OPERATING SYSTEM

text book: operating system concepts by Abraham silberschatz, peter b galvin and greg gagne, 8th edition

CHAPTER 1 : Introduction

What is operating system?

- An operating system is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.

Why operating system? Or 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.

Examples of os :

Windows : programming language?

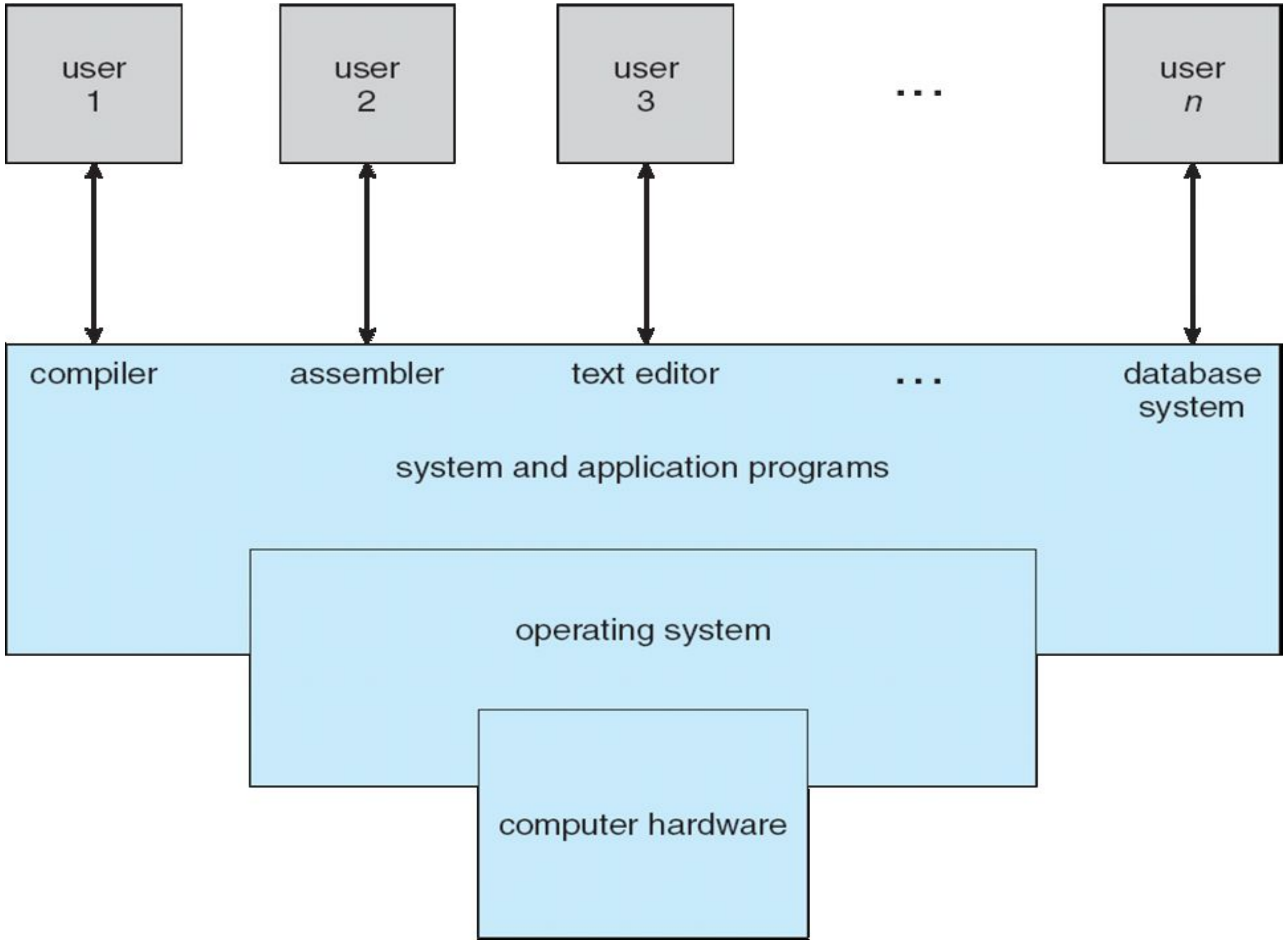
Linux:

Android :

ios:

1.1 What Operating Systems Do

abstract view of the components of a computer system



Continued..

- 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
  - 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.
- We can also view a computer system as consisting of hardware, software and data.
- **An operating system is similar to a *government*.** Like a government, it performs no useful function by itself. It simply provides an *environment* within which other programs can do useful work.

1.1.1 User view

- The user's view of the computer varies according to the interface being used.
- **The goal is to maximize the work (or play) that the user is performing.** In this case/ the operating system is designed mostly for ease of use with some attention paid to performance and none paid to resource utilization –how various hardware and software resources are shared.
- **In other cases, a user sits at a terminal connected to a mainframe or a minicomputer.** Other users are accessing the same computer through other terminals. These users share resources and may exchange information. **The operating system in such cases is designed to maximize resource utilization**.
- **In still other cases, users sit at workstations connected to networks of other workstations and servers. These users have dedicated resources at their disposal, but they also share resources such as networking and servers.** Therefore, their operating system is designed to compromise between individual usability and resource utilization.
- **Some computers have little or no user view**. For example, embedded computers in home devices.

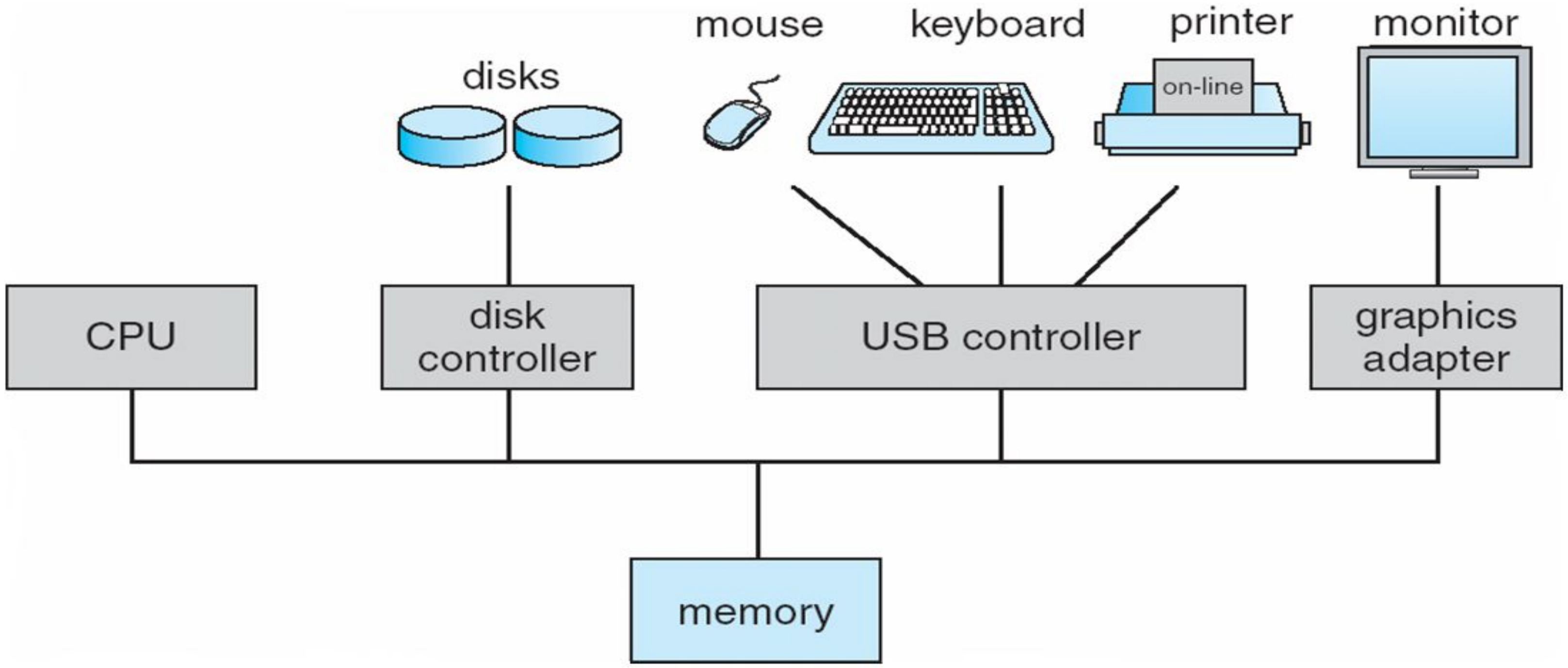
1.1.2 System view

- From the computer's point of view, the operating system is the program most intimately involved with the hardware. **In this context, we can view an operating system as a resource allocator**.
- **A computer system has many resources that may be required to solve a problem: CPU time, memory space and so on. The operating system acts as the manager of these resources.**
- **the operating system must decide how to allocate them to specific programs and users so that they are used efficiently.**
- A slightly different view of an operating system emphasizes the need to control the various I/0 devices and user programs.
- **A control program manages the execution of user programs to prevent errors and improper use of the computer.**

1.2 Computer System Organization

1.2.1 computer system operation

- A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory (Figure 1.2).



Continued…

- Each device controller is in charge of a specific type of device (for example, disk drives, audio devices, and video displays).
- The CPU and the device controllers can execute concurrently, competing for memory cycles.
- To ensure orderly access to the shared memory, a memory controller is provided whose function is to synchronize access to the memory.
- For a computer to start running-for instance, when it is powered up or rebooted-it needs to have an initial program to run. **This initial program is called bootstrap program.**
- **Typically, it is stored in read-only memory(ROM) or electrically erasable programmable read-only memory(EEPROM) known by the general term within the computer hardware.**
- It initializes all aspects of the system, from CPU registers to device controllers to memory contents.
- **The bootstrap program must know how to load the operating system and how to start executing that system.** To accomplish this goal, the bootstrap program must locate and load into memory the operating system **kernel**. The operating system then starts executing the first process, such **as "init,"** and waits for some event to occur.

Continued…

- **The occurrence of an event is usually signaled by an interrupt from either the hardware or the software.**
- **may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the Hardware system bus.**
- **Software may trigger an interrupt executing a special operation called system call.**
- **When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location usually contains the starting address where the service routine for the interrupt is located.**

1.2.2 storage structure

- The CPU can load instructions only from memory, so any programs to run must be stored there. General-purpose computers run most of their programs from rewriteable memory, called main memory.
- Main commonly is implemented in a semiconductor technology called **Dynamic random access memory(DRAM).**
- **All forms of memory provide an array of words. Each word has its own address. Interaction is achieved through a sequence of load or store instructions to specific memory addresses.**
- **The load instruction moves a word from main memory to an internal register within the CPU, whereas the store instruction moves the content of a register to main memory.**

Continued..

Ideally, we want the programs and data to reside in main memory permanently. This arrangement usually is not possible for the following two reasons:

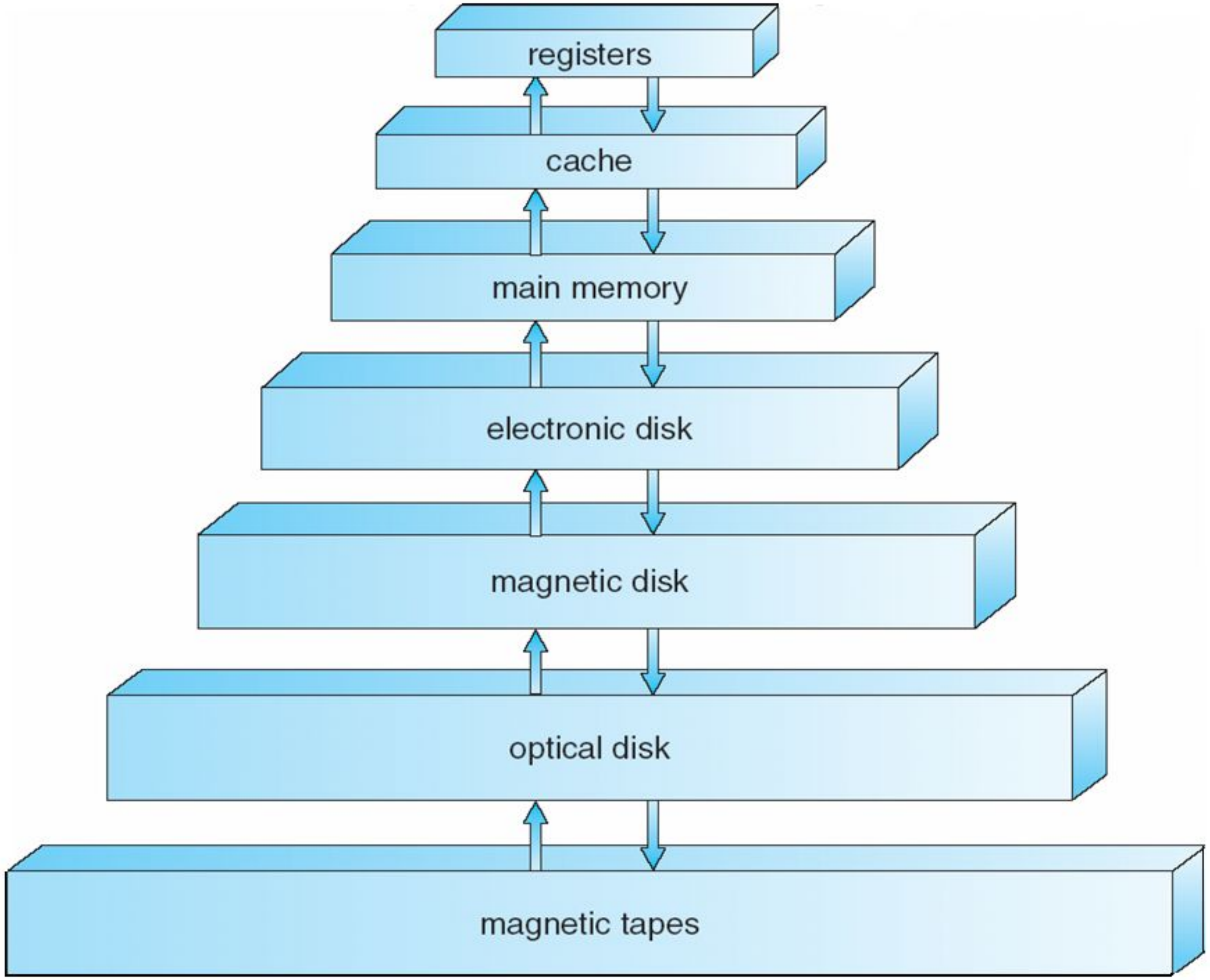
1. Main memory is usually too small to store all needed programs and data permanently.

2. Main memory is a *volatile* storage device that loses its contents when power is turned off or otherwise lost.

Thus, most computer systems provide as an extension of main memory. The main requirement for secondary storage is that it be able to hold large quantities of data permanently.

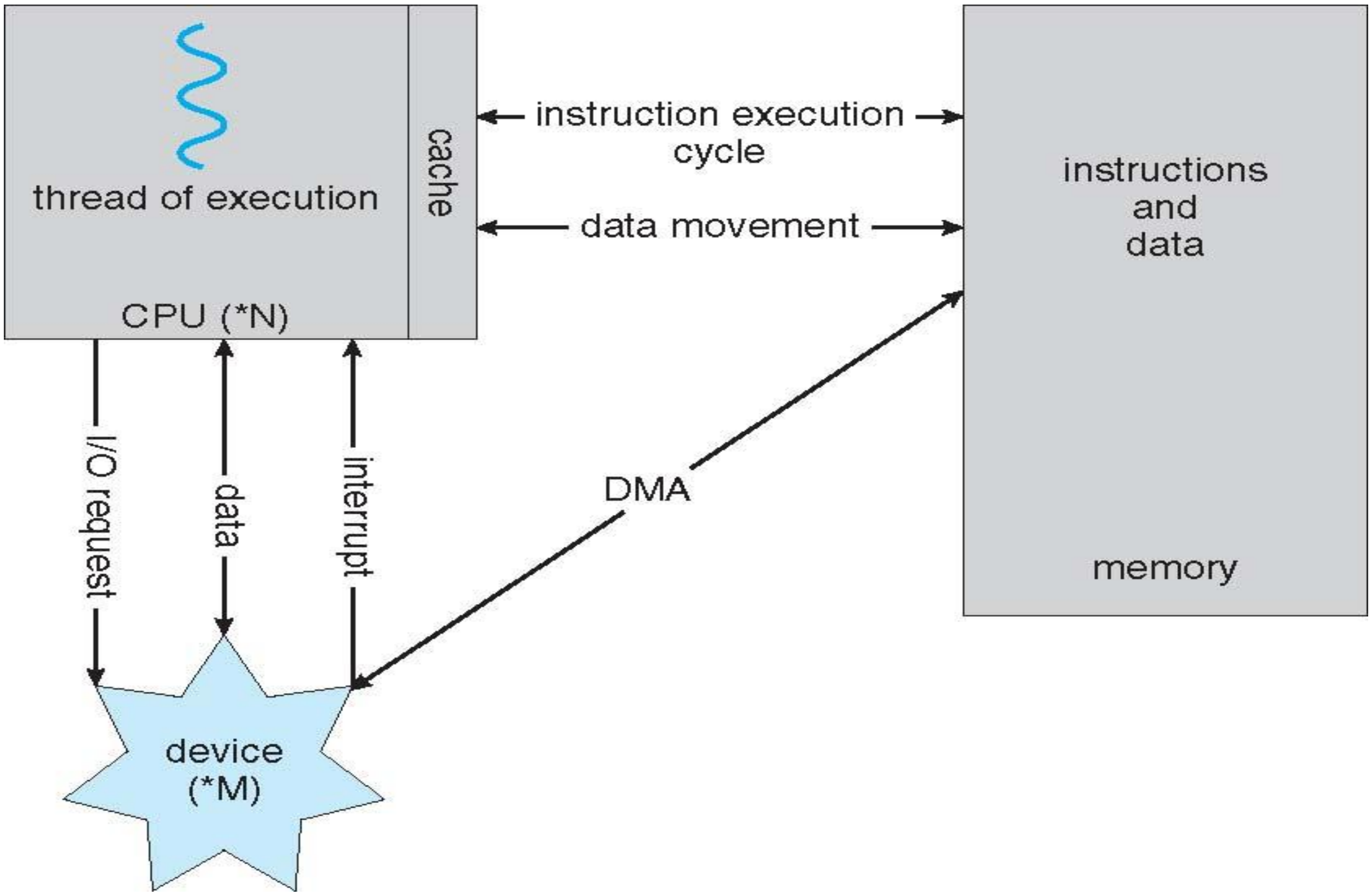
Continued…

- **The wide variety of storage systems in a computer system can be organized in a hierarchy (Figure 1.4) according to speed and cost. The higher levels are expensive, but they are fast. As we move down the hierarchy, the cost per bit generally decreases, whereas the access time generally increases.**
- **A volatile storage loses its contents when the power to the device is removed.**
- **Non volatile storage does not loose its contents when power is turned off.**
- In the hierarchy shown in Figure 1.4, the storage systems above the electronic disk are volatile, whereas those below are non volatile.
- Semiconductor memory have become faster and cheaper. The top four levels of memory in Figure 1.4 may be constructed using semiconductor memory.


registers
cache
main memory
electronic disk
magnetic disk
optical disk
magnetic tapes

1.2.3 I/O structure

- **A general purpose computer system consist of CPUs and multiple device controllers that are connected through a common bus.**
- **Each device controller is in charge of specific type of device. A device controller maintains some local buffer storage and set of special purpose registers.**
- **Each device controller is responsible for moving data between devices it controls and local buffer**.
- **To start I/O operation device driver loads appropriate registers within device controller. The device controller in turn examines registers to determine what action to take.**
- **The controller starts transfer of data from device to its local buffer. Once transfer of data is complete it informs device driver via interrupt. Then device driver returns control to operating system.**
- **This form of interrupt driven I/O is fine for moving small amounts of data but can produce high overhead. To solve this problem direct access memory(DMA) is used.**
- **In DMA device controller transfers entire block of data to or from its own buffer storage to memory with no intervention of CPU. Only one interrupt is generated per block to tell device driver that transfer is complete. Following fig shows interplay of all components of computer system.**


thread of execution
CPU (*N)
cache
instruction execution cycle
data movement
instructions and data
memory
I/O request
data
interrupt
DMA
device (*M)

1.3 computer system architecture

Divided into 3 types

1. single processor system

2. multiprocessor systems

3. Clustered systems

1.3.1 single processor system

On single processor system there is one main one CPU capable of executing a general purpose instruction set including instructions from user processes.

1.3.2 multiprocessor systems

- Multiprocessor systems also called as parallel systems or tightly coupled systems. Such systems have two or more processors in close communication sharing computer bus and sometimes the clock, memory and peripheral devices. **IT has 3 main advantages**
- **Increased throughput** : by increasing number of processors, we expect to get more work done in less time. The speed up ratio with N processors in not N, however its less then N because certain overhead is involved.
- **Economy of scale**: multiprocessor systems costs less than multiple single processor systems because they share peripherals, mass storage and power supplies.
- **Increased reliability** : If functions can be distributed properly among several processors then

Continued..

Graceful degradation : the ability to continue providing service proportional to the level of surviving hardware is called graceful degradation.

Fault tolerant: failure of one processor will not halt the entire system.

Types of multiprocessor systems.

1. Asymmetric multiprocessing .
2. Symmetric multiprocessing .

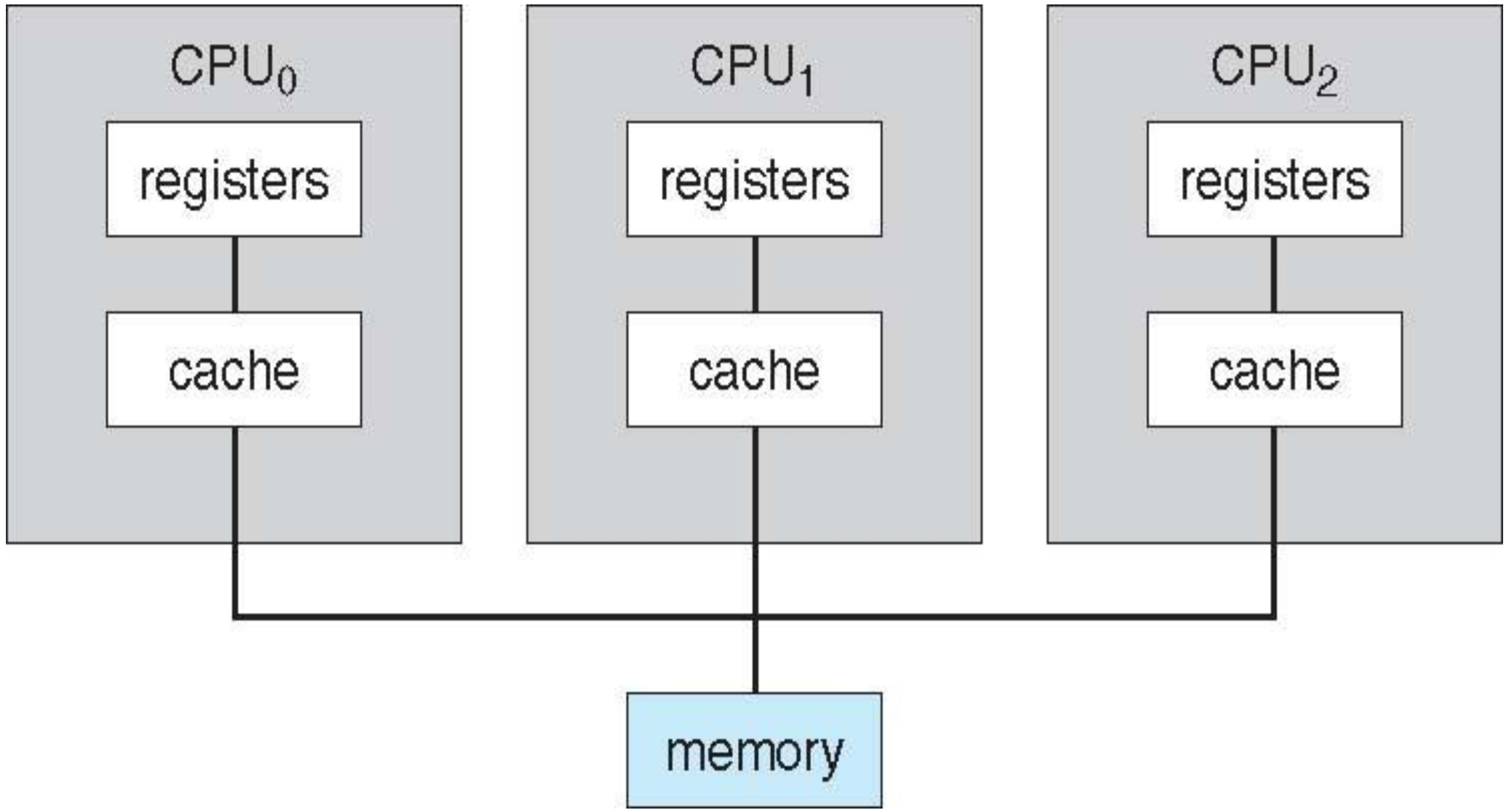
Asymmetric multiprocessing :

It works on master-slave relationship. Where a master processor controls system , a master assigns tasks to slave or slave has predefined tasks.

Symmetric multiprocessing :

Here each processor performs all tasks within the operating system. SMP means all processors are peers. No master-slave processor exits between processors. Following figure illustrates SMP architecture.

- Each processor has its own set of registers and local cache but all processors share physical memory.


CPU0
registers
cache
CPU1
registers
cache
CPU2
registers
cache
memory

Continued..

Advantages of multiprocessor system

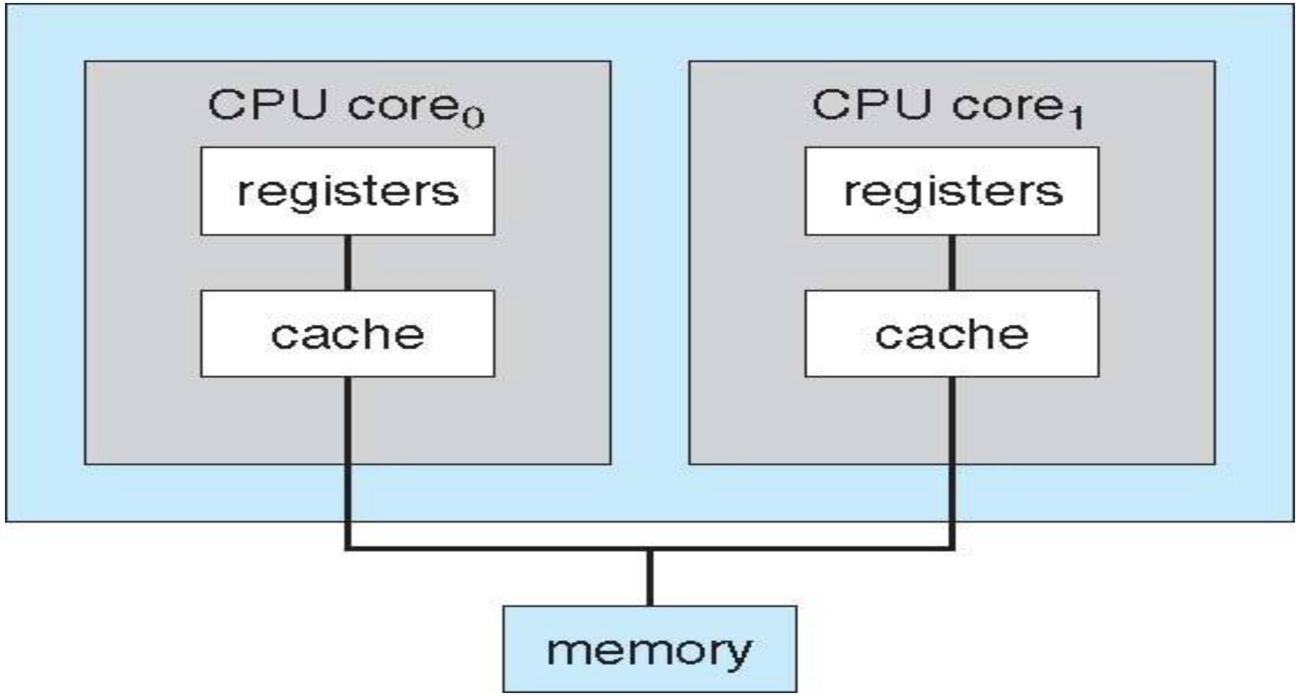
- All processors can run simultaneously without causing a significant deterioration of performance.

Disadvantage

Since CPUs are separate one may be sitting idle and another may be overloaded.

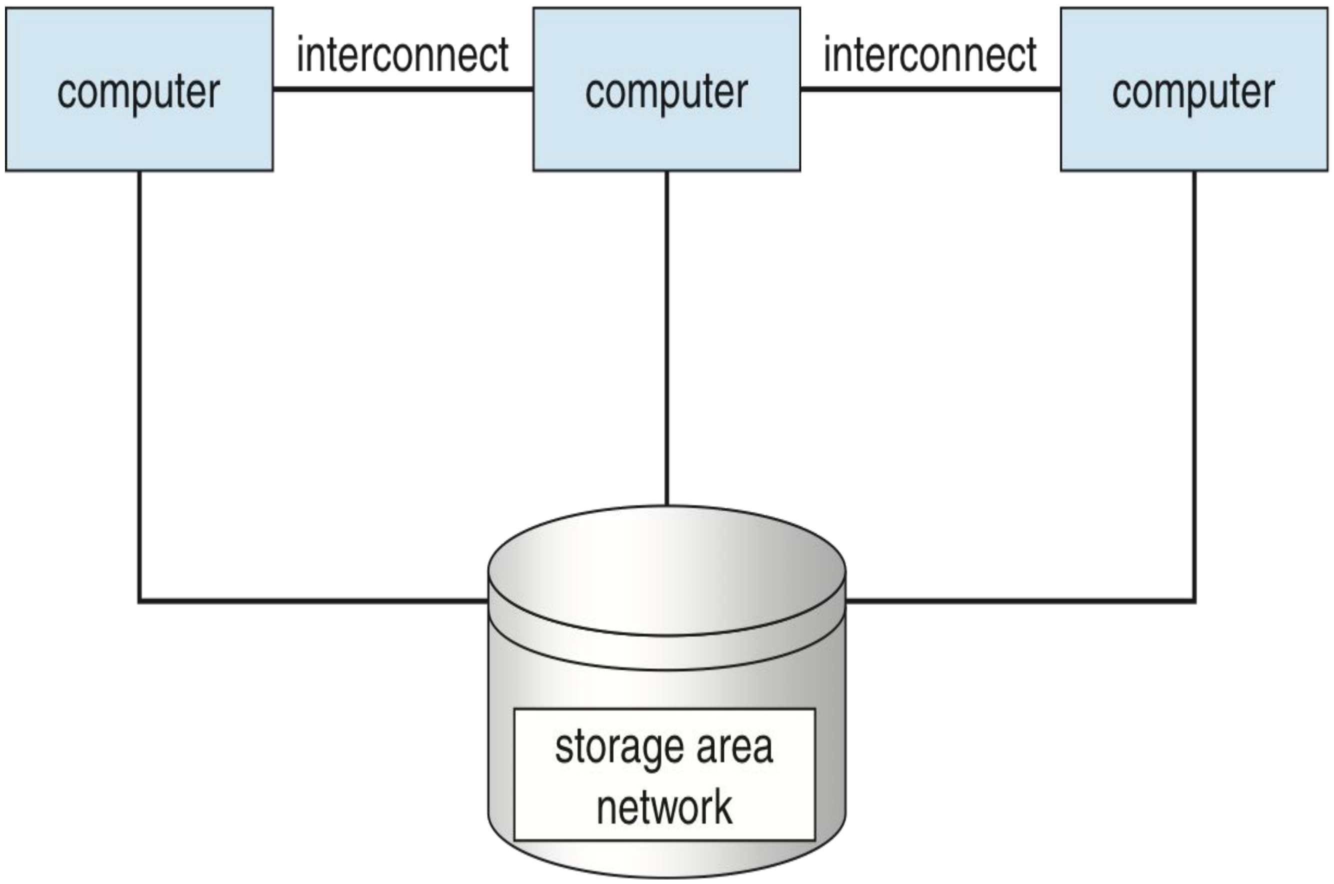
Dual care design

- In Figure we show a dual-core design with two cores on the same chip. In this design, each core has its own register set as well as its own local cache; other designs might use a shared cache or a combination of local and shared caches.
- Aside from architectural considerations, such as cache, memory, and bus contention, these multicore CPUs appear to the operating system as *N* standard processors


CPU core$_0$
registers
cache
CPU core$_1$
registers
cache
memory

1.3.3 clustered system

- Another type of multiple-CPU system is the clustered system. Like multiprocessor systems, clustered systems gather together multiple CPUs to accomplish computational work. Clustered systems differ from multiprocessor systems, however, in that they are composed of two or more individual systems-or nodes-joined together.
- **The generally accepted definition is that clustered computers share storage and are closely linked via a local area network.**
- **Clustering is usually used to provide high availability service**; that is, service will continue even if one or more systems in the cluster fail.
- General structure of clustered system is shown in figure..
- Cluster technology is changing rapidly. Some cluster products support dozens of systems in a cluster, as well as clustered nodes that are separated by miles.
- Many of these improvements are made possible by storage area Network(SAN) which allows many systems to attach to a pool of Storage.
- If the applications and their data are stored on the SAN, then the cluster software can assign the application to run on any host that is attached to the SAN


computer
interconnect
computer
interconnect
computer
storage area
network

Continued….

Types of clustered system

1. Asymmetric clustering
2. Symmetric clustering

- In Asymmetric clustering one machine is in hot stand by mode while other is running the application. The hot-standby host machine does nothing but monitor the active server. If that server fails, the hot-standby host becomes the active server.
- In symmetric clustering two or more hosts are running applications and are monitoring each other. This mode is obviously more efficient, as it uses all of the available hardware. It does require that more than one application be available to run.
- As a cluster consists of several computer systems connected via network, clusters may also be used to provide **high performance computing environment**.
- Other forms of clusters include **parallel clusters** and clustering over a **wide-area network (WAN)**. Parallel clusters allow multiple hosts to access the same data on the shared storage.

1.5 operating system operations

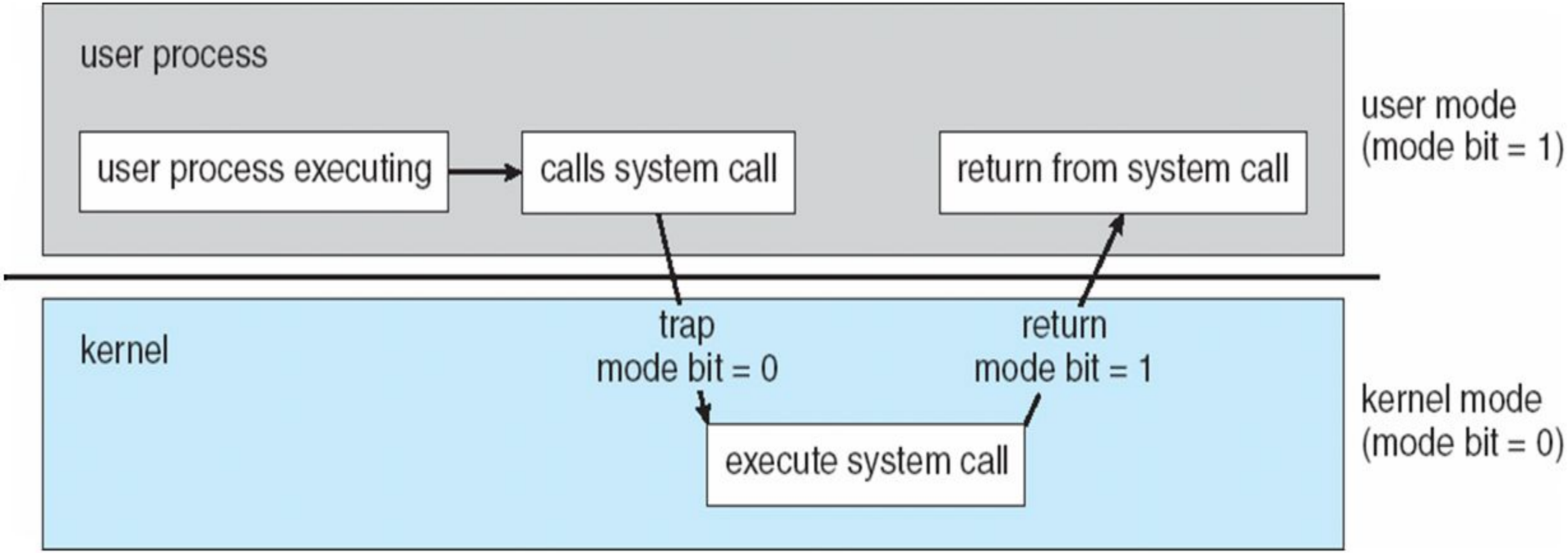
- Modern operating systems are interrupt driven. If there are no processes to execute, no I/0 devices to service, and no users to whom to respond, an operating system will sit quietly waiting for something to happen.
- Events are almost always signaled by the occurrence of an interrupt or a trap.
- **A trap is a software generated interrupt** caused by either by an error (for example divide by zero or invalid memory access) or by a specific request from a user program that an operating system service be performed.

1.5.1 dual mode operation

- **In order to ensure the proper execution of the operating system, we must be able to distinguish between the execution of operating-system code and user defined code.**
- **There are two types of modes.**
- **User mode and kernel mode(supervisory mode or privileged mode)**
- A bit, called mode bit is added to the hardware of the computer to indicate the current mode.

  Kernel (0) ;user(1).
- **With the mode bit we are able to distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.**
- **When computer system is executing on behalf of a user application, the system is in user mode. However,**
- **when a user application requests a service from the operating system (via a system call), it must transition from user to kernel mode to fulfill the request. This is shown in following fig.**

Continued..


user process
user process executing
calls system call
return from system call
user mode
(mode bit = 1)
kernel
trap
mode bit = 0
return
mode bit = 1
execute system call
kernel mode
(mode bit = 0)

Continued..

- Example: At system boot time, the hardware starts in kernel mode. The operating system is then loaded and starts user applications in user mode. Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode(that is, changes the state of the mode bit to 0).
- A system call provides the means for a user program to ask the operating system to perform tasks reserved for operating system on user program's behalf.
- Some of examples which works on dual mode operation are windows XP, unix and linux.

1.5.2 Timer

- **We must ensure that operating system maintains control over CPU**. We can not allow a user program to enter in infinite loop or fail to call system services and never return control to operating system. To accomplish this we can use Timer.
- Timer can be set to interrupt after specific period. The period may fixed or variable.
- The variable timer is implemented by a **fixed rate clock and a counter**. Operating system sets counter.
- For instance, a 10-bit counter with a 1-millisecond clock allows interrupts at intervals from 1 millisecond to 1,024 milliseconds, in steps of 1 millisecond.
- Thus, we can use the timer to prevent a user program from running too long. A simple technique is to initialize a counter with the amount of time that a program is allowed to run.

1.6 process management

- program in execution, as mentioned, is a process. A word-processing program being run by an individual user on a PC is a process.
- A process needs certain resources---including CPU time, memory, files and I/O devices to accomplish task.
- A program is passive activity and process is activity.
- Program counter specifies next instruction to be executed.

The operating system is responsible for the following activities in connection with process management:

- Scheduling processes and threads on the CPUs
- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication

1.7 memory management

- Main memory is a large array of words or bytes, ranging in size from hundreds of thousands to billions. Each word or byte has its own address.
- For a program to be executed, it must be mapped to absolute addresses and loaded into memory. As the program executes, it accesses program instructions and data from memory by generating these absolute addresses.
- To improve both the utilization of the CPU and the speed of the computer's response to its users multiple programs need to be there in memory.

The operating system is responsible for the following activities in connection with memory management:

- 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

1.8 Storage management

1.8.1 **File-System Management**

- File management is one of the most visible components of an operating system. Computers can store information on several different types of physical media. Magnetic disk, optical disk, and magnetic tape are the most common.

The operating system is responsible for the following activities in connection with file management.

- Creating and deleting files
- Creating and deleting directories to organize files
- Supporting primitives for manipulating files and directories
- Mapping files onto secondary storage
- Backing up files on stable (nonvolatile) storage media

1.8.2 Mass-Storage Management

- As we have already seen, because main memory is too small to accommodate all data and programs, and because the data that it holds are lost when power is lost, the computer system must provide secondary storage to back up main memory. The operating system is

responsible for the following activities in connection with disk management:

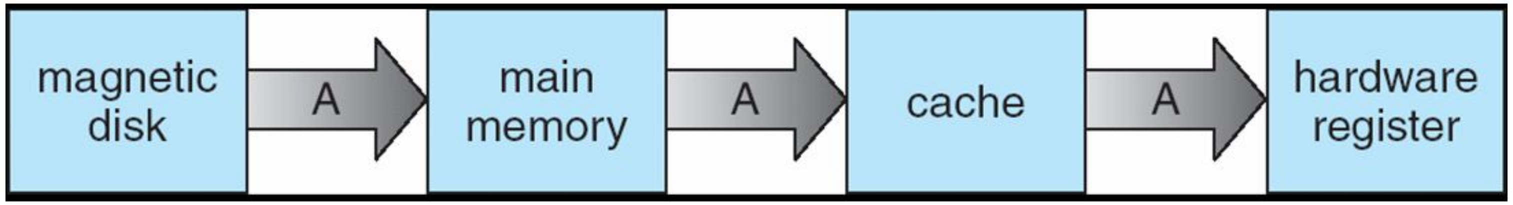
- Free-space management
- Storage allocation
- Disk scheduling

1.8.3 caching

- Information is normally kept in some storage system (such as main memory). As it is used, it is copied into a faster storage system-the cache-on a temporary basis. When we need a particular piece of information, we first check whether it is in the cache. If it is, we use the information directly from the cache; if it is not we use the information from the source.

Continue…

- In hierarchical storage structure, the same data may appear in different levels of the storage system.
- For example, suppose that an integer A that is to be incremented by 1 is located in file B, and file B resides on magnetic disk(in following figure).
- In a computing environment where only one process executes at a time, this arrangement poses no difficulties, since an access to integer A will always be to the copy at the highest level of the hierarchy.
- However, in a multitasking environment, where the CPU is switched back and -forth-among various processes extreme care must be taken to ensure that, if several processes access A, then each of these processes will obtain the most recently updated value of A.
- The situation becomes more complicated in a multiprocessor environment where, in addition to maintaining internal registers, each of the CPUs also contains a local cache.
- Copy of A exits simultaneously in several caches. Since the various CPUs can all execute concurrently we must make sure that update to copy of A in one cache is immediately reflected in all other caches where A resides. This situation is called **cache coherency.**



1.9 Protection and security

- **Protection, then, is any mechanism for controlling the access of processes or users-to the resources defined by a computer system.**
- **Protection can improve reliability by detecting latent errors at the interfaces between component subsystems.**
- A system can have adequate protection but still be prone to failure and allow inappropriate access. Consider a user whose authentication information is stolen. **Her data could be copied or deleted, even though file and memory protection are working.**
- **It is the job of security to defend a system from external and internal attacks. Such attacks spread across a huge range and include viruses and worms, denial-of service attacks**.
- Protection and security require the system to be able to distinguish among all its users. **Most operating systems maintains a list of user names and associated user identifiers.**
- In some circumstances, we wish to distinguish among sets of users rather than individual users. **For example, the owner of a file on a UNIX system may be allowed to issue all operations on that file, whereas a selected set of users may only be allowed to read the file. To accomplish this, we need to define a group name and the set of users belonging *to* that group.**

Distributed systems

- **A distributed system is a collection of physically separate, possibly heterogeneous, computer systems that are networked *to* provide the users with access to the various resources that the system maintains**.
- Access *to* a shared resource increases **computation speed, functionality, data availability, and reliability.**
- A network in the simplest terms, is a communication path between two or more systems.
- Distributed systems depend on networking for their functionality. **Networks vary by the protocols used, the distances between nodes, and the transport media.**
- **TCP /IP is the most common network protocol**. Likewise, operating system support of protocols varies. Most operating systems support TCP /IP, including the Windows and UNIX operating systems.
- Networks are characterized based on the distances between their nodes. **A local area network(LAN) connects computers within a room, a floor, or a building.**
- **A wide area network(WAN) usually links buildings, cities, or countries.**
- **A Metropolitan area network(MAN) could link buildings within a city.**