Memory Management,
Process Management
Storage Management,
Mass-Storage Management

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

Process Management

- Process and Program:
  - A **process** is a program in execution (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 (received upon creation and along execution)
  - Initialization data (e.g., a process for presenting the status of a file)
- 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

Storage Management

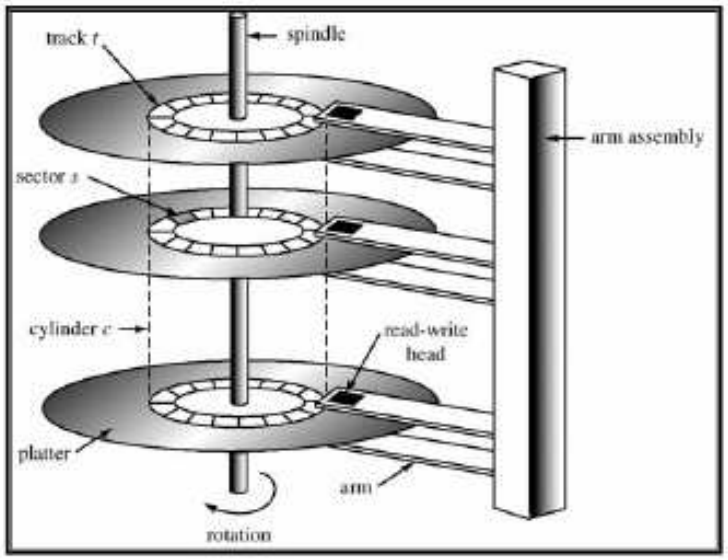
- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential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

Mass-Storage Management

- Main memory – only large storage media that the CPU can access directly
- Why using disks?
  - Store data that does not fit in main memory
  - Store 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

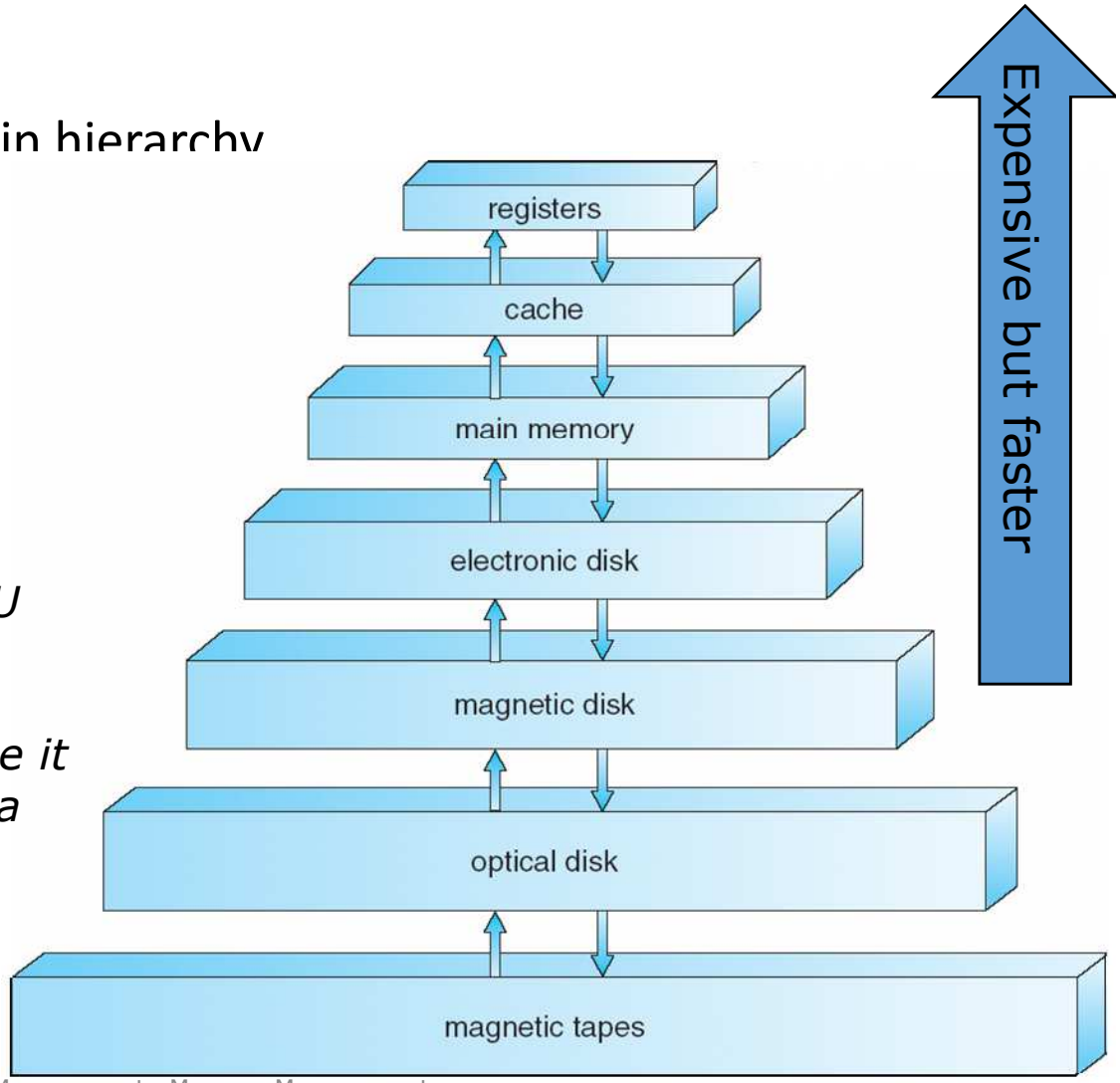
Storage Structure

- Secondary storage:
  - Extension of main memory
  - 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



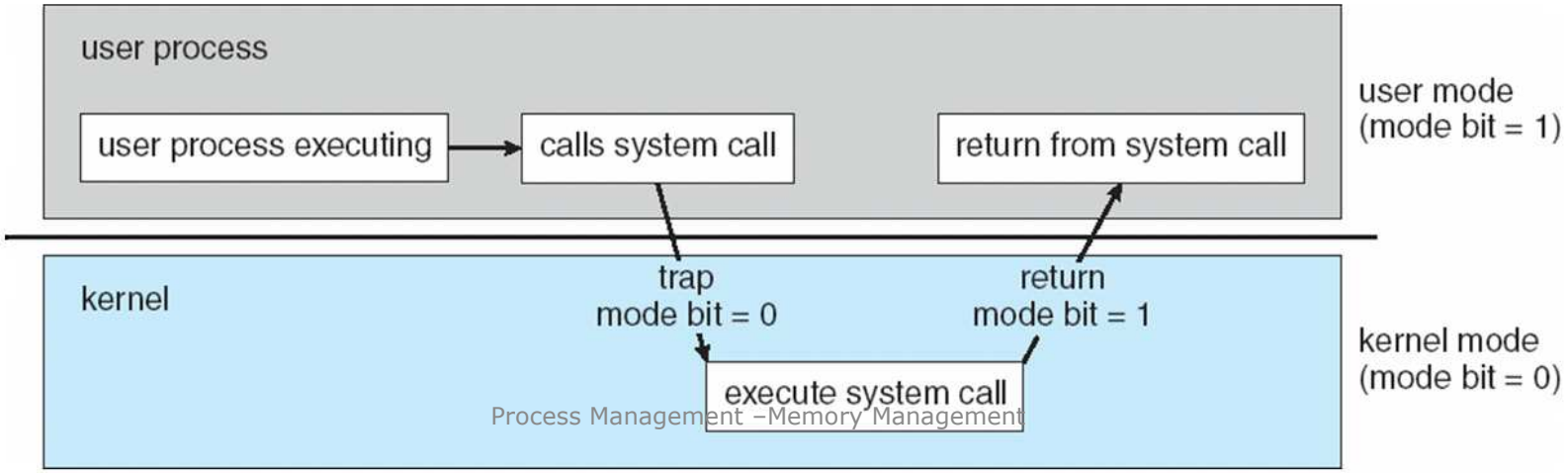
It takes some time (several CPU cycles) to read/write to main memory – in the meantime the processor needs to stall because it doesn't have the necessary data

Operating-System Operations

- Handle flow problems:
  - 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
  - Example: in MS-Dos, originally written for Intel 8088:
    - a user program can wipe out the operating system by writing over it with data

Transition from User to Kernel Mode

- **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



Example

- Which of the following instructions should be privileged?
  - b. Read the clock.
  - c. Clear memory.
  - d. Issue a trap instruction.
  - e. Turn off interrupts.
  - f. Modify entries in device-status table.
  - g. Switch from user to kernel mode.
  - h. Access I/O device.

Answer

- Which of the following instructions should be privileged?
  - b. Read the clock.
  - c. Clear memory.
  - d. Issue a trap instruction.
  - e. Turn off interrupts.
  - f. Modify entries in device-status table.
  - g. Switch from user to kernel mode.
  - h. Access I/O device.

Process Management

- Process and Program:
  - A **process** is a program in execution (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 (received upon creation and along execution)
  - Initialization data (e.g., a process for presenting the status of a file)
- 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:

- 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