PET ENGINEERING COLLEGE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CLASS: II CSE / IV SEM

SUBJECT CODE :CS8493

SUBJECT NAME: OPERATING SYSTEMS

S.SAMSUDEEN SHAFFI

UNIT – I OPERATING SYSTEMS OVERVIEW

Computer System Overview-Basic Elements, Instruction Execution, Interrupts, Memory Hierarchy, Cache Memory, Direct Memory Access, Multiprocessor and Multicore Organization. Operating system overview-objectives and functions, Evolution of Operating System.- Computer System Organization-Operating System Structure and Operations-System Calls, System Programs, OS Generation and System Boot.

1. COMPUTER SYSTEM OVERVIEW:

What is an 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 a user of a computer and the computer hardware.
- ✓ The purpose of an operating system is to provide an environment in which a user can execute programs.

Goals of an Operating System

- ✓ The primary goal of an operating system is thus to make the computer system convenient to use.
- ✓ The secondary goal is to use the computer hardware in an efficient manner.

1.1 BASIC ELEMENTS OF A COMPUTER SYSTEM

- ✓ An operating system is an important part of almost every computer system.

A computer system can be divided roughly into four components.

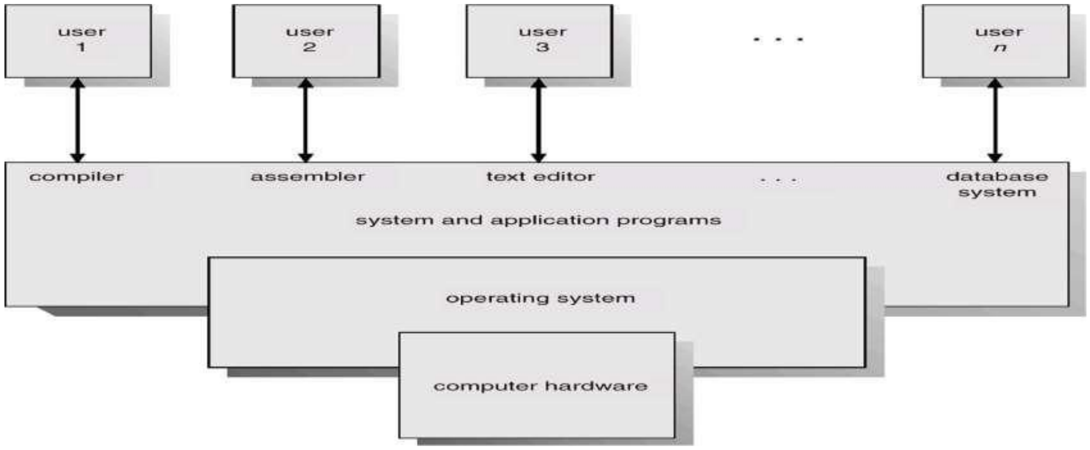
- ❖ **Hardware**
- ❖ **Operating system**
- ❖ **The application programs**
- ❖ **Users**

  - ➢ **The hardware** - the central processing unit (CPU), the memory, and the Input/output (I/O) devices-provides the basic computing resources.
  - ➢ **The application programs-** such as word processors, spreadsheets, compilers, and web browsers- define the ways in which these resources are used to solve the computing problems of the users.
  - ➢ **An operating system** is similar to a government. The OS simply provides an environment within which other programs can do useful work.

Abstract view of the components of a computer system.

Operating system can be viewed as a resource allocator.

- ✓ The OS acts as the manager of the resources (such as CPU time, memory space, file



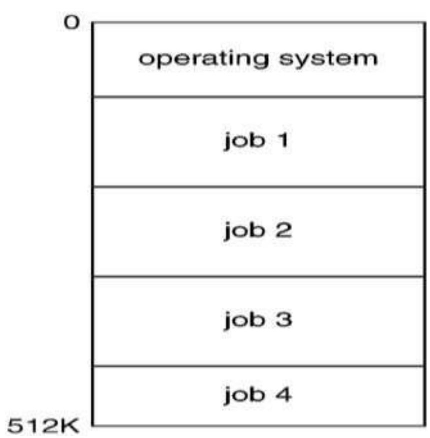
storage space, I/O devices) and allocates them to specific programs and users as necessary for tasks.

- ✓ An operating system is a control program. It controls the execution of user programs to prevent errors and improper use of computer.

1.1.1 Mainframe Systems

- ✓ Early computers were physically enormous machines run from a console.
- ✓ The common input devices were card readers and tape drives.
- ✓ The common output devices were line printers, tape drives, and card punches.
- ✓ The user did not interact directly with the computer systems.
- ✓ Rather, the user prepared a job - which consisted of the program, the data, and some control information about the nature of the job (control cards)-and submitted it to the computer operator.
- ✓ The job was usually in the form of punch cards.
- ✓ The operating system in these early computers was fairly simple.
- ✓ Its major task was to transfer control automatically from one job to the next.
- ✓ The operating system was always resident in memory

Memory layout for a simple batch system.



- ✓ A batch operating system, thus normally reads a stream of separate jobs.
- ✓ When the job is complete its output is usually printed on a line printer.
- ✓ The definitive feature of batch system is the lack of interaction between the user and the job while the job is executing.
- ✓ Spooling is also used for processing data at remote sites.

1.1.2 Multiprogrammed Systems

- ✓ A pool of jobs on disk allows the OS to select which job to run next, to increase CPU utilization.
- ✓ Multiprogramming increases CPU utilization by organizing jobs such that the CPU always has one to execute.
- ✓ The idea is as follows: The operating system keeps several jobs in memory simultaneously. This set of jobs is a subset of the jobs kept in the job pool. The operating system picks and begins to execute one of the jobs in the memory.

Memory layout for a multiprogramming system.

1.1.3 Time-Sharing Systems

- ✓ Time sharing (or multitasking) is a logical extension of multiprogramming.
- ✓ The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- ✓ A time-shared operating system allows many users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.
- ✓ As the system switches rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to her use, even though it is being shared among many users.

1.1.4 Desktop Systems

- ✓ As hardware costs have decreased, it has once again become feasible to have a computer system dedicated to a single user. These types of computer systems are usually referred to as personal computers(PCS).
- ✓ They are microcomputers that are smaller and less expensive than mainframe computers.
- ✓ Operating systems for these computers have benefited from the development of

operating systems for mainframes in several ways.

1.1.5 Multiprocessor Systems

- ✓ Multiprocessor systems (also known as parallel systems or tightly coupled systems) have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices.
- ✓ **Multiprocessor systems have three main advantages.**
  - ❖ Increased throughput.
  - ❖ Economy of scale.
  - ❖ Increased reliablility.
- ✓ If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. If we have ten processors and one fails, then each of the remaining nine processors must pick up a share of the work of the failed processor.
- ✓ Thus, the entire system runs only 10 percent slower, rather than failing altogether. This ability to continue providing service proportional to the level of surviving hardware is called graceful degradation. Systems designed for graceful degradation are also called fault tolerant.
- ✓ Continued operation in the presence of failures requires a mechanism to allow the failure to be detected, diagnosed, and, if possible, corrected.
- ✓ The most common multiple-processor systems now use symmetric multiprocessing (SMP), in whch each processor runs an identical copy of the operating system, and these copies communicate with one another as needed.
- ✓ Some systems use asymmetric multiprocessing, in which each processor is assigned a specific task. A master processor controls the system; the other processors either look to the master for instruction or have predefined tasks.
- ✓ This scheme defines a master-slave relationship. The master processor schedules and allocates work to the slave processors.

1.1.6 Distributed Systems

- ✓ In contrast to the tightly coupled systems, the processors do not share memory or a clock. Instead , each processor has its own local memory.
- ✓ The processors communicate with one another through various communication lines, such as high speed buses or telephone lines. These systems are usually referred to as loosely coupled systems, or distributed systems.

Advantages of distributed systems

- ❖ Resource Sharing
- ❖ Computation speedup
- ❖ Reliability

- Communication

1.1.7 Real-Time Systems

- ✓ Systems that control scientific experiments, medical imaging systems, industrial control systems, and certain display systems are real-time systems.
- ✓ Some automobile-engine fuel-injection systems, home-appliance controllers, and weapon systems are also real-time systems. A real-time system has well-defined, fixed time constraints.
- ✓ Real-time systems come in two flavors: hard and soft.
- ✓ A hard real-time system guarantees that critical tasks be completed on time.
- ✓ This goal requires that all delays in the system be bounded, from the retrieval of stored data to the time that it takes the operating system to finish any request made of it.

1.2 OPERATING SYSTEM COMPONENTS

There are eight major operating system components. They are:

- **Process management**
- **Main-memory management**
- **File management**
- **I/O-system management**
- **Secondary-storage management**
- **Networking**
- **Protection system**
- **Command-interpreter system**

(i) Process Management

- ✓ A process can be thought of as a program in execution. A batch job is a process. A time shared user program is a process.
- ✓ A process needs certain resources-including CPU time, memory, files, and I/O devices-to accomplish its task.
- ✓ A program by itself is not a process; a program is a passive entity, such as the contents of a file stored on disk, whereas a process is an active entity, with a program counter specifying the next instruction to execute.
- ✓ A process is the unit of work in a system.
- ✓ 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

(ii) Main – 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.
- ✓ Main memory is a repository of quickly accessible data shared by the CPU and I/O devices.
- ✓ To improve both the utilization of the CPU and the speed of the computer's response to its users, we must keep several programs 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.

(iii) File Management

- ✓ File management is one of the most visible components of an operating system.
- ✓ The operating system is responsible for the following activities in connection with file management:
- ✓ Creating and deleting files
- ✓ Creating and deleting directories
- ✓ Supporting primitives for manipulating files and directories
- ✓ Mapping files onto secondary storage
- ✓ Backing up files on stable (nonvolatile) storage media

(iv) I/O System management

- ✓ One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. This is done using the I/O subsystem.
- ✓ The I/O subsystem consists of a memory-management component that includes buffering, caching, and spooling.
- ✓ A general device-driver interface
- ✓ Drivers for specific hardware devices

(v) Secondary storage management

- ✓ 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

(vi) Networking

- ✓ A distributed system is a collection of processors that do not share memory, peripheral devices, or a clock.
- ✓ Instead, each processor has its own local memory and clock, and the processors communicate with one another through various communication lines, such as high-speed buses or networks.
- ✓ The processors in the system are connected through a communication network, which can be configured in a number of different ways.

(vii) Protection System

- ✓ Various processes must be protected from one another's activities. For that purpose, mechanisms ensure that the files, memory segments, CPU, and other resources can be operated on by only those processes that have gained proper authorization from the operating system.
- ✓ Protection is any mechanism for controlling the access of programs, 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.

(viii) Command-Interpreter System

- ✓ One of the most important systems programs for an operating system is the command interpreter.
- ✓ It is the interface between the user and the operating system.
- ✓ Some operating systems include the command interpreter in the kernel. Other operating systems, such as MS-DOS and UNIX, treat the command interpreter as a special program that is running when a job is initiated, or when a user first logs on (on time-sharing systems).
- ✓ Many commands are given to the operating system by control statements.
- ✓ When a new job is started in a batch system, or when a user logs on to a time-shared system, a program that reads and interprets control statements is executed automatically.
- ✓ This program is sometimes called the control-card interpreter or **the command-line interpreter, and is often known as the shell.**

1.3 BASIC ELEMENTS

1.3.1 Main Memory

- referred to as real memory or primary memory
- volatile

1.3.2 I/O modules

- secondary memory devices
- communications equipment
- terminals

1.3.3 System bus

- communication among processors, memory, andI/O modules.

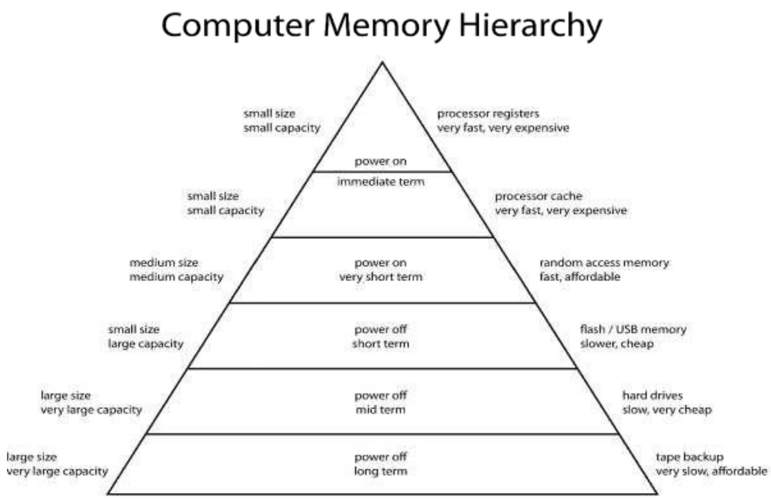
1.4 INTERRUPTS

- ✓ An interrupt is a signal from a device attached to a computer or from a program within the computer that causes the main program that operates the computer (the operating system) to stop and figure out what to do next. Almost all personal (or larger) computers today are interrupt-driven - that is, they start down the list of computer instruction s in one program (perhaps an application such as a word processor) and keep running the instructions until either (A) they can't go any further or (B) an interrupt signal is sensed. Basically, a single computer can perform only one computer instruction at a time. But, because it can be interrupted, it can take turns in which programs or sets of instructions that it performs. This is known as multitasking. It allows the user to do a number of different things at the same time. The computer simply takes turns managing the programs that the user effectively starts. Of course, the computer operates at speeds that make it seem as though all of the user's tasks are being performed at the same time. (The computer's operating system is good at using little pauses in operations and user thinks time to work on other programs.)
- ✓ An operating system usually has some code that is called an interrupt handler. The interrupt handler prioritizes the interrupts and saves them in a queue if more than one is waiting to be handled. The operating system has another little program, sometimes called a scheduler that figures out which program to give control to next.
- ✓ In general, there are hardware interrupts and software interrupts. A hardware interrupt occurs, for example, when an I/O operation is completed such as reading some data into the computer from a tape drive. A software interrupt occurs when an application program terminates or requests certain services from the operating system. In a personal computer, a hardware interrupt request (IRQ) has a value associated with it that associates it with a particular device.

1.5 MEMORY HIERARCHY

Memory is categorized into **volatile and nonvolatile memories**, with the former requiring constant power ON of the system to maintain data storage.

Furthermore, a typical computer system provides a hierarchy of different times of memories for data storage.



1.5.1 Different levels of the memory hierarchy

- ✓ **Cache (MB):** Cache is the fastest accessible memory of a computer system. It's access speed is in the order of a few nanoseconds. It is volatile and expensive, so the typical cache size is in the order of megabytes.

- ✓ **Main memory (GB):** Main memory is arguably the most used memory. When discussing computer algorithms such as quick sort, balanced binary sorted trees, or fast Fourier transform, one typically assumes that the algorithm operates on data stored in the main memory. The main memory is reasonably fast, with access speed around 100 nanoseconds. It also offers larger capacity at a lower cost. Typical main memory is in the order of 10 GB. However, the main memory is volatile.

- ✓ **Secondary storage (TB):** Secondary storage refers to nonvolatile data storage units that are external to the computer system. Hard drives and solid state drives are examples of secondary storage. They offer very large storage capacity in the order of terabytes at very low cost. Therefore, database servers typically have an array of secondary storage devices with data stored distributed and redundantly across these devices. Despite the continuous improvements in access speed of hard drives, secondary storage devices are several magnitudes slower than main memory. Modern hard drives have access speed in the order of a few milliseconds.

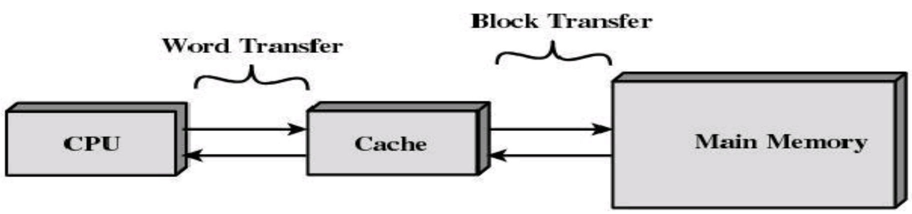
- ✓ **Tertiary storage (PB):** Tertiary storage refers storage designed for the purpose data backup. Examples of tertiary storage devices are tape drives are robotic driven disk arrays. They are capable of petabyte range storage, but have very slow access speed with data access latency in seconds or minutes.

1.6 CACHE MEMORY

- ✓ **Cache (pronounced cash)** memory is extremely fast memory that is built into a computer’s central processing unit (CPU), or located next to it on a separate chip. The CPU uses cache memory to store instructions that are repeatedly required to run programs, improving overall system speed. The advantage of cache memory is that the CPU does not have to use the motherboard’s system bus for data transfer. Whenever data must be passed through the system bus, the data transfer speed slows to the motherboard's capability. The CPU can process data much faster by avoiding the bottleneck created by the system bus.

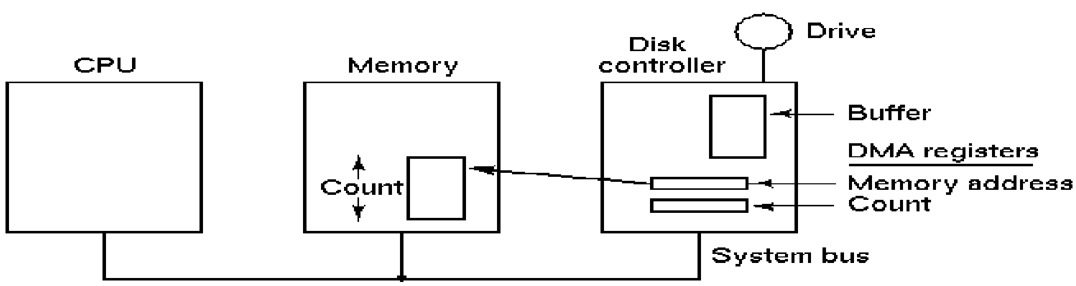
- ✓ As it happens, once most programs are open and running, they use very few resources. When these resources are kept in cache, programs can operate more quickly and efficiently. All else being equal, cache is so effective in system performance that a computer running a fast CPU with little cache can have lower benchmarks than a system running a somewhat slower CPU with more cache. Cache built into the CPU itself is referred to as Level 1 (L1) cache. Cache that resides on a separate chip next to the CPU is called Level 2 (L2) cache. Some CPUs have both L1 and L2 cache built-in and designate the separate cache chip as Level 3 (L3) cache.

- ✓ Cache that is built into the CPU is faster than separate cache, running at the speed of the microprocessor itself. However, separate cache is still roughly twice as fast as Random Access Memory (RAM). Cache is more expensive than RAM, but it is well worth getting a CPU and motherboard with built-in cache in order to maximize system performance.



1.7 DIRECT MEMORY ACCESS

- ✓ **Direct Memory Access (DMA)** is a capability provided by some computer bus architectures that allows data to be sent directly from an attached device (such as a disk drive) to the memory on the computer's motherboard. The microprocessor is freed from involvement with the data transfer, thus speeding up overall computer operation.



- ✓ Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU initiates the transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller when the operation is done.
- ✓ This feature is useful any time the CPU cannot keep up with the rate of data transfer, or where the CPU needs to perform useful work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards.
- ✓ DMA is also used for intra-chip data transfer in multi-core processors. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without DMA channels. Similarly, a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time, allowing computation and data transfer to proceed in parallel.

1.7.1 Modes of operation

1.7.1.1Burst mode

- ✓ An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system bus by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU, but renders the CPU inactive for relatively long periods of time. The mode is also called "Block Transfer Mode". It is also used to stop unnecessary data.

1.7.1.2 Cycle stealing mode

- ✓ The cycle stealing mode is used in systems in which the CPU should not be disabled for the length of time needed for burst transfer modes.
- ✓ In the cycle stealing mode, the DMA controller obtains access to the system bus the same way as in burst mode, using BR (Bus Request) and BG (Bus Grant) signals, which are the two signals controlling the interface between the CPU and the DMA controller.
- ✓ However, in cycle stealing mode, after one byte of data transfer, the control of the system bus is deasserted to the CPU via BG.

1.7.1.3 Transparent mode

- ✓ The transparent mode takes the most time to transfer a block of data, yet it is also the most efficient mode in terms of overall system performance. The DMA controller only transfers data when the CPU is performing operations that do not use the system buses.
- ✓ It is the primary advantage of the transparent mode that the CPU never stops executing its programs and the DMA transfer is free in terms of time. The disadvantage of the transparent mode is that the hardware needs to determine when the CPU is not using the system buses, which can be complex.

1.8 MULTIPROCESSOR AND MULTICORE ORGANIZATION

- ✓ Multiprocessor Operating System refers to the use of two or more central processing units (CPU) within a single computer system. These multiple CPUs are in a close communication sharing the computer bus, memory and other peripheral devices. These systems are referred as tightly coupled systems.

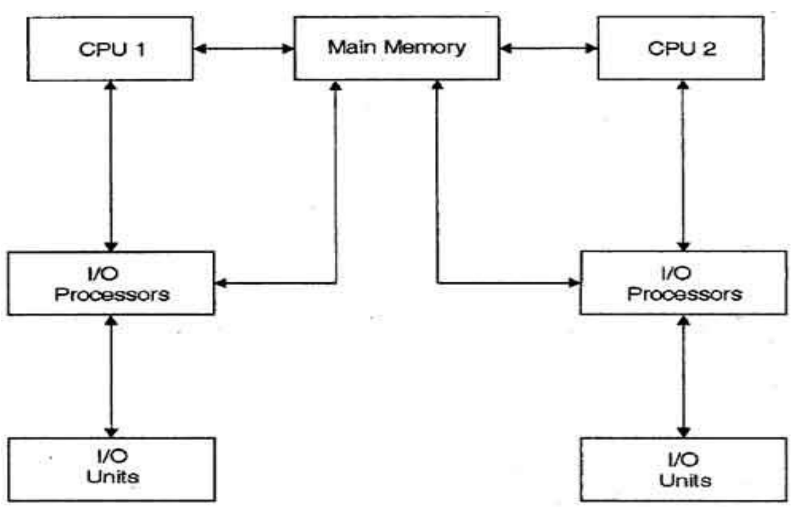


Fig 1.8 Multiprocessor Organization

- ✓ Multiprocessing system is based on the symmetric multiprocessing model, in which each processor runs an identical copy of operating system and these copies communicate with each other. In this system processor is assigned a specific task. A master processor controls the system. This scheme defines a master-slave relationship.

- ✓ These systems can save money in compare to single processor systems because the processors can share peripherals, power supplies and other devices. The main advantage of multiprocessor system is to get more work done in a shorter period of time. Moreover, multiprocessor systems prove more reliable in the situations of failure of one processor. In this situation, the system with multiprocessor will not halt the system; it will only slow it down.

- ✓ In order to employ multiprocessing operating system effectively, the computer system must have the followings:

1.8.1 Motherboard Support:

A motherboard capable of handling multiple processors. This means additional sockets or slots for the extra chips and a chipset capable of handling the multiprocessing arrangement.

1.8.2 Processor Support:

- ✓ Processors those are capable of being used in a multiprocessing system.

- ✓ The whole task of multiprocessing is managed by the operating system, which allocates different tasks to be performed by the various processors in the system.
- ✓ Multiprocessor system supports the processes to run in parallel. Parallel processing is the ability of the CPU to simultaneously process incoming jobs. This becomes most important in computer system, as the CPU divides and conquers the jobs. Generally the parallel processing is used in the fields like artificial intelligence and expert system, image processing, weather forecasting etc.

- ✓ In a multiprocessor system, the dynamically sharing of resources among the various processors may cause therefore, a potential bottleneck. There are three main sources of contention that can be found in a multiprocessor operating system:

1.8.2.1 Locking system:

- ❖ In order to provide safe access to the resources shared among multiple processors, they need to be protected by locking scheme. The purpose of a locking is to serialize accesses to the protected resource by multiple processors. Undisciplined use of locking can severely degrade the performance of system.

- ❖ This form of contention can be reduced by using locking scheme, avoiding long critical sections, replacing locks with lock-free algorithms, or, whenever possible, avoiding sharing altogether.

1.8.2.2 d data:

- ❖ The continuous accesses to the shared data items by multiple processors (with one or more of them with data write) are serialized by the cache coherence protocol. Even in a moderate-scale system, serialization delays can have significant impact on the system performance.
- ❖ In addition, bursts of cache coherence traffic saturate the memory bus or the interconnection network, which also slows down the entire system. This form of contention can be eliminated by either avoiding sharing or, when this is not possible, by using replication techniques to reduce the rate of write accesses to the shared data.

1.8.2.3 False sharing:

- ❖ This form of contention arises when unrelated data items used by different processors are located next to each other in the memory and, therefore, share a single cache line: The effect of false sharing is the same as that of regular sharing bouncing of the cache line among several processors. Fortunately, once it is identified, false sharing can be easily eliminated by setting the memory layout of non-shared data.

1.9 OPERATING SYSTEM OVERVIEW

1.9.1 Objectives and Functions

- A program that is executed by the processor that frequently relinquishes control and must depend on the processor to regain control.
- A program that mediates between application programs and the hardware
- A set of procedures that enable a group of people to use a computer system.
- A program that controls the execution of application programs
- An interface between applications and hardware

1.9.2 Functions

Usage

Computer system

Control

Support

Usage

- ❖ Users of a computer system:
- ❖ Programs - use memory, use CPU time, use I/O devices
- ❖ Human users
- ❖ Programmers - use program development tools such as debuggers, editors end users - use application programs, e.g. Internet explorer

Computer system

hardware + software

OS is a part of the computer software, it is a program. It is a very special program, that is the first to be executed when the computer is switched on, and is supposed to control and support the execution of other programs and the overall usage of the computer system.

Control

The operating system controls the usage of the computer resources - hardware devices and software utilities. We can think of an operating system as a ***Resource Manager*.** Here are some of the resources managed by the OS:

- Processors,
- Main memory,
- Secondary Memory,

- Peripheral devices,
- Information.

Support

- ✓ The operating system provides a number of services to assist the users of the computer system:

For the programmers:

Utilities - debuggers, editors, file management, etc.

For the end users - provides the interface to the application programs

For programs - loads instructions and data into memory, prepares I/O devises for usage, handles interrupts and error conditions.

1.10 EVOLUTION OF OPERATING SYSTEM

1.10.1 Serial Processing - 1940's – 1950's programmer interacted directly with hardware. No operating system.

Problems

Scheduling - users sign up for machine time. Wasted computing time

Setup Time- Setup included loading the compiler, source program, saving compiled program, and loading and linking. If an error occurred - start over.

1.10.2 Simple Batch Systems

Improve the utilization of computers**.**

Jobs were submitted on cards or tape to an operator who batches jobs together sequentially. The program that controls the execution of the jobs was called **monitor** - a simple version of an operating system. The interface to the monitor was accomplished through Job Control Language (JCL). For example, a JCL request could be to run the compiler for a particular programming language, then to link and load the program, then to run the user program.

Hardware features:

Memory protection: do not allow the memory area containing the monitor to be altered

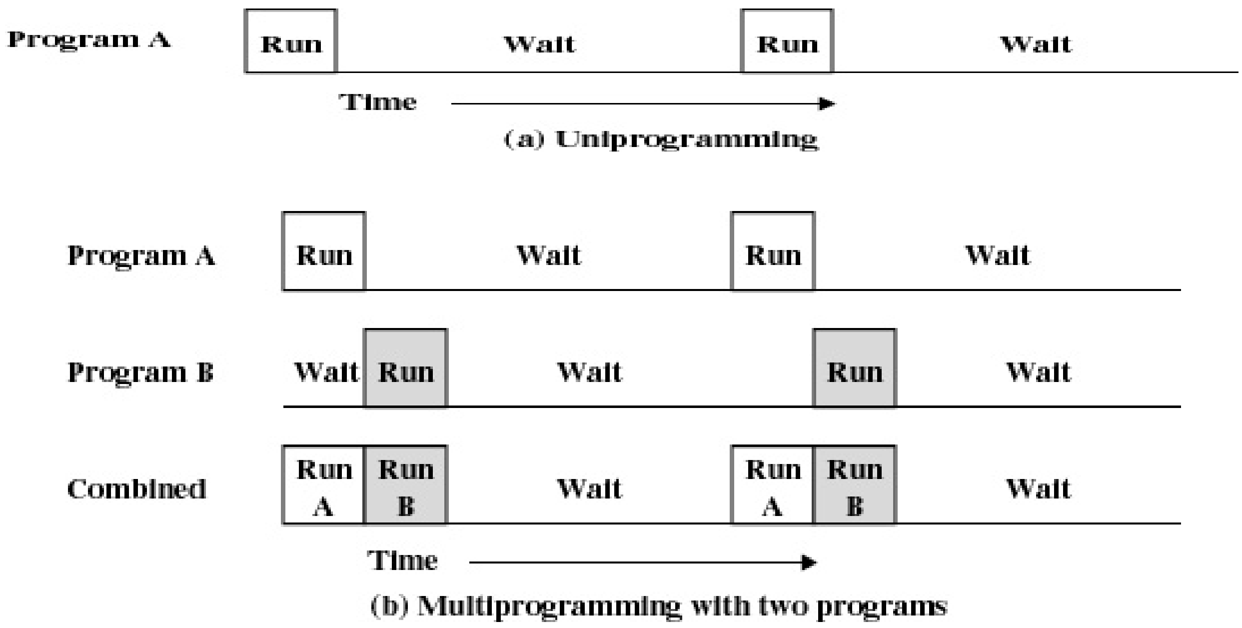
Timer: prevents a job from monopolizing the system

Problems:

Bad utilization of CPU time - the processor stays idle while I/O devices are in use.

1.10.3 Multiprogrammed Batch Systems

More than one program resides in the main memory. While a program A uses an I/O device the processor does not stay idle, instead it runs another program B.



New features:

Memory management - to have several jobs ready to run, they must be kept in main memory

Job scheduling - the processor must decide which program to run.

1.10.4 Time-Sharing Systems

Multiprogramming systems: Several programs use the computer system.

Time-sharing systems: Several (human) users use the computer system interactively.

Characteristics:

- Using multiprogramming to handle multiple interactive jobs
- Processor's time is shared among multiple users
- Multiple users simultaneously access the system through terminals

1.10.5 Operating-System Services

The OS provides certain services to programs and to the users of those programs.

1. **Program execution: The system must be able to load a program into memory and to run that program. The program must be able to end its execution, either normally or abnormally (indicating error).**

2. **I/O operations:**

   A running program may require I/O. This I/O may involve a file or an I/O device.

3. **File-system manipulation:**

   The program needs to read, write, create and delete files.

4. **Communications :**

In many circumstances, one process needs to exchange information with another process. Such communication can occur in two major ways. The first takes place between processes that are executing on the same computer; the second takes place between processes that are executing on different computer systems that are tied together by a computer network.

5. **Error detection:**

The operating system constantly needs to be aware of possible errors. Errors may occur in the CPU and memory hardware (such as a memory error or a power failure), in I/O devices (such as a parity error on tape, a connection failure on a network, or lack of paper in the printer), and in the user program (such as an arithmetic overflow, an attempt to access an illegal memory location, or a too-great use of CPU time). For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.

6. **Resource allocation:**

Different types of resources are managed by the Os.

When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them.

7. **Accounting:**

We want to keep track of which users use how many and which kinds of computer resources. This record keeping may be used for accounting or simply for accumulating usage statistics.

8. **Protection:**

The owners of information stored in a multiuser computer system may want to control use of that information. Security of the system is also important.

1.10.6 CP derivatives

1. CP
2. CP-VM
3. CP/M
    1. CP/M-86
    2. DOS
    3. DRDOS
        1. FreeDOS
4. Microsoft Windows
    1. Windows 3.x
    2. Windows 95/98/Me
    3. Windows Xp
    4. Windows Vista
    5. Windows 7
    6. windows 8

1.10.7 MULTICS derivatives

1. UNIX
    - Unix V5 (SCO Unix)
    - Modern Unix (Solaris / BSD)
    - FreeBSD
    - OpenBSD
2. Xenix
3. Linux
4. Plan 9 jim bading tahaha
    - Inferno to hell
5. QNX
6. VSTa
7. Mac OSX
8. MIPS RISC/os
9. RISC iX

1.10.8 VMS derivatives

1. VMS
    1. OpenVMS
2. OS/2
    1. eComStation
3. Microsoft Windows =rude-magic!
    1. ReactOS

1.10.9 MacOS derivatives

1. Mac OS
    1. Mac OS 9

1.10.10 Cambridge CAP Computer derivatives

1. KeyKOS
    1. EROS
    2. Coyotos
    3. CapROS

1.10.11 IBSYS derivatives

1. OS/360
2. OS/400

1.10.12 AmigaOS Derivatives

1. AmigaOS
2. AROS
    1. AmigaOS 4.x
    2. MorphOS

1.10.13 Novel Operating Systems

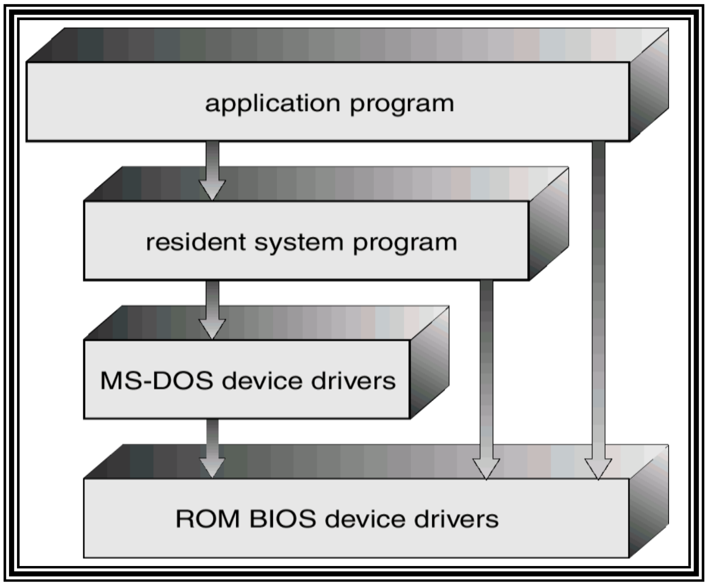
1. Grasshopper
2. ITS
3. BeOS

1.11 COMPUTER SYSTEM ORGANIZATION

1.11.1 Operating System Structure

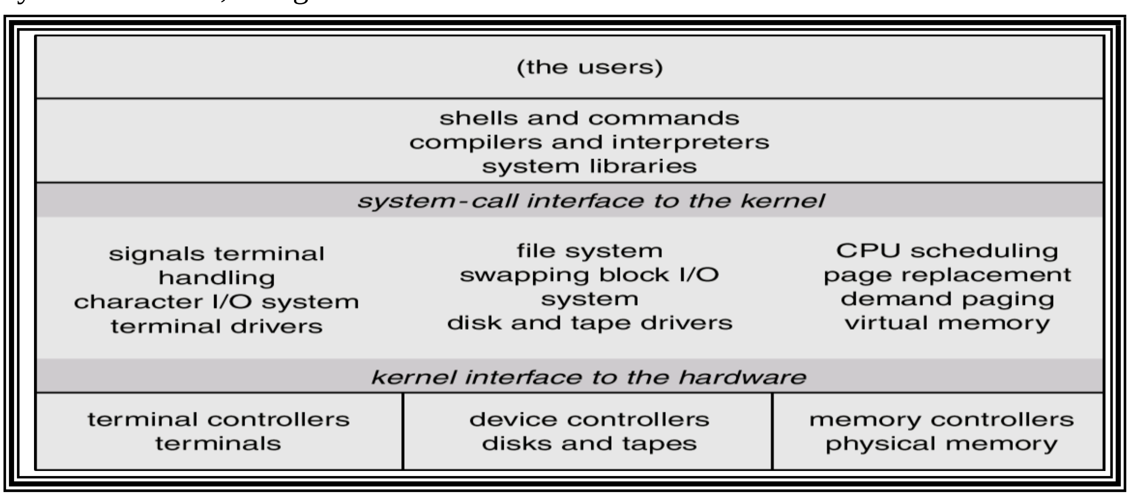
1.11.1.1 MS-DOS System Structure

- ✓ MS-DOS – written to provide the most functionality in the least space.
- ✓ Not divided into modules.
- ✓ Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated.



1.11.1.2 Unix System Structure

- ➢ **UNIX** – limited by hardware functionality, the original UNIX operating system had limited structuring. The UNIX OS consists of two separable parts.
- ➢ **Systems programs** – use kernel supported system calls to provide useful functions such as compilation and file manipulation.
- ➢ **The kernel** - Consists of everything below the system-call interface and above the physical hardware
- ➢ Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level.



1.11.1.3 Layered Approach

- ✓ The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- ✓ An OS layer is an implementation of an abstract object that is the encapsulation of data and operations that can manipulate those data. These operations (routines) can be invoked by higher-level layers. The layer itself can invoke operations on lower-level layers.
- ✓ Layered approach provides modularity. With modularity, layers are selected such that each layer uses functions (operations) and services of only lower-level layers.
- ✓ Each layer is implemented by using only those operations that are provided lower level layers.
- ✓ The major difficulty is appropriate definition of various layers.

1.11.1.4 Microkernel System Structure

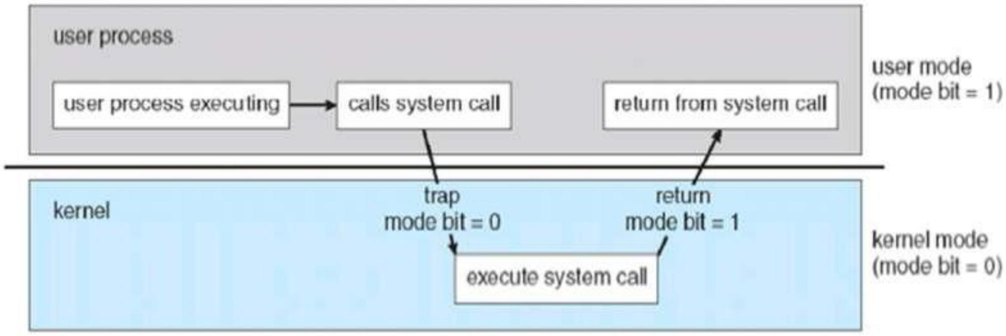
- ✓ Moves as much from the kernel into “user” space.
- ✓ Communication takes place between user modules using message passing.
    - ❖ Benefits:
        - ➢ Easier to extend a microkernel
        - ➢ Easier to port the operating system to new architectures
        - ➢ More reliable (less code is running in kernel mode)
        - ➢ More secure

1.11.2 Operating-System Operations

- ✓ If there are no processes to execute, no I/O 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 (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed. The interrupt-driven nature of an operating system defines that system’s general structure.
- ✓ Without protection against these sorts of errors, either the computer must execute only one process at a time or all output must be suspect. A properly designed operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.

1.11.2.1 Dual-Mode and Multimode 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. The approach taken by most computer systems is to provide hardware support that allows us to differentiate among various modes of execution.



- ✓ At the very least, we need two separate modes of operation: **user mode and kernel mode (also called supervisor mode, system mode, or privileged mode)**.
- ✓ A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). With the mode bit, we can distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.

1.11.2.2 Timer

- ✓ We cannot allow a user program to get stuck in an infinite loop or to fail to call system services and never return control to the operating system.
- ✓ To accomplish this goal, we can use a timer. A timer can be set to interrupt the computer after a specified period. The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).
- ✓ A variable timer is generally implemented by a fixed-rate clock and a counter.
- ✓ The operating system sets the counter. Every time the clock ticks, the counter is decremented. When the counter reaches 0, an interrupt occurs. 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.

1.13 SYSTEM CALLS

- ✓ System calls provide the interface between a process and the operating system.
- ✓ These calls are generally available as assembly-language instructions.
- ✓ System calls can be grouped roughly into five major categories:

  1. **Process control**
  2. **file management**
  3. **device management**
  4. **information maintenance**
  5. **communications.**

1. Process Control

- end,abort
- load, execute
- Create process and terminate process
- get process attributes and set process attributes.
- wait for time, wait event, signal event
- Allocate and free memory.

2. File Management

- Create file, delete file
- Open , close
- Read, write, reposition
- Get file attributes, set file attributes.

3. Device Management

- Request device, release device.
- Read, write, reposition
- Get device attribtues, set device attributes
- Logically attach or detach devices

4. Information maintenance

- Get time or date, set time or date
- Get system data, set system data
- Get process, file, or device attributes
- Set process, file or device attributes

5. Communications

- Create, delete communication connection
- Send, receive messages
- Transfer status information
- Attach or detach remote devices

Two types of communication models

(a) Message passing model

(b) Shared memory model

1.14 SYSTEM PROGRAMS

- ✓ System programs provide a convenient environment for program development and execution.
- ✓ They can be divided into several categories:

1. File management: These programs create, delete, copy, rename, print, dump, list and generally manipulate files and directories.

2. Status information: The status such as date, time, amount of available memory or diskspace, number of users or similar status information.

3. File modification: Several text editors may be available to create and modify the content of files stored on disk or tape.

4. Programming-language support: Compilers, assemblers, and interpreters for common programming languages are often provided to the user with the operating system.

5. Program loading and execution: The system may provide absolute loaders, relocatable loaders, linkage editors, and overlay loaders.

6. Communications: These programs provide the mechanism for creating virtual connections among processes, users, and different computer systems. (email, FTP, Remote log in)

7. Application programs: Programs that are useful to solve common problems, or to perform common operations.

Eg. Web browsers, database systems.