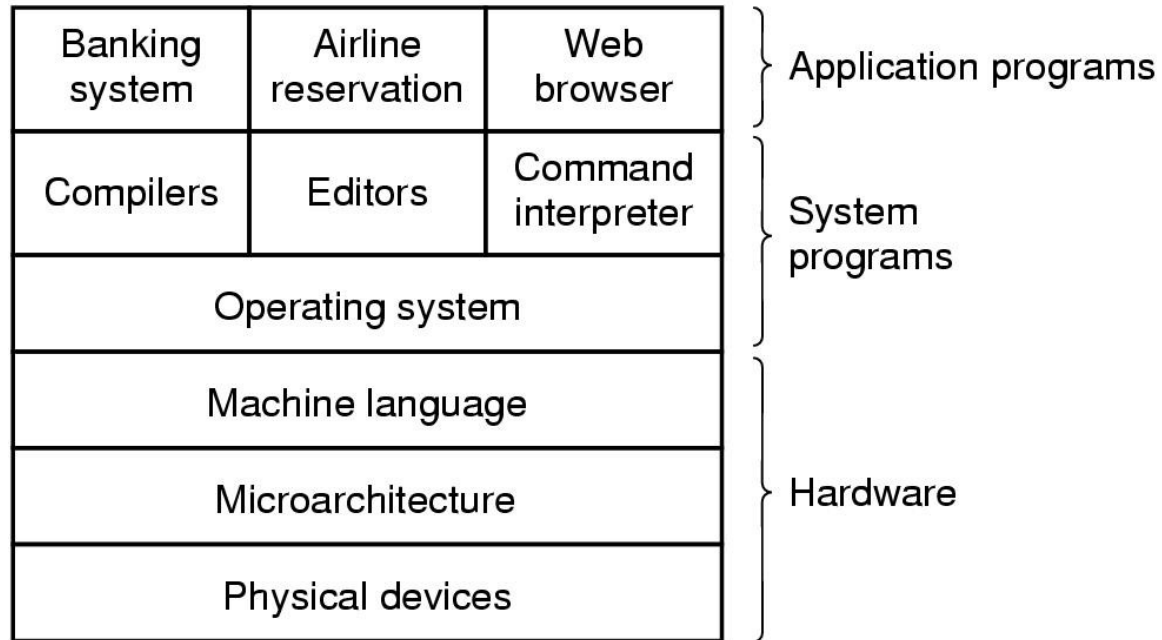# Distributed Operating Systems

Introduction to Operating Systems

Topics

- Computer Systems

- Operating System Definition

- Multiprogramming and time sharing

- Interrupts

- Dual mode operation

- Process, Memory, Storage management

Components of a Computer System

Banking system	Airline reservation	Web browser	} Application programs
Compilers	Editors	Command interpreter	} System programs
Operating system			
Machine language			} Hardware
Microarchitecture			
Physical devices			

- A computer system consists of
  - hardware
  - system programs
  - application programs

What is an Operating System?

- 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

4

What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware

- 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

Operating System Definition

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use

- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

Operating System Structure (1)

- **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

Operating System Structure (2)

- **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

Multiprogramming (1)
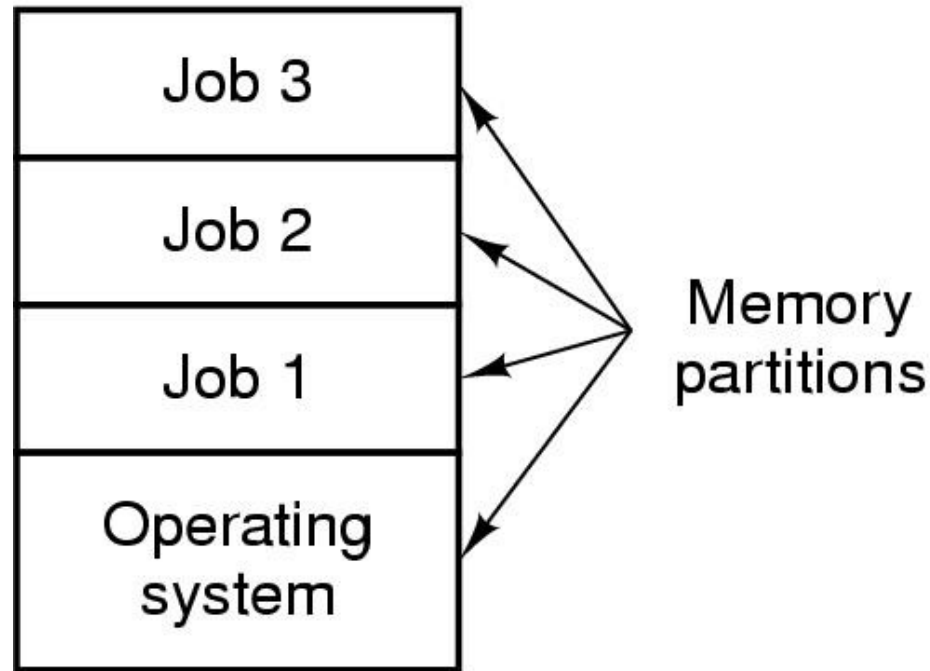
- Multiprogramming system
  - Three jobs in memory
 - Operating System

Multiprogramming (2)

Address
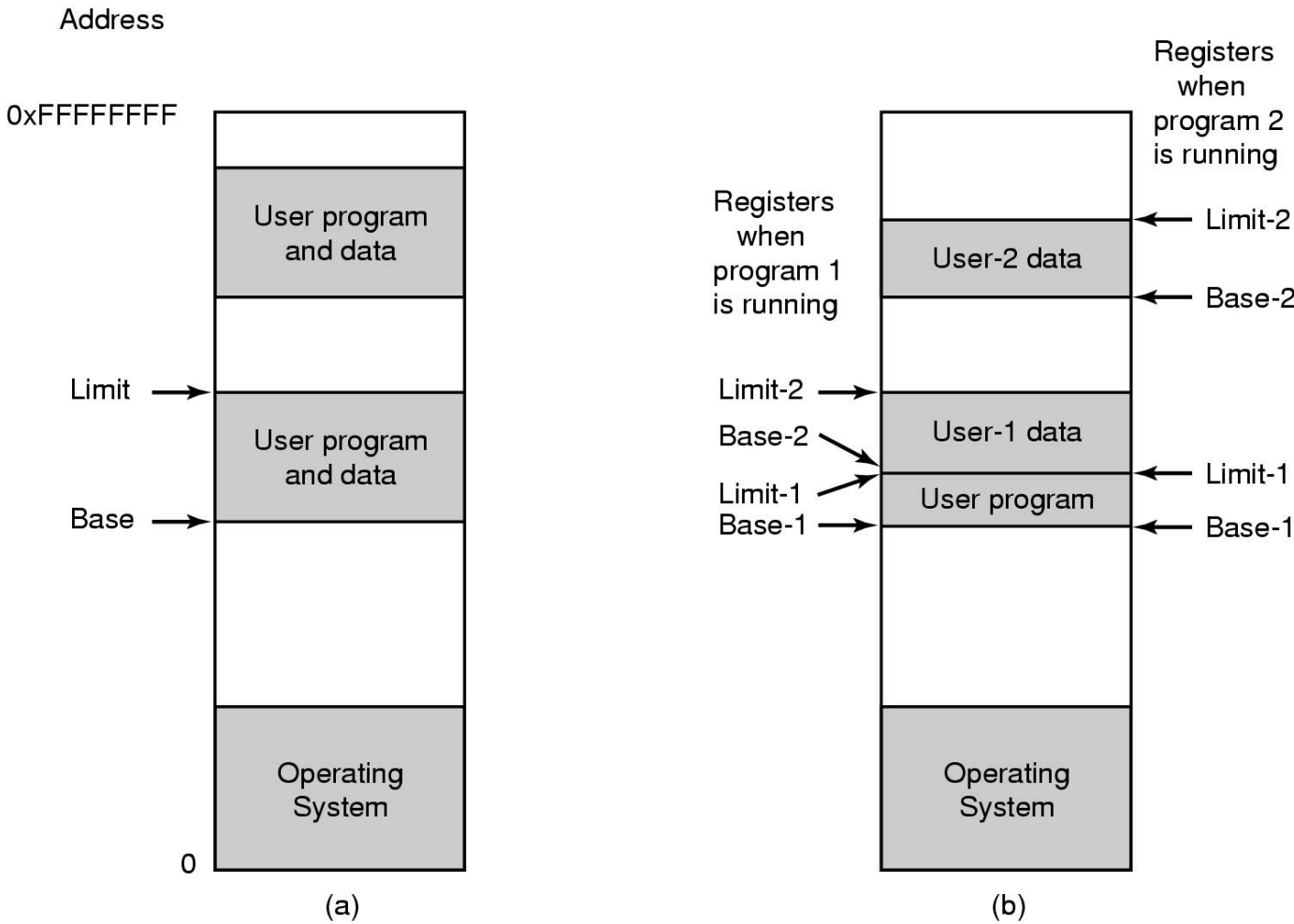
One base-limit pair and two base-limit pairs

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

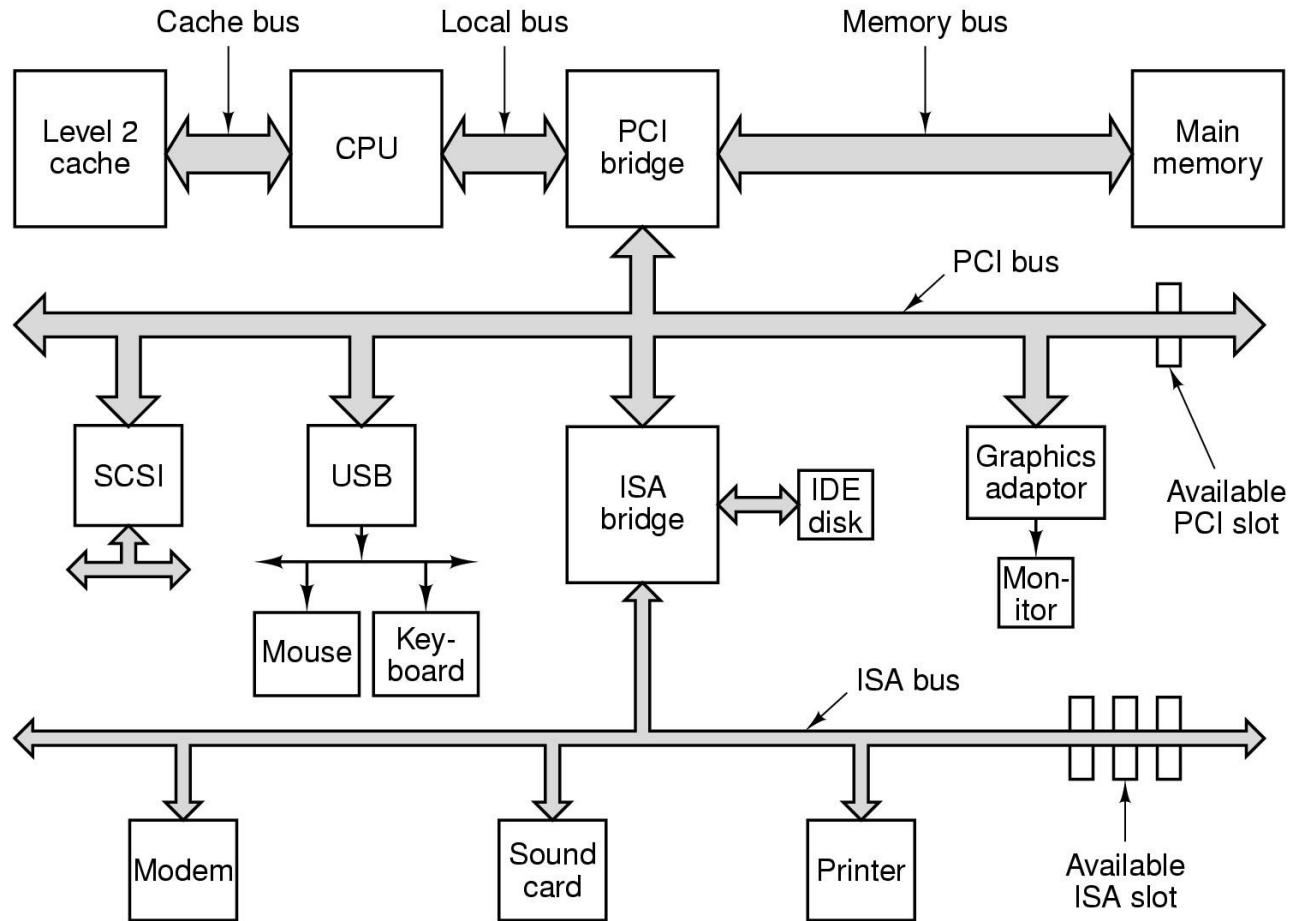# Computer Hardware Review (1)
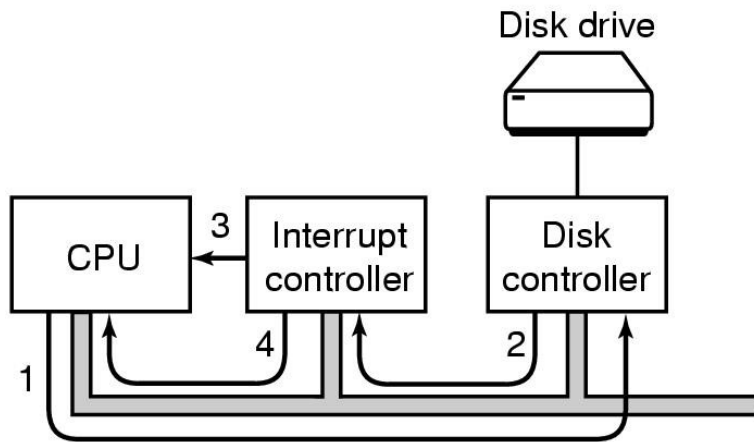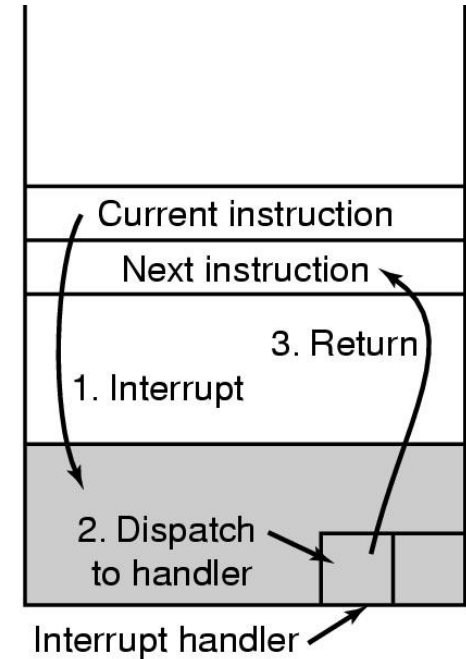
(a)
(b)

(a) Steps in starting an I/O device and getting interrupt
(b) How the CPU is interrupted

Computer Hardware Review (2)



Structure of a large Pentium system

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

- Operating systems are **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 (Interrupt Service Routine (ISR))

Interrupt Timeline
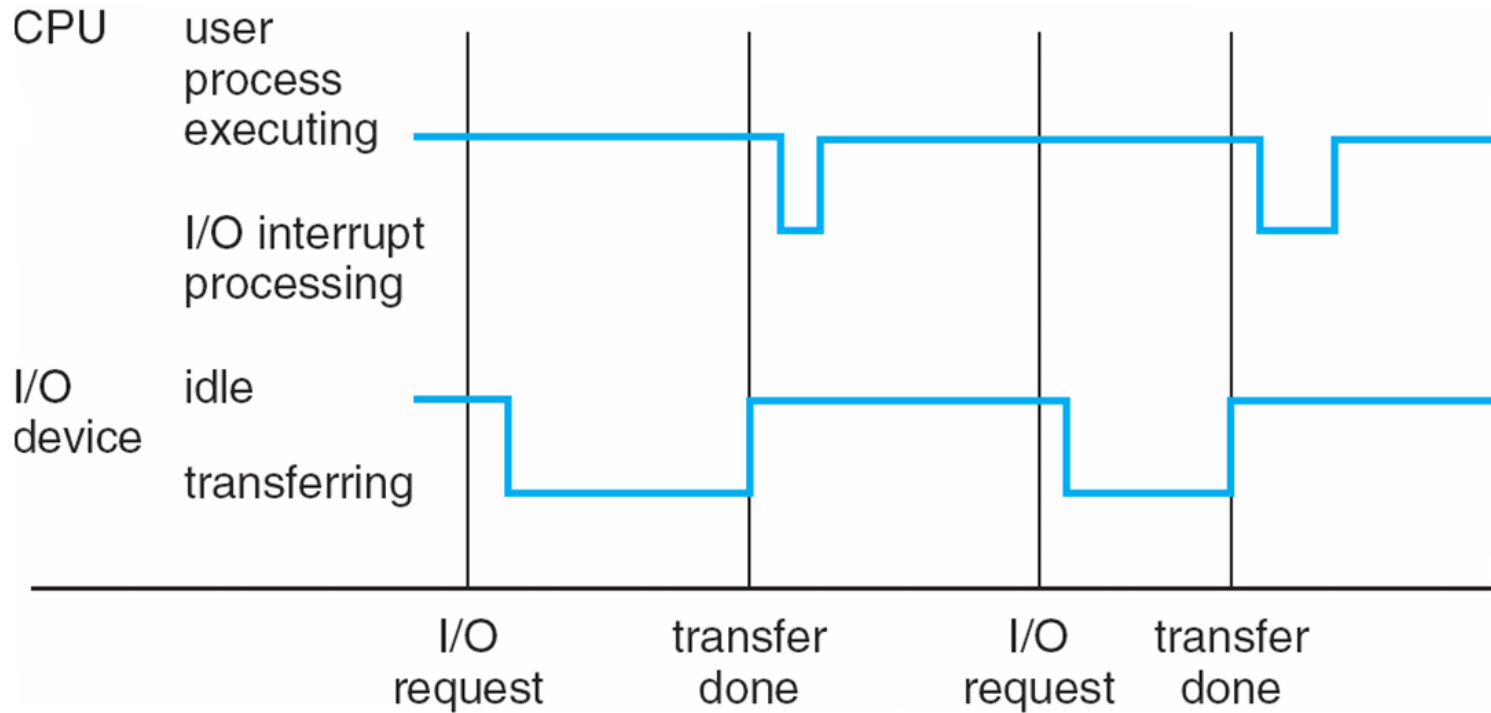
I/O Structure

- After I/O starts, control returns to user programs without waiting for I/O completion
  - **System call** – User program sends a request to the operating system to perform I/O operation
  - **Context Switching** – Operating system switches to another user program while the first one is waiting for I/O operation to complete

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

Dual Mode Operation

- **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

Transition from User to Kernel Mode
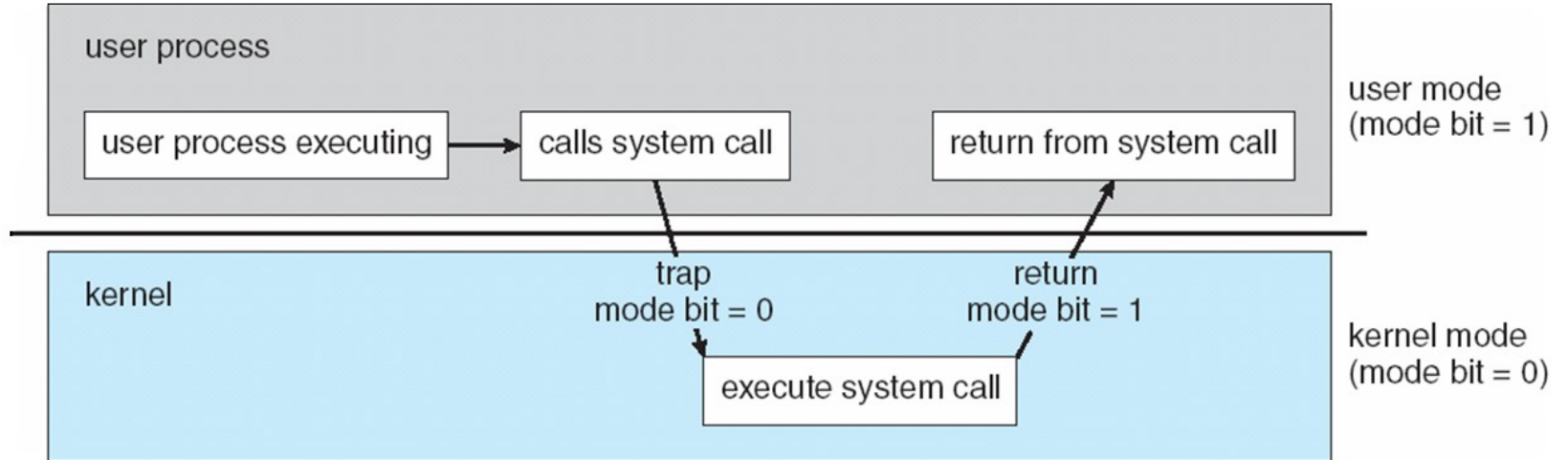
Operating System Functions

- Process Management

- Memory Management

- Storage Management

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 in 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

Storage Hierarchy

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

- **Caching** – copying information into faster storage system; main memory can be viewed as a *cache* for secondary storage

Storage-Device Hierarchy
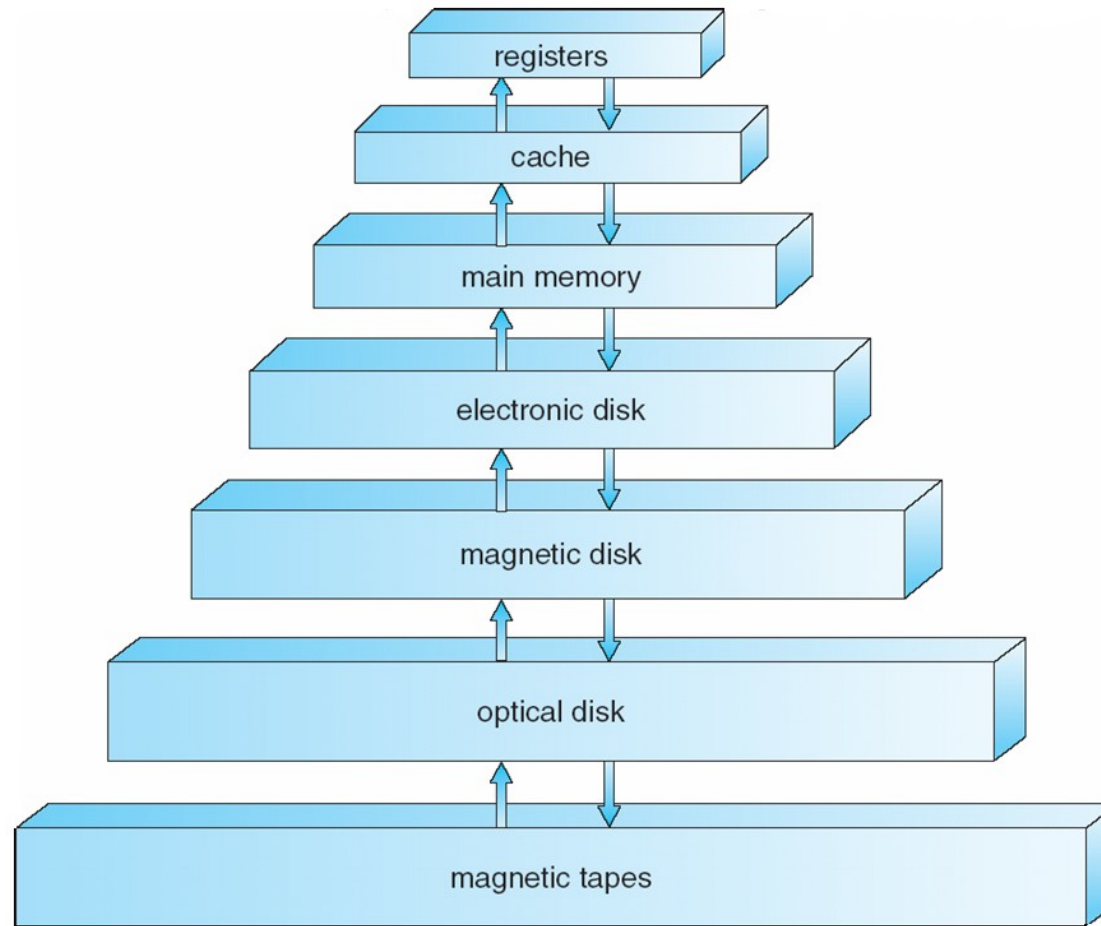
Mass-Storage Management

- 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 – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

Questions?