Module 1

INTRODUCTION TO OPERATING SYSTEMS

Program

A Program is a set of statements or instructions which collectively performs a particular task.

Software

A Software is a set of programs which collectively performs a set of activities.

Types of software

1) System software
2) Application software

System software

Software which provides services to other software is called as system software.

Example system software are: Operating System, Compiler, Linker, Loader, Assembler, debugger, driver and so on.

System Software is used for operating the computer hardware.

System Software is generally installed in the computer when the operating system is installed.

The user does not interact with the system software because it works in the background.

System software provides platform for running application software.

Application software

Software which depends on other software for services is called as application software.

Example application software are: VLC player, web browser, word processor and so on.

Application software is used by the user to perform a specific task.

Application software are installed according to user’s requirements.

The user interacts with application software.

Application software can’t run without the presence of system software.

Computer

A computer contains number of hardware and software components.

Each component of the computer system is called a resource.

Some of the hardware resources are: processor, HD, RAM, keyboard, monitor and so on.

Some of the software resources are: operating system, compiler, interpreter, loader, assembler and so on.

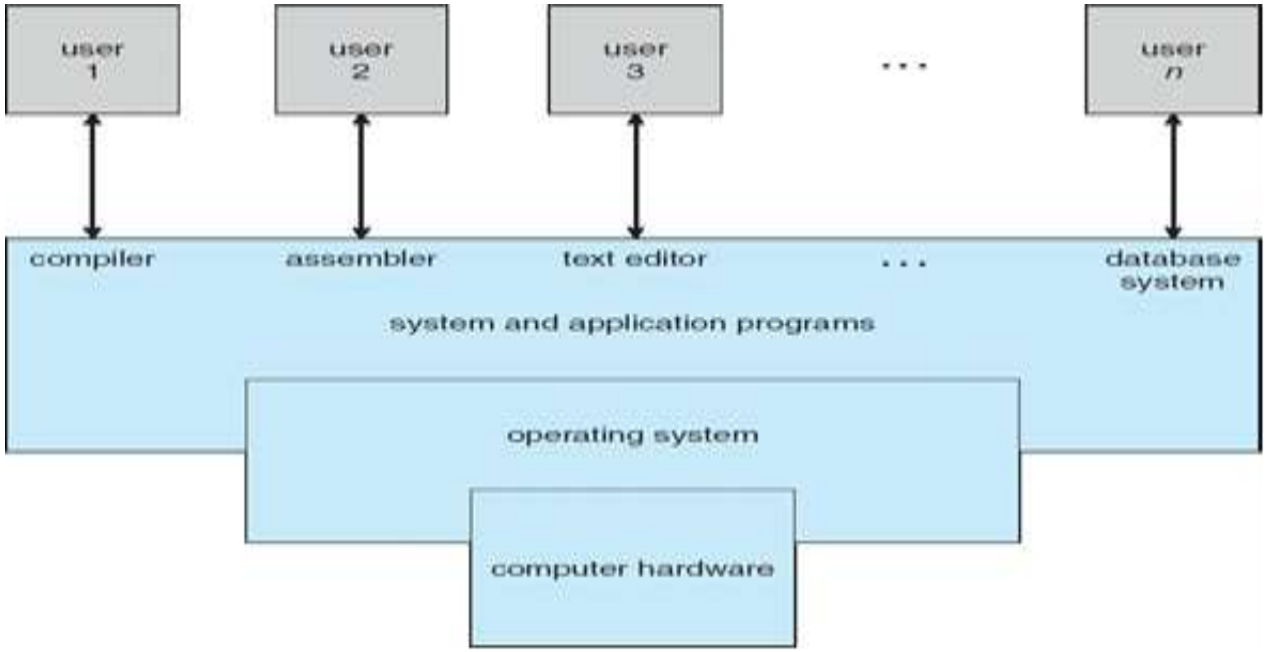
Operating System

An Operating System is a system software that manages the computer hardware.

Operating System acts as an intermediary between the computer user and the computer hardware.

Some operating systems are designed to provide convenient communication between user and computer system, others for efficient utilization of resources, and others to provide both.

Role of Operating System



A computer system can be logically divided into four components: the hardware, the operating system, the application programs, and the users as shown in above figure.

The **hardware** (the central processing unit, the memory and the I/O devices) provides the computing resources for the computer system.

The **system programs** (compilers, assemblers and so on) and **application programs** (word processors, spreadsheets, web browsers and so on) help the user programs in their execution.

The **operating system** controls the hardware and coordinates the use of hardware among various user programs.

The role of operating system is fully explained by considering two views: user view and system view.

User view

Computer systems are divided into four categories from user point of view.

1) Personal computers

Used by single person for home applications. The Operating System used in personal computers should focus on the following

- Easy use of the system
- Utilization of resources
- Performance of the system

2) Thin clients

A thin client is a system which do not have any processing capability and storage capacity.

It is used for typing the program and displaying the output.

Thin clients are connected to a server system which has storage capacity and execution capability.

The programs typed in the thin clients are stored and executed in server system.

The results of execution are passed to client systems.

Ex: Oracle server

The operating system used in thin clients should concentrate on resource utilization.

3) Client – Server system

Number of client systems are connected to a server system.

A client system makes a request to a server system for any service.

The server system provides service to all client systems connected to it.

The Operating System used in client system should use resources of the client system as well as resources available at the server system.

4) Home devices or devices used in automobiles

An example for home device is a microwave oven.

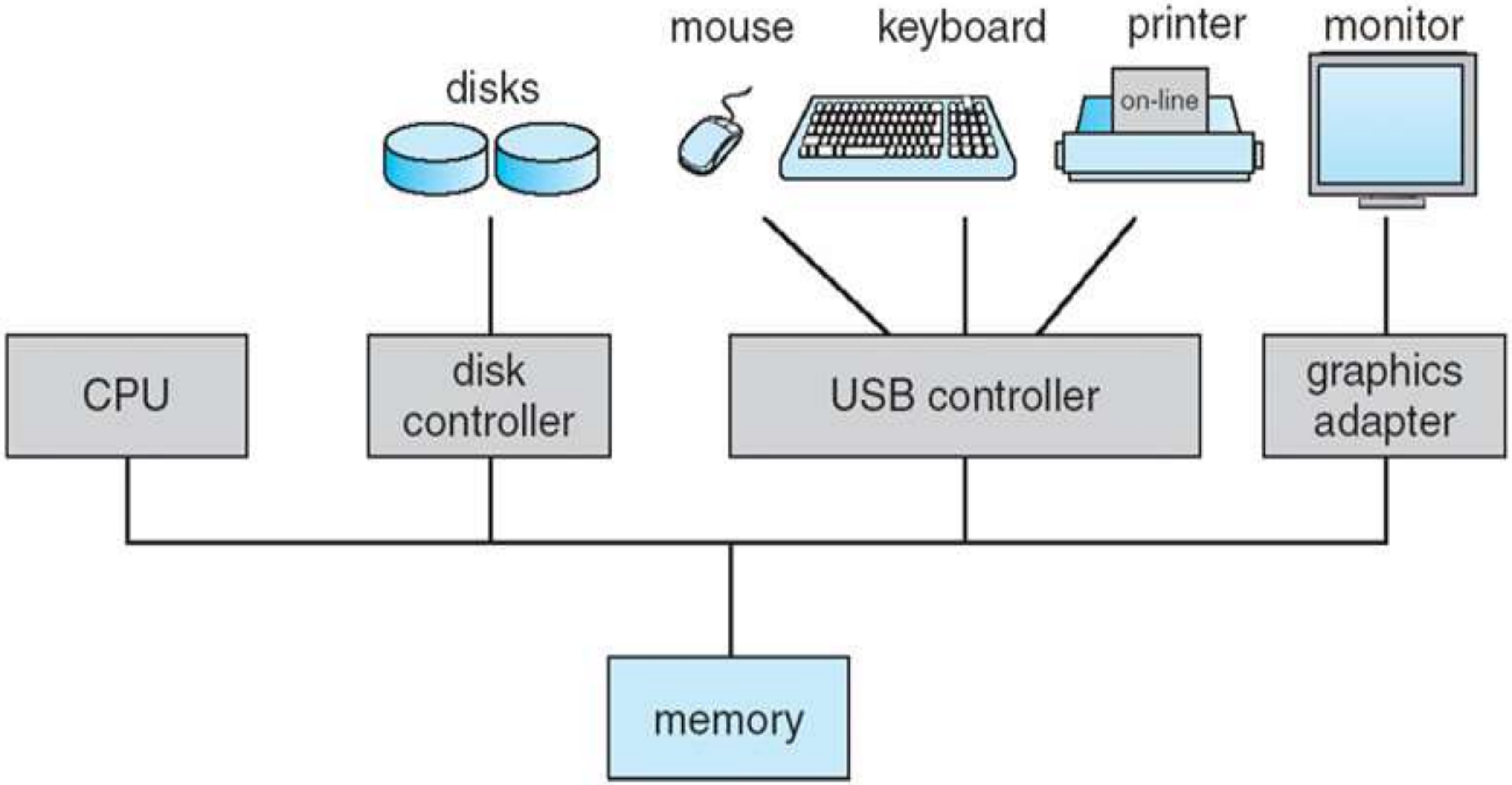
The Operating System used in home devices should work automatically without any intervention of user.

System view

From system point of view, the Operating System can be viewed as resource allocator.

The Operating System allocates the resources of the computer system to the programs that are currently running in the system.

Computer System Organization



A modern computer system consists of one or more CPUs and number of devices.

The CPUs and devices are connected to the system bus as shown in diagram.

The devices are connected to the system bus through device controllers.

Each device controller controls the operations of a device or a set of devices.

A device controller maintains some local buffer storage and a set of special-purpose registers.

The device controller is responsible for moving the data between the devices that it controls and its local buffer storage.

operating systems have a **device driver** for each device controller.

When the computer system is booted, a program called bootstrap program starts running.

The bootstrap program resides in ROM or EEPROM.

Role of bootstrap program is locating the operating system, loading the operating system into RAM and starting the execution of operating system.

When the operating system is started then the operating system controls the operations of computer system.

Interrupts

Consider a program performing I/O.

To start an I/O operation, the device driver loads the appropriate registers in the device controller.

The device controller, in turn, examines the contents of these registers to determine what action to take (such as “read a character from the keyboard”).

The controller starts the transfer of data from the device to its local buffer.

Once the transfer of data is complete, the device controller informs the device driver that it has finished its operation by raising an interrupt (a signal).

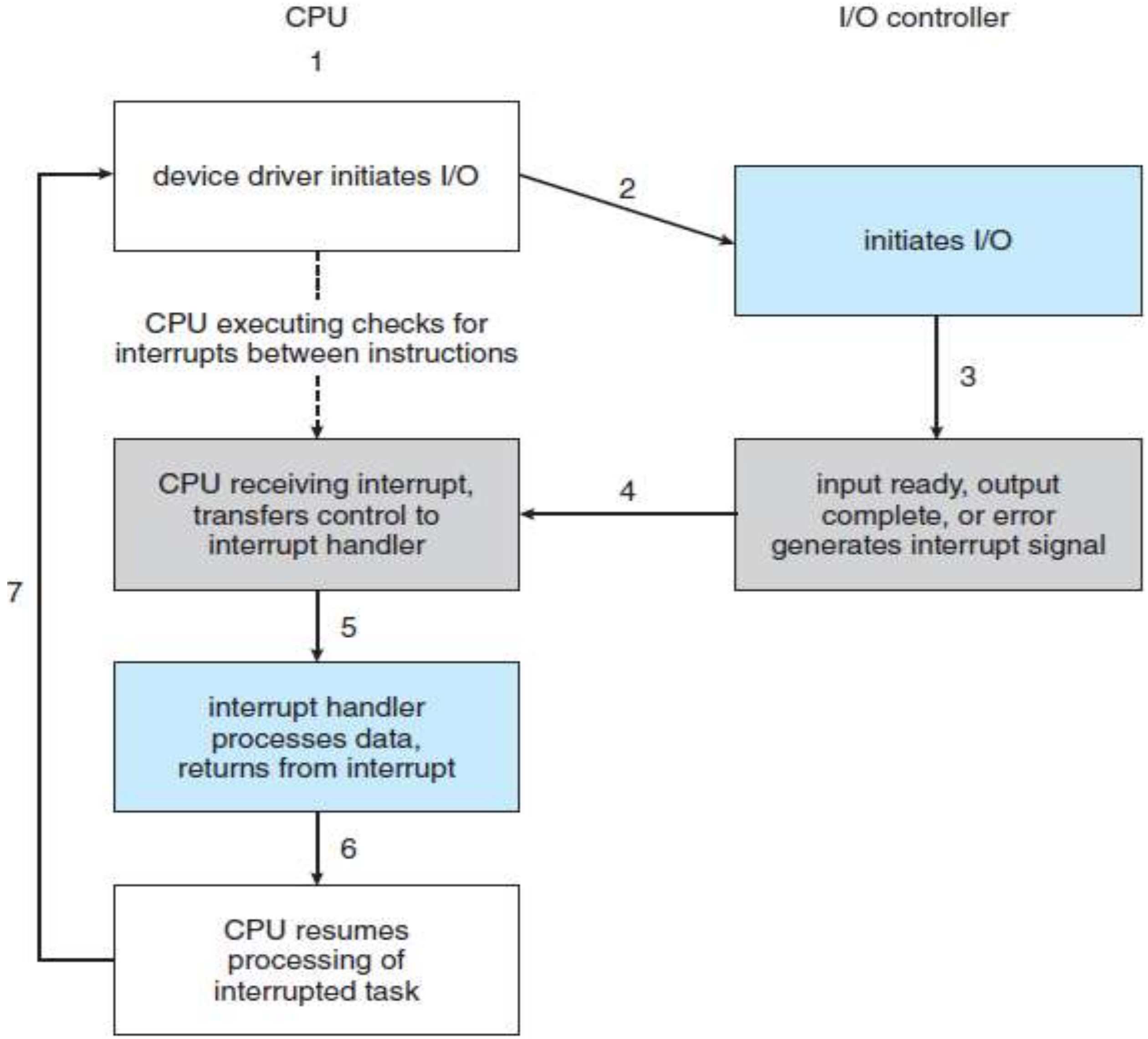
The device driver then returns the data if the operation was a read.

For other operations, the device driver returns status information such as “write completed successfully” or “device busy”.

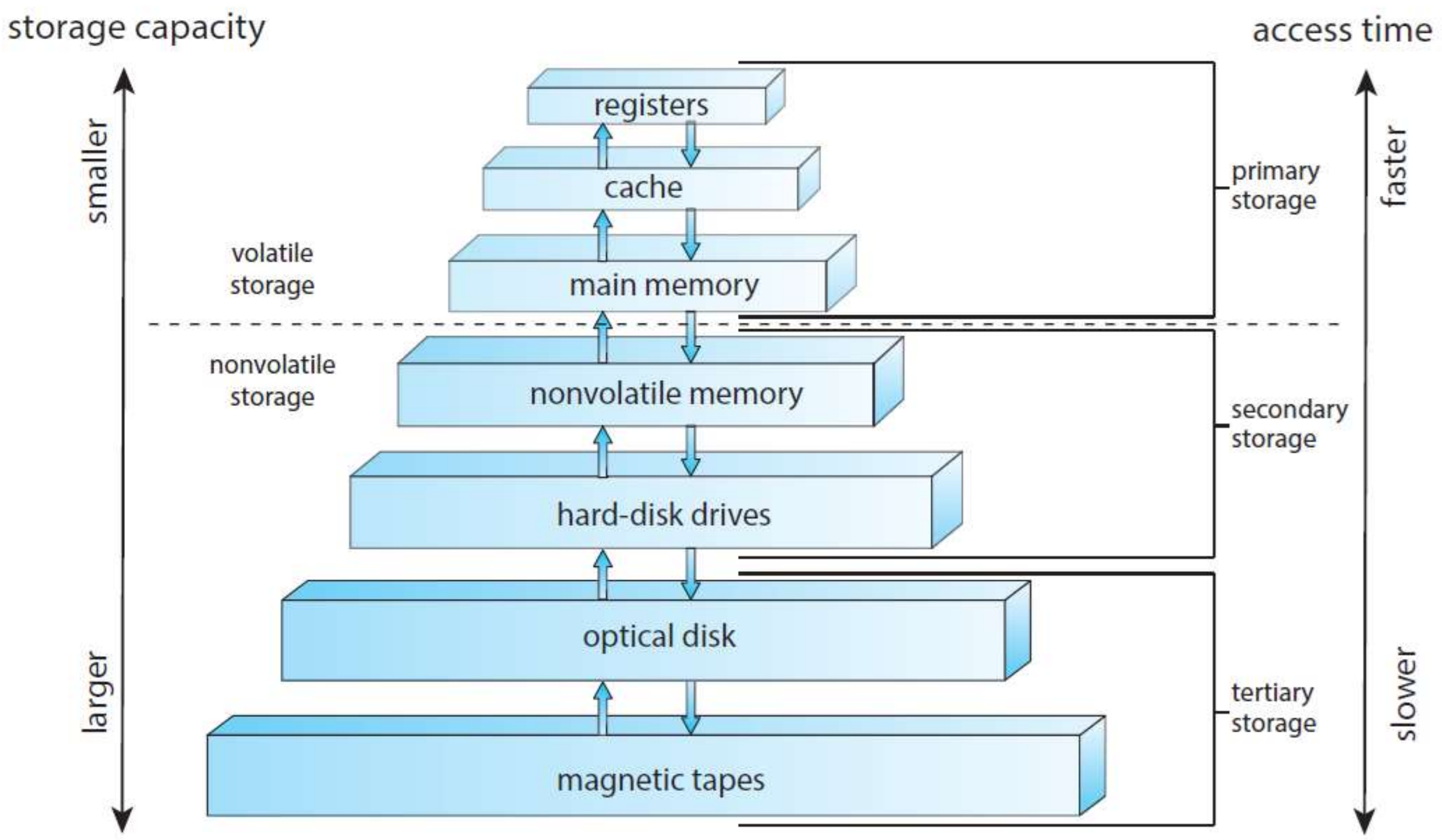
When an interrupt is raised, the CPU stops what it is doing and immediately transfers execution to the interrupt service routine.

The interrupt service routine executes; on completion, the CPU resumes the interrupted computation.

Following figure summarizes the interrupt-driven I/O cycle.



Storage Structure



Registers

Registers are small, high-speed memory units located in the CPU.

They are used to store the most frequently used data and instructions.

Registers have the fastest access time and the smallest storage capacity, typically ranging from 16 to 64 bits.

Cache Memory

Cache memory is a small, fast memory unit located close to the CPU.

It stores frequently used data and instructions that have been recently accessed from the main memory.

Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

Main Memory

Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system.

It has a larger storage capacity than cache memory, but it is slower.

Main memory is used to store data and instructions that are currently in use by the CPU.

Secondary Storage

Secondary storage, such as hard disk drives (HDD) and solid-state drives (SSD), is a non-volatile memory unit that has a larger storage capacity than main memory.

It is used to store data and instructions that are not currently in use by the CPU.

Secondary storage has the slowest access time and is typically the least expensive type of memory in the memory hierarchy.

Magnetic Disk

Magnetic disks are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material.

The Magnetic disks work at a high speed inside the computer and these are frequently used.

Magnetic Tape

Magnetic tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backup of data.

In the case of a magnetic tape, the access time for a computer is a little slower and therefore, it requires some amount of time for accessing the strip.

Operating System Operations

Following are the major operations or functions of an operating system

1. Process Management
2. Memory Management
3. File Management
4. I/O device Management
5. Protection
6. Security
7. Networking

Process Management

Process is a program in execution state.

To mange processes, the Operating System performs the following activities

1) creating and deleting processes
2) suspending and resuming processes
3) providing synchronization
4) providing communication
5) handling deadlocks
6) CPU scheduling

Creating and deleting processes

To create a new process, the operating system has to do the following activities

1) Move the program from secondary memory (HD) to primary memory (RAM)
2) Allocate CPU or processor to the program in primary memory

To delete a process, the Operating System has to move the program form primary memory (RAM) to secondary memory (HD).

Suspending and resuming processes

Suspending a process is temporarily stopping the execution of process due to several reasons like requesting for the input from the user.

Before suspending the process, the Operating System has to save the status of the execution of the process.

Resuming a suspended process is restarting the execution of the suspended process when the process becomes ready for execution.

Before resuming the process, the Operating System has to restore the saved status of the process.

Providing synchronization

A resource that can be shared by a number of processes at a time is called Sharable resource.

Ex: RAM, File etc.

A resource that can be used by only one process at a time is called Non-sharable resource.

Ex: CPU, printer.

When a number of processes requests for a non-sharable resource at the same time, the Operating System has to provide synchronous access to the resource by allocating the resource to only one process at a time.

Providing communication

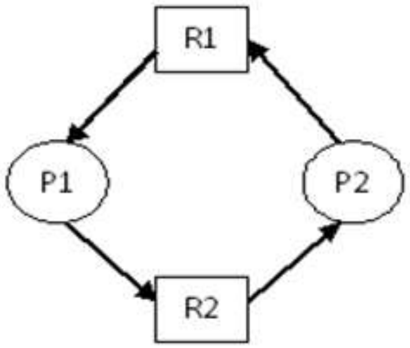
Programs or processes communicate for sharing or accessing data.

Two methods used by operating system for providing communication between processes are:

1) Shared memory
2) Message passing

Handling deadlocks

A set of processes is said to be in deadlock state if each process in the set is waiting for another process in the same set of processes.



The Operating System has to allocate the resources to processes such that the system should not go into the deadlock state.

If a deadlock state occurs in the computer system, then the operating system has to detect the occurrence of deadlock state and recover the computer system from deadlock state.

CPU Scheduling

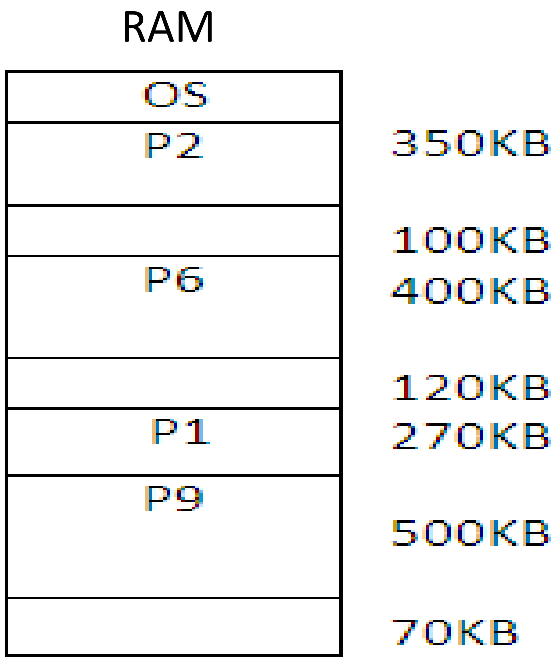
When a number of programs are ready for execution, the operating system decides an order for executing the program.

Memory Management

Operating system has to manage both primary memory (RAM) and secondary memory (Hard disk).

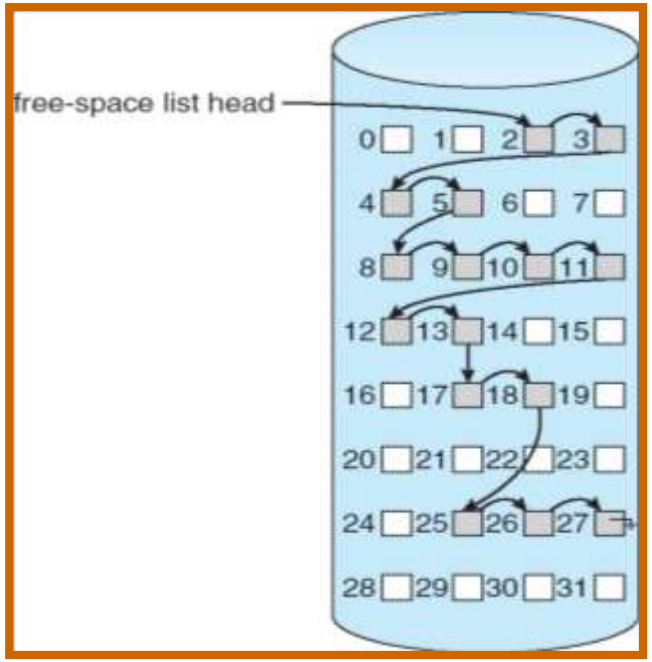
For managing the primary memory, the Operating System is responsible for

1) Knowing which parts of the RAM are in use and by whom
2) Allocating & de-allocating space when required
3) Deciding which process needs to be moved between RAM and Hard disk



The hard disk contains number of storage blocks.

At any time, some of the blocks contains programs or data and remaining blocks are free.



For managing the secondary memory, the Operating System is responsible for

1) Free – space management
2) Storage allocation
3) Disk scheduling

File Management

File is a logical storage unit.

The Operating System maps files onto storage devices.

Files are organized into directories to make them easier to use.

The Operating System is responsible for the following activities for managing files

1) Creating and deleting files
2) Creating and deleting directories
3) Supporting primitives for manipulating files and directories
4) Backing up files on stable storage

I/O device Management

A computer system contains a number of input and output devices like keyboard, mouse, monitor, printer and so on.

Each device is controlled by a device driver.

The Operating System has to manage all these device drivers.

Networking

A network is a collection of systems connected to each other.

The Operating System used in a computer which is connected to a network must provide support for

1) Sharing the resources available in the network.
2) Dividing the program into parts and distributing them to different systems in the network.
3) Providing transparency to the users.

Protection and Security

Protection: restricting access to the resources of the system from the processes that are running in the system.

For example, if two or more processes request the CPU at the same time then the Operating System has to restrict the access to the CPU by allocating the CPU to only one process at a time.

Security: restricting access to the system by the users.

Username and password are used to provide security.

Types of Operating Systems

1) Batch processing operating system
2) Multiprogramming operating system
3) Timesharing operating system
4) Distributed operating system
5) Real-time operating system
6) Embedded operating system
7) Mobile operating system

Batch processing operating system

If the computer system is running with batch processing operating system then the execution of programs in the computer system is as follows:

The programs that users want to execute and the input data required for executing the programs are collected into a batch.

The batch of programs is then loaded into computer system for execution.

The batch of programs is then executed in sequential manner.

There is no interaction between users and computer during execution of the programs.

The outputs generated after execution of programs is collected into a batch and then distributed to users.

IBM’s Z/OS is an example Batch operating system.

Multiprogramming operating system

When only one program is loaded into main memory and if that program requires any i/o operation during its execution then the CPU will be in idle state until the i/o operation is completed.

This leads to less utilization of resources (CPU, i/o devices).

To increase the utilization of resources, a number of programs is loaded into main memory.

During execution of a program, if the program requires any i/o operation then the CPU is switched to another program so that the CPU is busy at all times.

Loading a number of programs at a time into main memory is called **multiprogramming.**

Multiprogramming operating system allows loading of a number of programs at a time into RAM.

Multiprogramming operating system switches the CPU from current program to another program when the current process is completed or goes to the waiting state.

Windows and LINUX are examples for multiprogramming operating systems.

Timesharing operating system

Timesharing operating system loads a number of programs at a time into main memory.

Allow each program to execute for a certain amount of time only. After that the CPU is switched to another program.

Each program has equal chance of getting the CPU.

Timesharing operating system switches the CPU from current program to another program in the following cases:

1) when the current process is completed
2) when the current process goes to the waiting state
3) when the allocated time slice is over.

The user can interact with his program while it is running.

A time-shared operating system allows many users to share the computer simultaneously.

Windows and LINUX are examples for timesharing operating systems.

Distributed operating system

Distributed operating system manages a group of independent computers and makes them appear to be a single computer.

The group of computers is connected through a network.

Following are the features of distributed operating systems:

<u>Resource sharing</u>

Resources can be shared by the computer systems connected to the network.

For example, if a printer is connected to the network then all systems in the network can share the printer.

<u>Reliability of resources</u>

If any resource of any system fails then the resource of other system in the network can be used.

<u>Speed up of computations</u>

Programs can be executed in less time by dividing the program into number of parts and executing these parts on different systems in the network.

Communication between systems

If a number of persons is involved in the development of any project then there should be some communication between the systems that are being used by the persons.

This communication can be provided easily by the distributed operating system.

Providing Transparency

The distributed operating system hides the details of execution of the program from the user.

Amoeba and LOCUS are examples for distributed operating systems.

Real time operating system

Real time operating systems are used in computer systems which executes real time applications.

Real time applications have fixed time constraints on processing.

For example, if there is a satellite which sends some data for every 100 seconds and if a computer system receives and stores that data then the operating system used in that computer system is a real time operating system.

In this example, the operating system has to receive and store the data within 99 seconds.

Windows CE and Symbian are examples for real time operating systems.

Embedded operating system

The operating systems designed for being used in embedded computer systems are known as embedded operating systems.

Embedded operating systems are designed to operate on small machines like PDAs with less autonomy.

Embedded operating systems are able to operate with a limited number of resources.

Embedded operating systems are very compact and extremely efficient.

Windows CE and FreeBSD are some examples of embedded operating systems.

Mobile operating system

A mobile operating system controls a mobile device and its design supports wireless communication and mobile applications.

Tablet PCs and smart phones run on mobile operating systems.

Blackberry OS, Google's Android and Apple's iOS are some of the most known names of mobile operating systems.

System calls

The operating system provides services to the user programs through system calls.

System calls are generally developed in C and C++ languages.

Some system calls which directly interacts with hardware devices are developed in assembly language.

System calls are executed in kernel of operating system when a user program requests the operating system for any service.

Kernel is the major part in the operating system (like a heart in human body).

The programmers develop programs using Application Programming Interface (API).

The API contains set of functions/methods using which the programmer develops programs.

Each function or method in the API is linked to a system call.

When any API method is executed in the program then the corresponding system call is invoked.

For example, when a user program calls a scanf() function for reading data from keyboard then the system call 'read()' is invoked and executed in the kernel of the operating system.

There are two main reasons for writing programs using API instead of system calls directly.

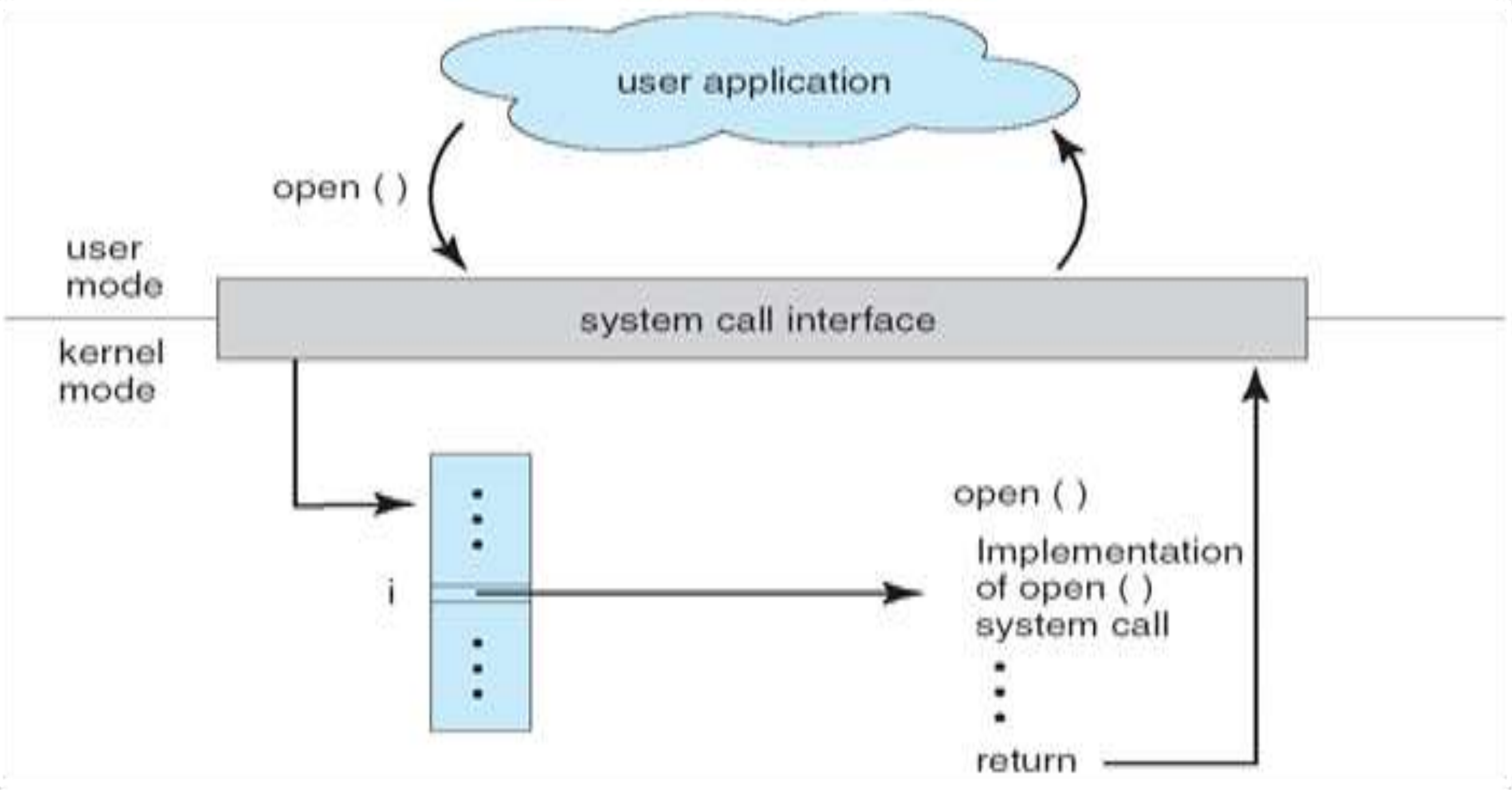
1) Program written using API can be executed on any system that supports the same API.
2) Working with system calls is more difficult than API.

System call interface

The run-time support system for most programming languages provides a system-call interface that serves as the link to system calls made available by the operating system.

During execution of user program, when any API function is executed then the system-call interface identifies the corresponding system call and activates that system call within the operating system.

The system-call interface also returns the status of the system call and any return values.


user application
open ( )
user mode
system call interface
kernel mode
i
open ( )
Implementation of open ( ) system call
return

#include <stdio.h>
int main ( )
{
printf ("Greetings");
return 0;
}
user
mode
standard C library
ernel
mode
write ( )
write ( )
system call

Types of system calls

1) Process control system calls
2) File management system calls
3) Device management system calls
4) Information maintenance system calls
5) Communication maintenance system calls
6) Protection and security maintenance system calls

Process control system calls

The various system calls for controlling processes are

1. end, abort
2. load, execute
3. create process, terminate process
4. get process attributes , set process attributes
5. wait, signal

File management system calls

1. create, delete
2. open, close
3. read, write, reposition
4. get file attributes, set file attributes

Device management system calls

1. request device, release device
2. read, write
3. get attributes, set attributes

Information maintenance system calls

1. get time or date, set time or date
2. get system data, set system data

Communication maintenance system calls

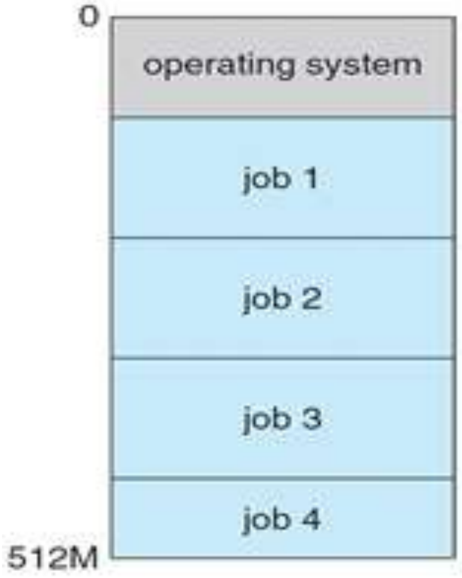
1. open connection, close connection
2. get hostid, get processid
3. send message, receive message
4. shared memory create, shared memory attach

protection

1. set permission, get permission
2. allow user, deny user

Dual mode operation of the computer system

A modern computer system operates in 2 modes in order to protect the operating system code from user processes and also the code of each process from other processes.



The two modes are:

1. User mode
2. Kernel mode (supervisor mode, privileged mode or system mode)

A bit called 'mode bit' indicates the current mode.

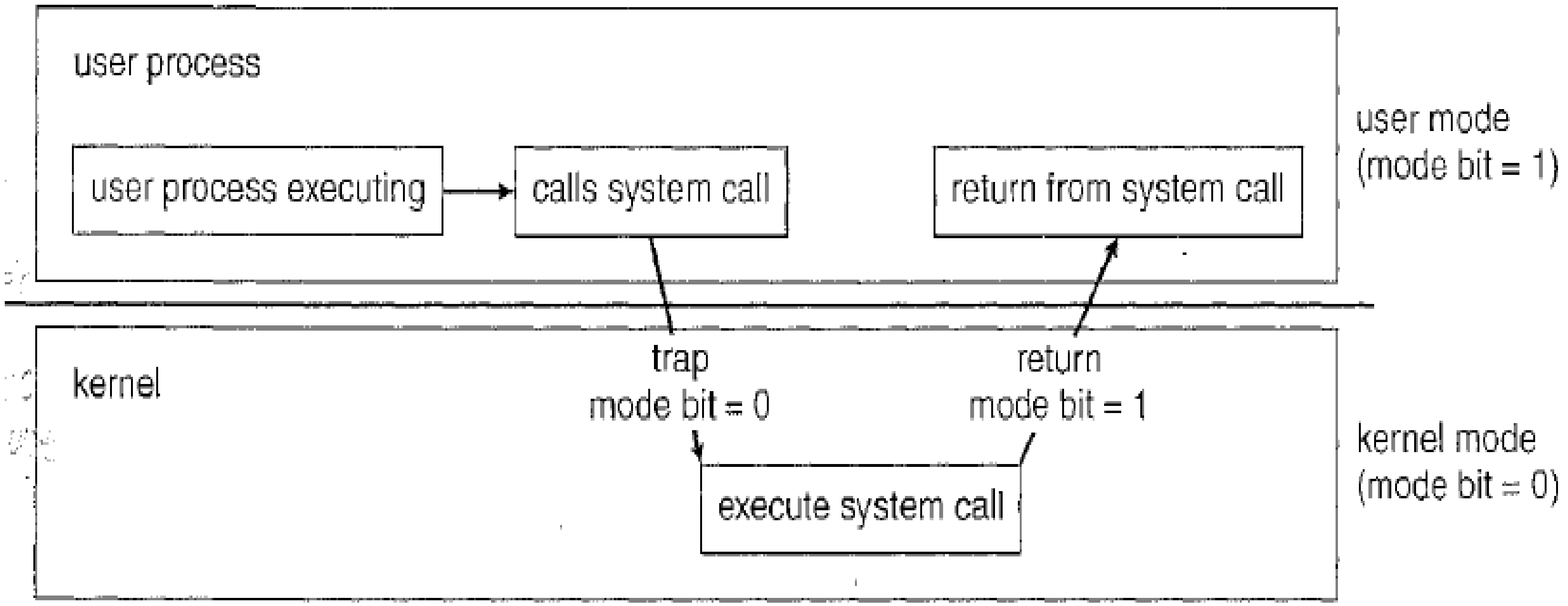
The value of mode bit is '0' for kernel mode and '1' for user mode.

When the system is booted, the system is running in kernel mode.

When the operating system starts any user program then the mode of system is switched to user mode.

During execution, if the user program requests any service from the operating system then the mode is changed to kernel mode.

After executing the system call, the mode is switched to user mode by setting mode bit to ‘1’ before passing the control to user program.



To protect the operating system code, some of the machine instructions are designated as privileged instructions.

These privileged instructions are executed only in kernel mode.

The instruction used to change the mode bit is an example for privileged instruction.

When any privileged instruction is executed in user mode then the hardware informs to the operating system.

Virtualization

Virtualization is a technology that allows us to abstract the hardware of a single computer into several different execution environments, thereby creating the illusion that each separate environment is running on its own private computer.

These environments can be viewed as different individual operating systems that may be running at the same time and may interact with each other.

A user of a **virtual machine** can switch among the various operating systems.

Virtualization allows operating systems to run as applications within other operating systems.

With virtualization, an operating system that is natively compiled for a particular CPU architecture runs within another operating system also native to that CPU.

VMware created a new virtualization technology in the form of an application that ran on Windows.

That application ran one or more **guest** copies of Windows or other native x86 operating systems, each running its own applications. (See Figure 1.16.)

Windows was the **host** operating system, and the VMware application was the **virtual machine manager** (**VMM**).

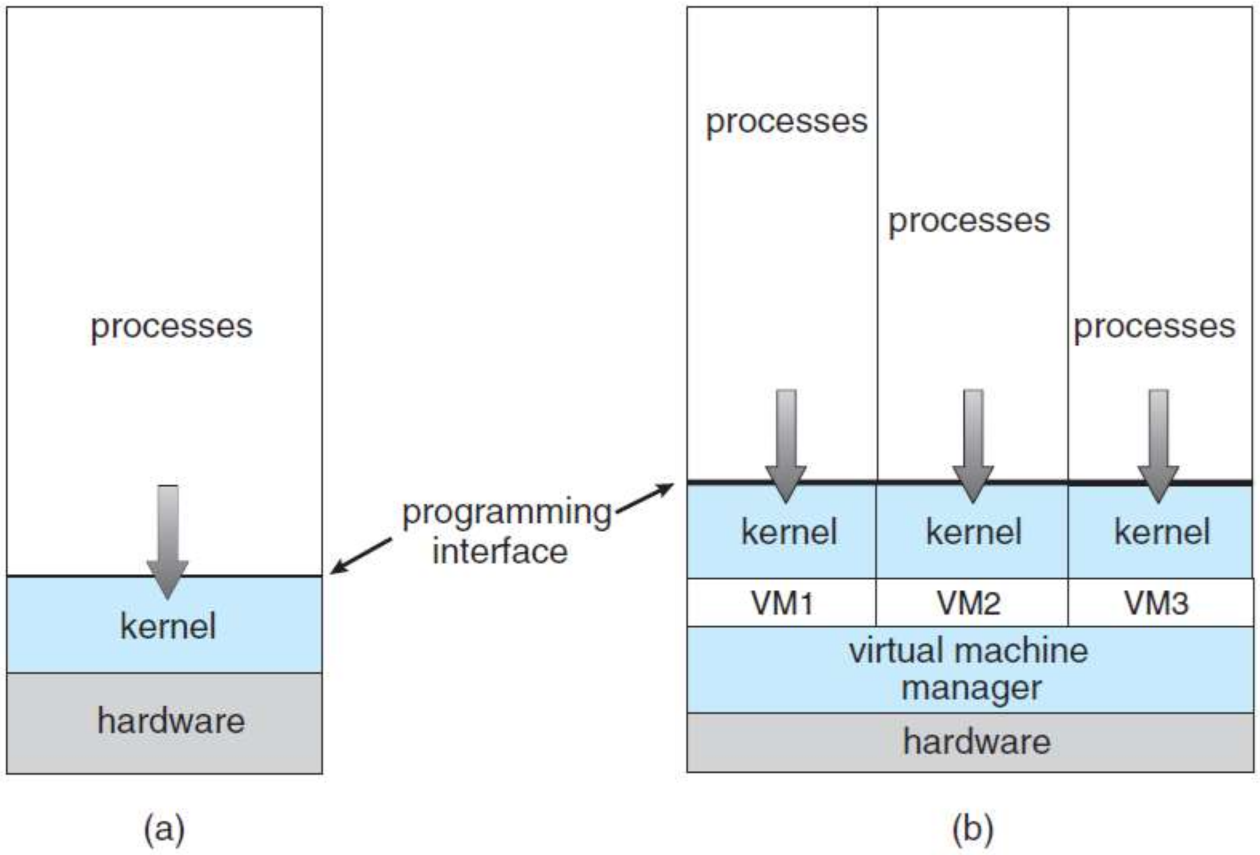


Figure 1.16 A computer running (a) a single operating system and (b) three virtual machines.

The VMM runs the guest operating systems, manages their resource use, and protects each guest from the others.

On laptops and desktops, a VMM allows the user to install multiple operating systems to run applications written for operating systems other than the native host.

For example, an Apple laptop running macOS on the x86 CPU can run a Windows 10 guest to allow execution of Windows applications.