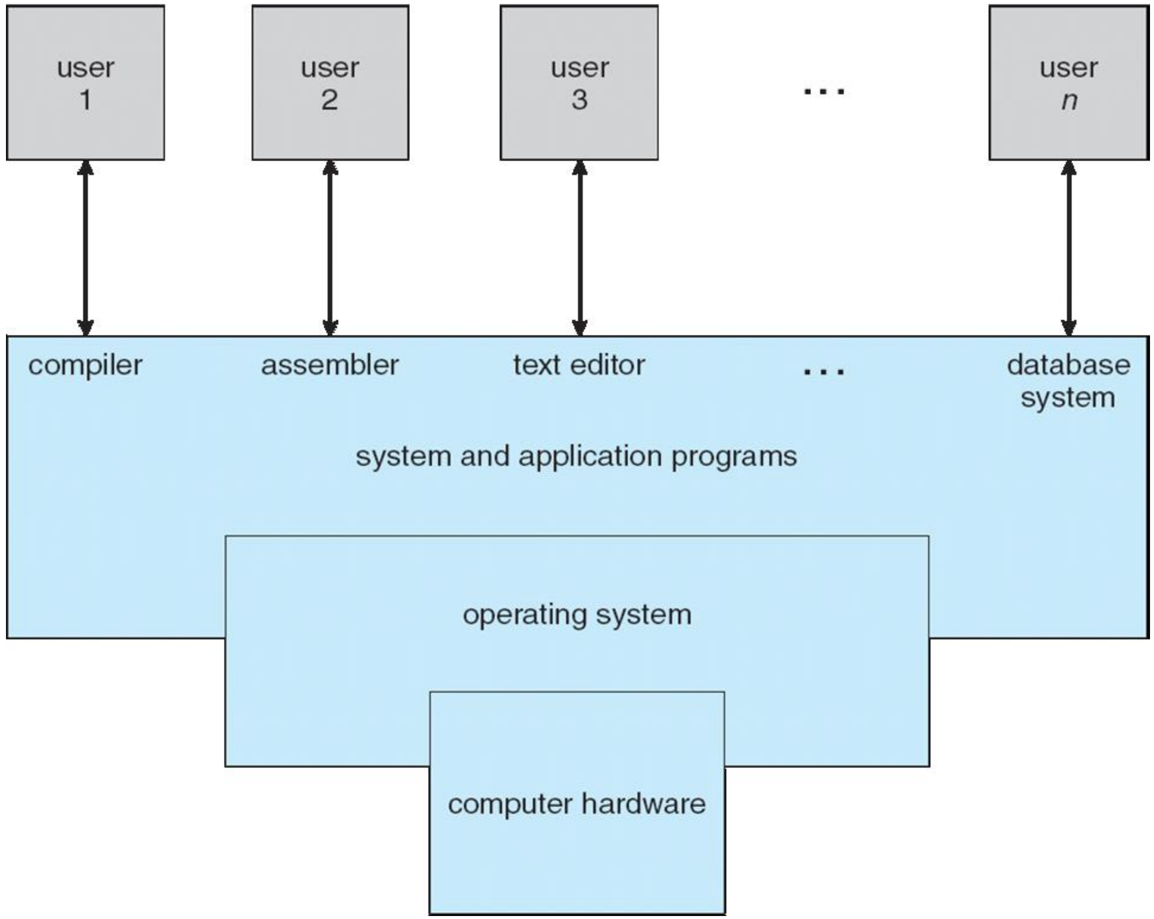
What is an Operating System?

- An operating system is a program that manages a computer’s hardware.
- A program that acts as an intermediary between a user of a computer and the computer hardware
- some operating systems are designed to be convenient, others to be efficient, and others to be some combination of the two.
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure

- Computer system can be divided into four components:
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users.
    - It hides the background details of hardware and makes it convenient for user to use it
  - 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
  - Users
    - People, machines, other computers

Four Components of a Computer System



What Operating Systems Do

- Depends on the point of view –**User view & System view**

User View:

- Users want convenience, **ease of use** and **good performance**
  - Don't care about **resource utilization,** the goal is to maximize the work done
- But in shared computer such as **mainframe** or **minicomputer** must keep all users happy, and maximize resource utilization.
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers,** thus here OS is designed between individual usability and resource utilization
- Also, in case of handheld computers are resource poor, optimized for usability and battery life
- Some computers have little or no user interface or view, such as embedded computers in devices and automobiles.

Operating System Definition

System View: OS is the program involved with hardware. Thus, we can view OS as

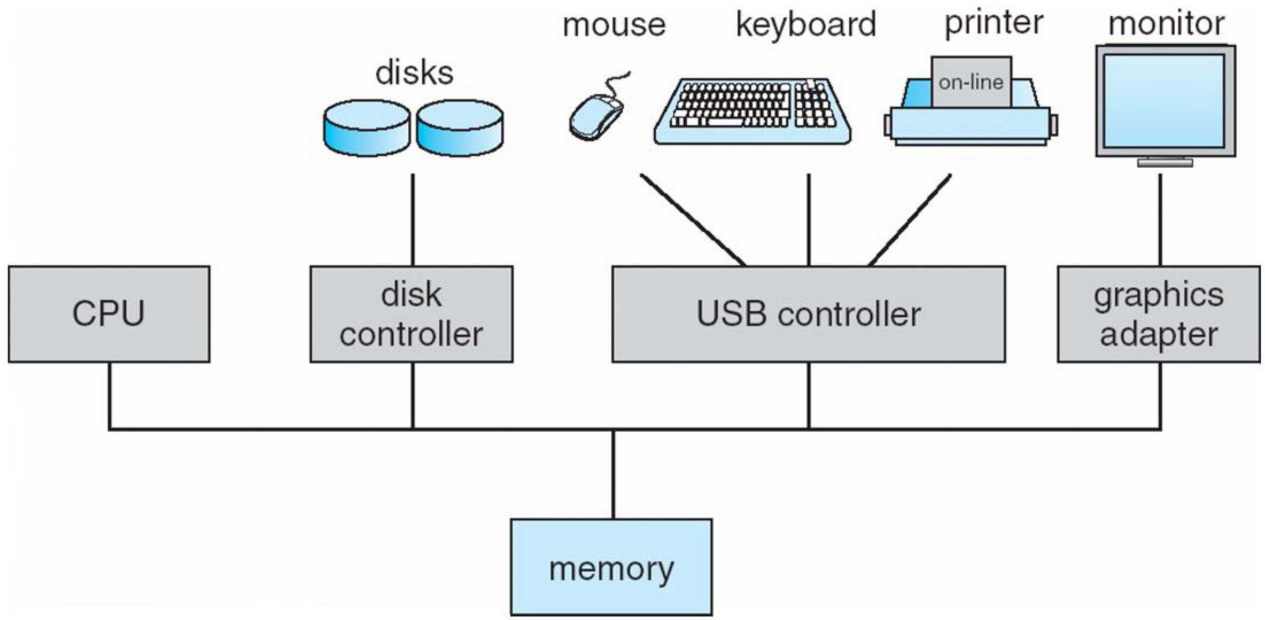
- OS is a **resource allocator**
  - Manages all resources (CPU, memory, I/O devices)
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls and manages the execution of programs to prevent errors and improper use of the computer
  - Concerned with operation and control of I/O devices

Operating System Definition (Cont.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is a good approximation
- “The one program running at all times on the computer” is the **kernel**.
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.

Computer System Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Each device controller is in charge of a particular device type
  - Concurrent execution of CPUs and devices competing for memory cycles



A modern computer system

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

Computer Startup

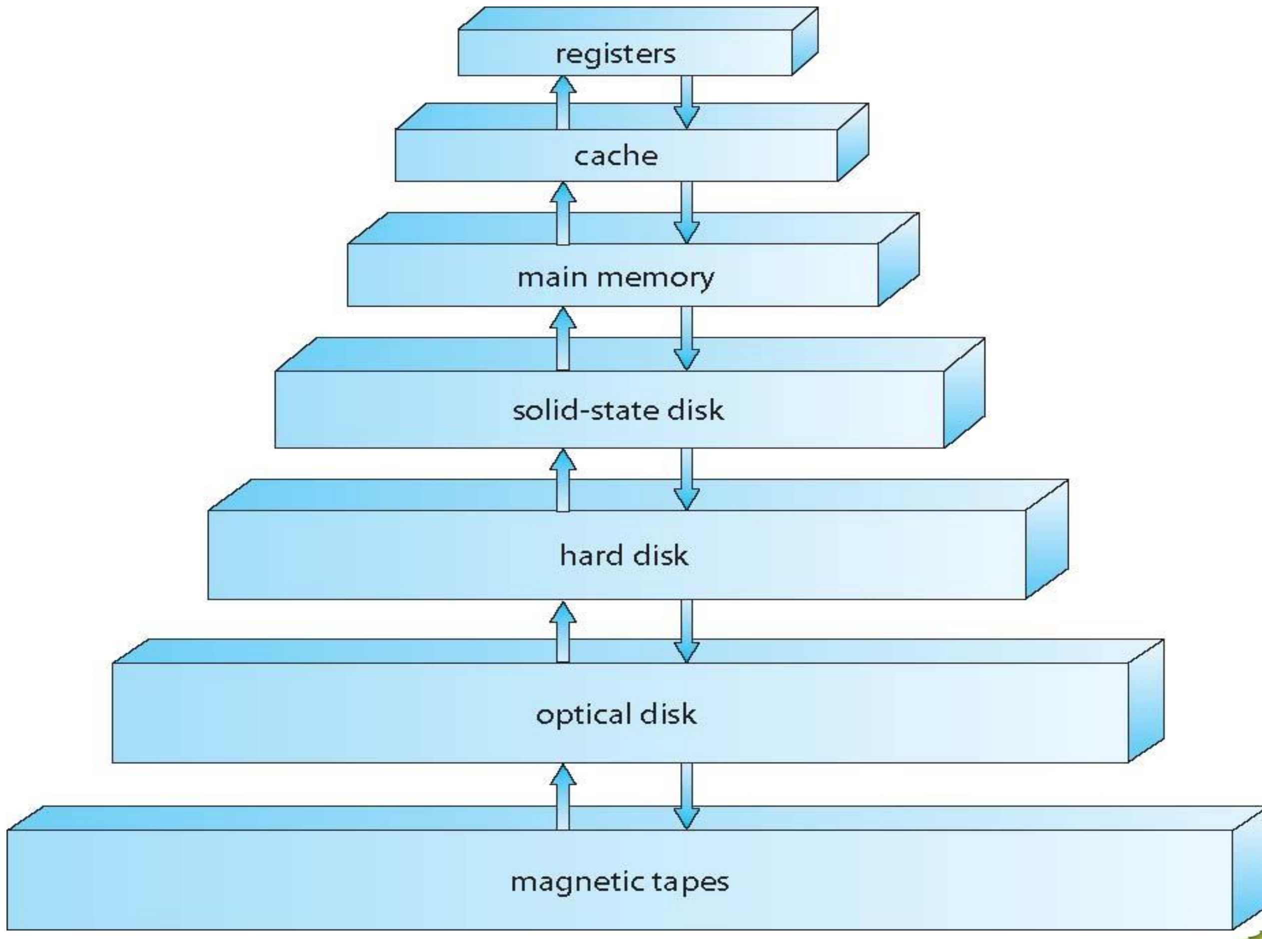
- **bootstrap program** is the initial program, loaded at power-up or reboot of a computer
  - Typically stored in **ROM or EPROM**, generally known as **firmware**
  - Initializes all aspects of system, from CPU register, device controller etc
  - It locates OS kernel and loads it into memory and starts execution
  - Once the kernel is loaded and executing, it can start providing services to the system and its users

Storage Structure

- Main memory – is the only large storage media that the CPU can access directly
  - **Random access**
  - **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard 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
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

Storage Hierarchy

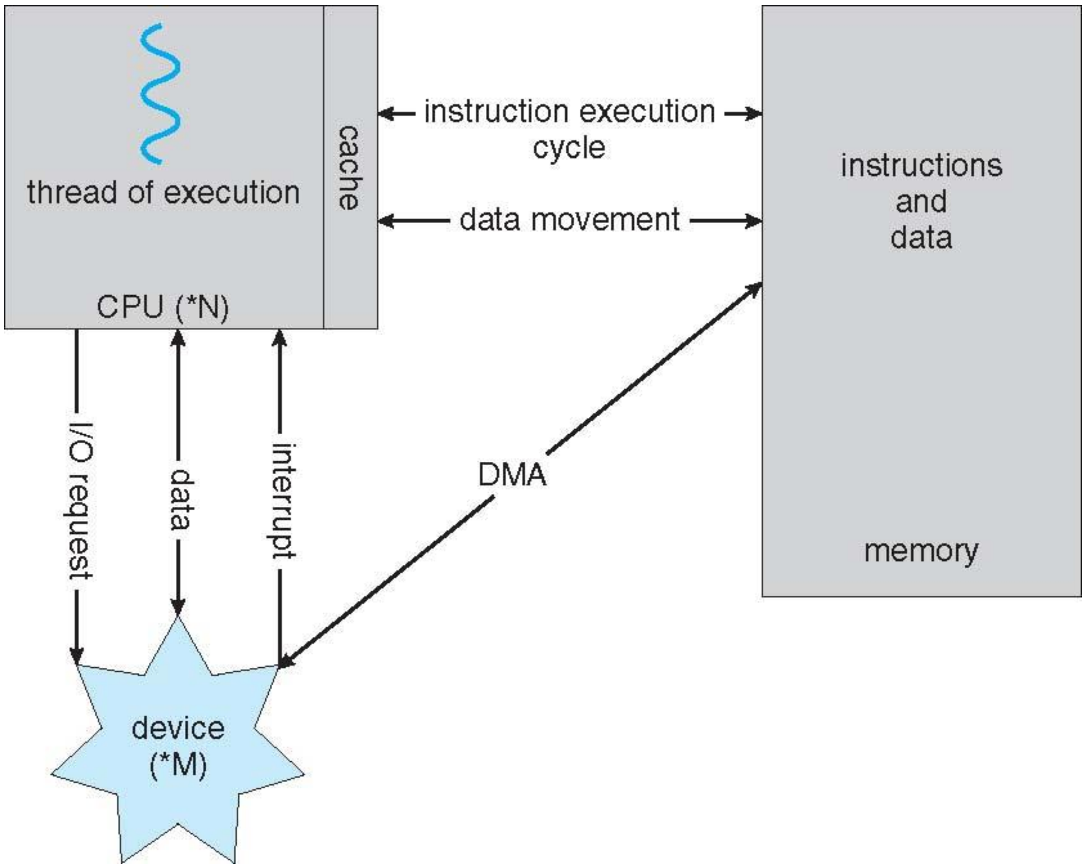
- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility



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 management important design problem
  - Cache size and replacement policy

How a Modern Computer Works



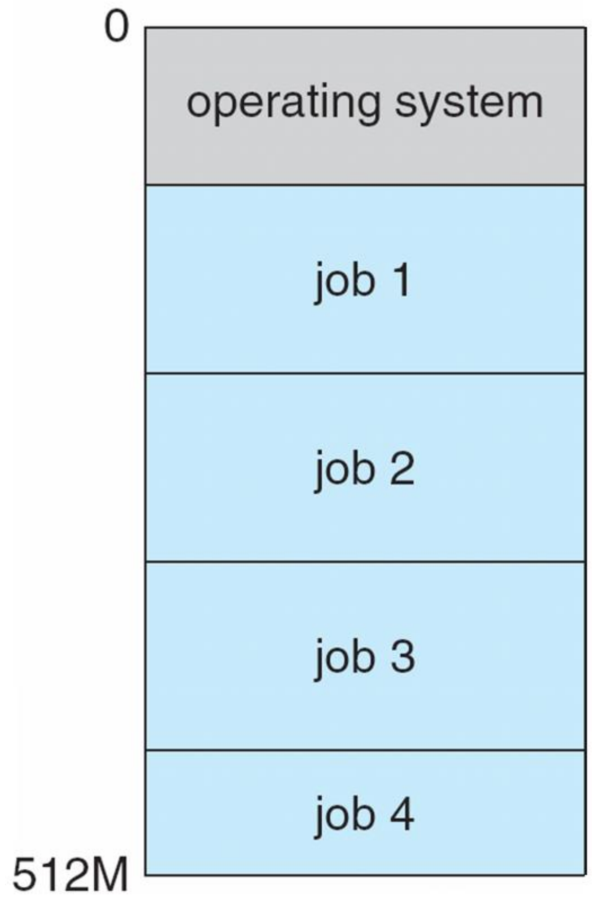
A von Neumann architecture

Operating System Structure

- **Multiprogramming** (**Batch system**) 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 of multiprogramming in which CPU executed multiple jobs and 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 of main memory to the disk to run and ensure reasonable response time.
  - **Virtual memory** allows execution of processes not completely in memory

  to support swapping and ensure reasonable response time.

Memory Layout for Multiprogrammed System



Operating-System Operations

- **Operating Systems are Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap) caused by**
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system
- For each type of interrupt, separate segments of code in the operating system determine what action should be taken.
- An interrupt service routine is provided to deal with the interrupt
- A OS must be properly designed to handle interrupts properly and ensure that an incorrect program does not affect execution of other programs.
- Thus, most computer systems provide hardware support that allows us to differentiate among various **modes of execution**.

Operating-System Operations (cont.)

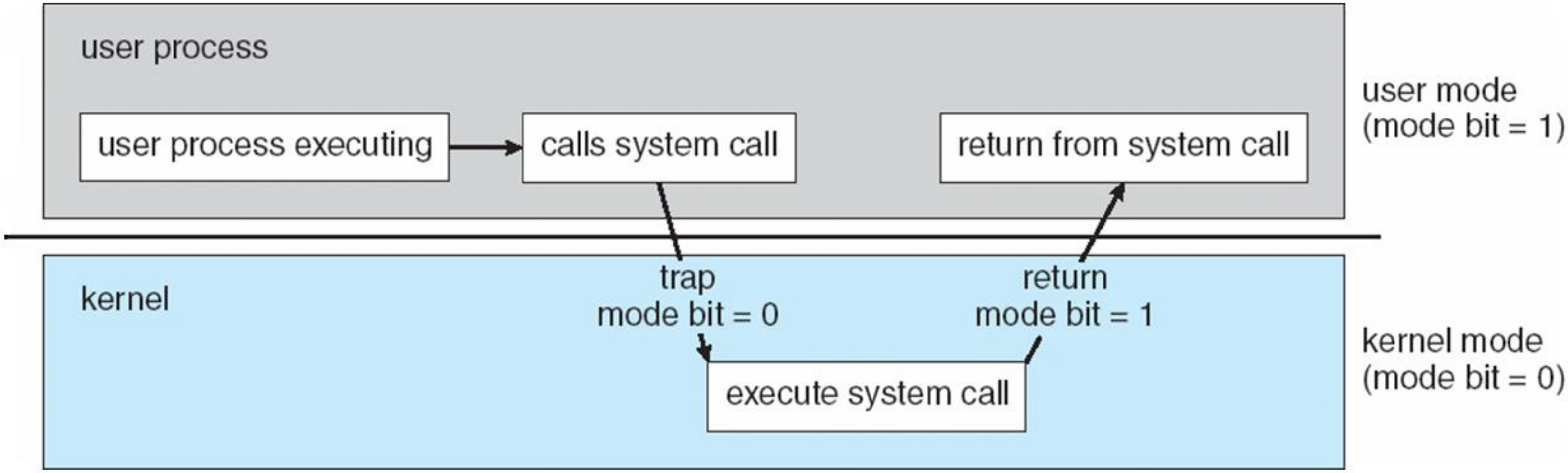
Two modes of operations:

- **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 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
- The concept of modes can be extended beyond two modes when

increasingly CPUs support virtualization frequently. Thus, separate mode to indicate

  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode



Transition from user to kernel mode.

Timer

We must ensure that OS maintains control over the CPU and prevent user program from not letting control back to OS. Thus,

- Timer prevents infinite loop / process hogging resources and never return control to the operating system.
  - Timer can be set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter, every time the clock ticks, the counter is decremented
  - When counter zero generate an interrupt
  - OS ensure to set up the timer before scheduling process to regain control or terminate program that exceeds allotted time

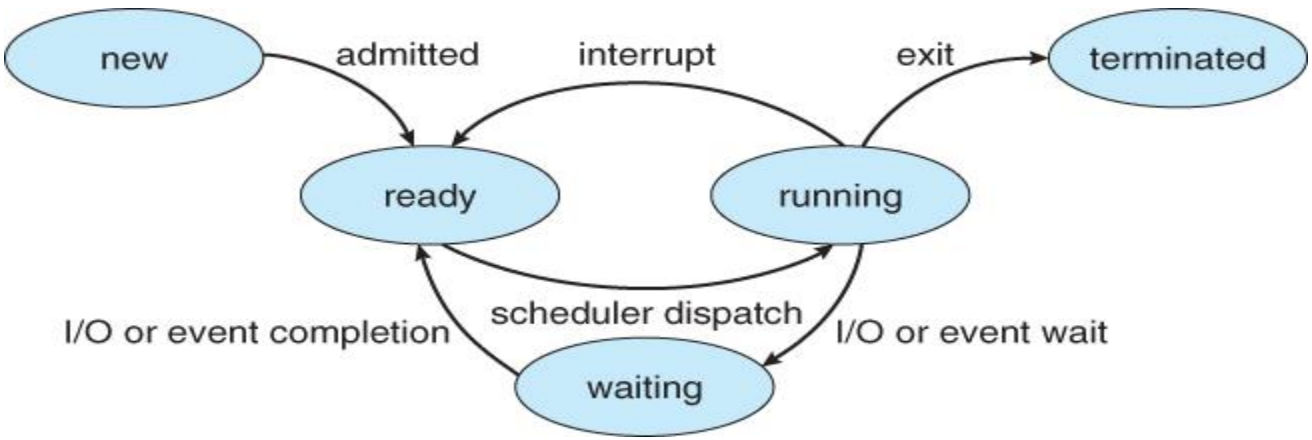
Process Management

- 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

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and 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 is one of the visible component of OS
  - 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 like, Tertiary storage includes optical storage, magnetic tape, CD and DVD. Tertiary storage is not crucial to system performance, but
  - Still must be managed – by OS or applications

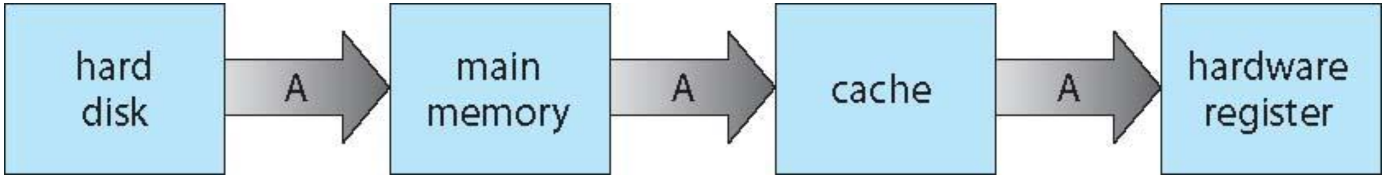
Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

Migration of data “A” 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

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

Only the device drives know the peculiarities of the device assigned to it.

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

End of Chapter 1