UNIT-I

OPERATING SYSTEMS

→ Overview of Computer operating Systems:

- An operating system is a program that manages the computer hardware and provides a basis for application programs and acts as an interface between the user and computer hardware.
- Operating system execute user programs and make the computer system convenient.
- An abstract view of computer system components:

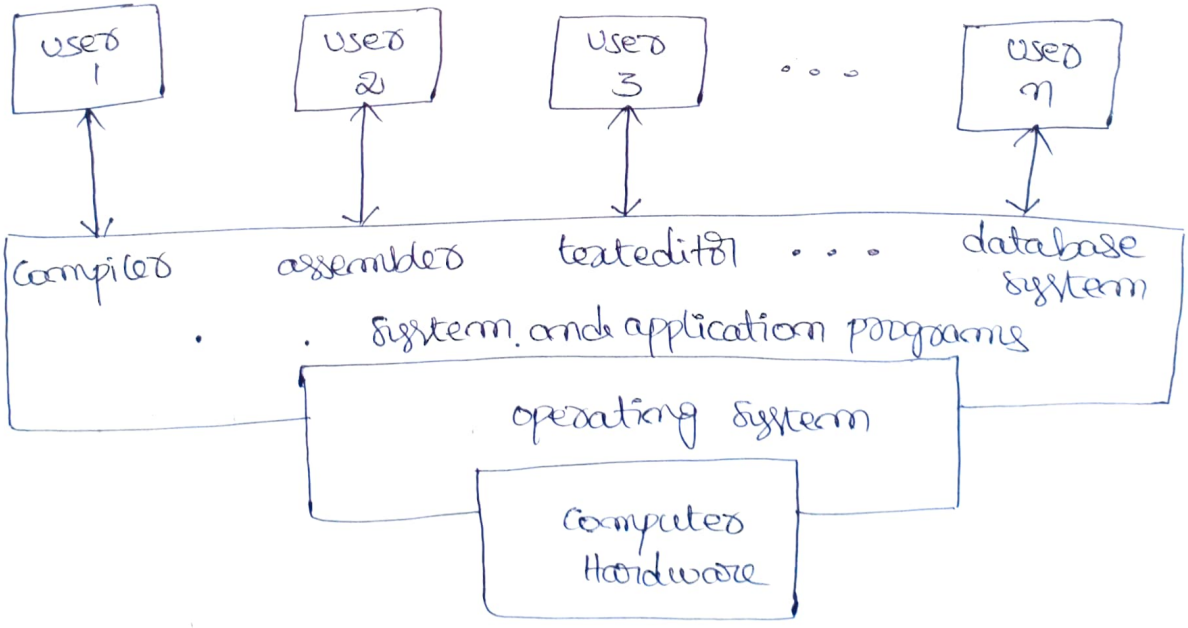
Computer system can be broadly divided into four components:

① Hardware provides basic computing resources such as I/O devices, CPU, memory etc.

② operating system which controls and coordinates the h/w.

③ Application programs provides the way in which the system resources are used to solve the computing problems of users.

④ Application programs users



→ Computer System organization:

- A general purpose computer system consists of one & more CPUs and a number of device controllers connected through

a common bus that provides access to shared memory.

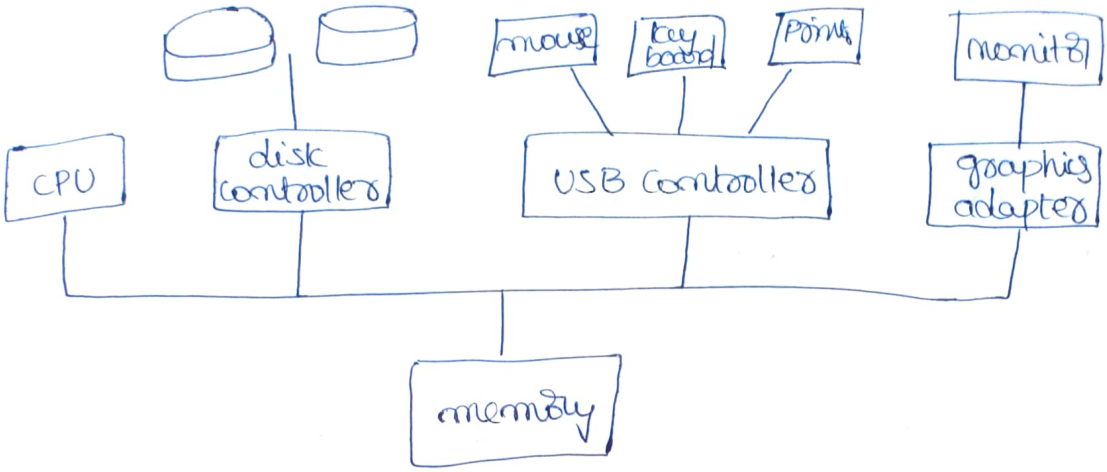


fig: A modern Computer System.

- The CPU and the device controllers can execute concurrently competing for memory cycles.
- Bootstrap program: When a computer is powered up or rebooted an initial program to run which is stored in read-only memory and known as bootstrap program. The boot-strap program must load the operating system kernel into main memory and start executing the first process known as init and waits for some event to occur.
- Interrupts: Interrupts are an important part of computer architecture. Each computer design has its own interrupt mechanism but several functions are common. These are hardware interrupts and software interrupts. These interrupts must transfer control to interrupt service routine generally

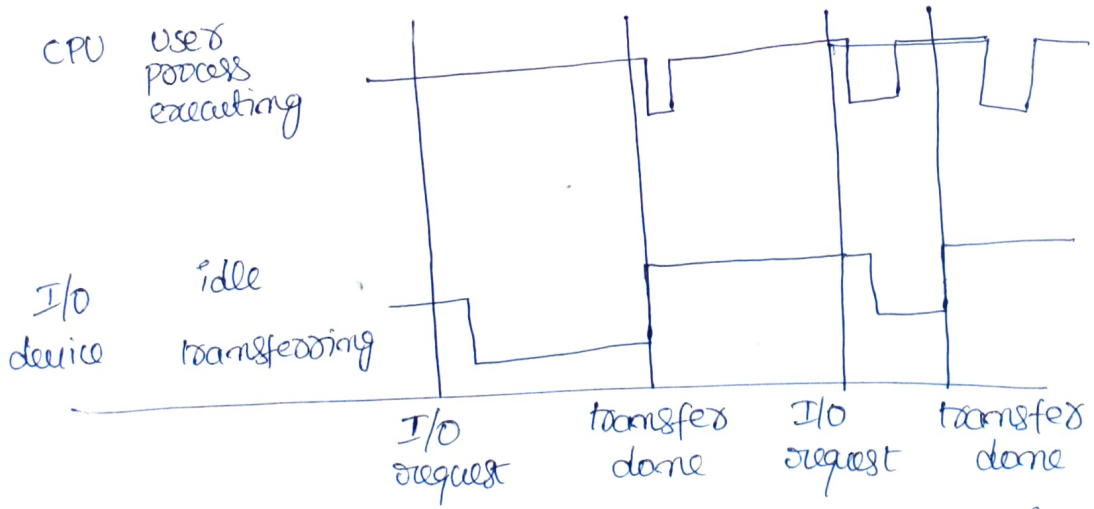


fig: Interrupt time line for a single process

through interrupt vector, which contains the addresses of all the service routines. Interrupt architecture must save the address of the interrupted instruction. Incoming interrupts are disabled while another interrupt is being processed.

- A trap is a software-generated interrupt caused either by an error or a user request.
- The operating system is interrupt-driven. The os preserves the state of the CPU by storing it into the registers. It determines what action to be taken for each type of interrup

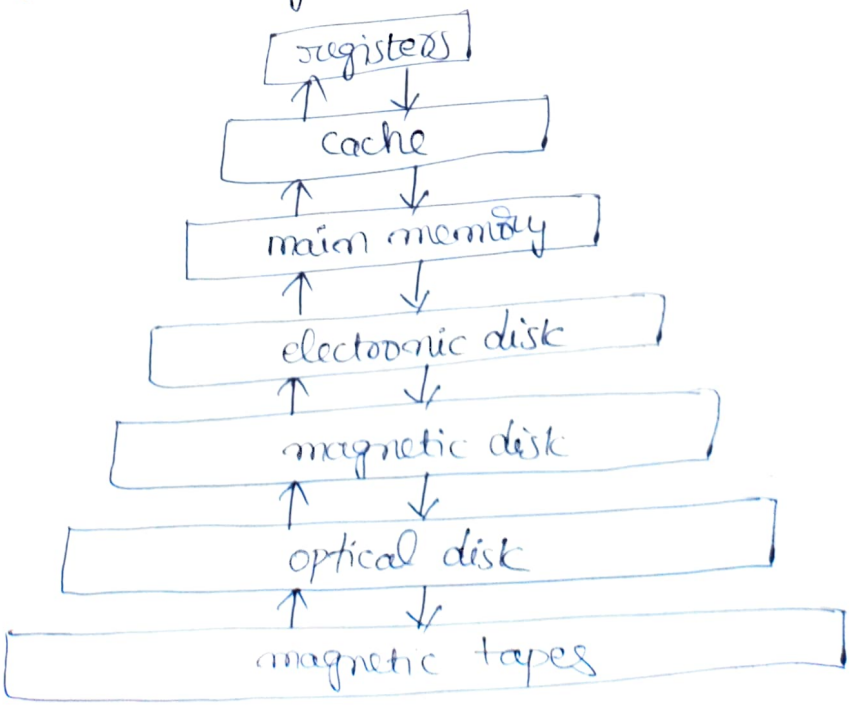
- I/O structure:

- After I/O starts controls returns to user program only upon I/O completion. wait instruction idles the CPU until the next interrupt.
- storage structure: @main memory

Computer programs must be in main memory to be executed and the processor directly access it. It is called as Random access memory which forms an array of words. Each word has its own address. main memory is a volatile memory

(b) Secondary storage device is an extension of main memory that provides nonvolatile storage capacity.

(c) magnetic disks are rigid metal or glass platters covered with magnetic recording material.



(d) The cache memory stores data for future requests so that the data can be served faster and is temporary storage area

(e) A register is one of the small storage devices which holds a computer instruction. These are the fastest storage devices

(f) An optical disc is an electronic data storage medium that can be written to and read using a laser beam.

(g) Magnetic tape is a medium for magnetic recording made of magnetizable coating on a long, narrow strip of plastic film

→ Computer-System Architecture:

- A Computer system may be organized in a number of different ways, such as categorized according to the number of processors used:

(1) Single-Processor Systems: on a single-processor system, there is one CPU capable of executing a general-purpose instruction set, including instructions from user processes. Allmost all systems have other special-purpose processors in the form of device-specific processors, such as disk, keyboard and graphics controllers. All of these special-purpose processors run on a limited instruction set and do not run user processes.

(2) Multiprocessor Systems: (Tightly coupled systems)

- Multiprocessor systems also known as parallel systems contains two or more processors in close communication, sharing the computer bus and clock, memory and peripheral devices. Multiprocessor systems have the following advantages

(a) Increased throughput: By increasing the number of processors more work is done in less time.

(b) Economy of scale: multiprocessor systems can cost less than equivalent multiple single processor systems because they can share peripherals, mass storage and power supplies.

(c) Increased reliability: If functions can be distributed properly among several processors then the failure of one

processor will not halt the system but only slow it down.
- The multiprocessor systems are two types:

(i) Asymmetric multiprocessing: In this each processor is assigned a special task and a master processor controls the system, + other processors either look to the master for instructions or have predefined tasks. This scheme defines a master-slave relationship.

(ii) Symmetric multiprocessing: In this each processor performs all tasks within the operating system. All the processors are peers, no master-slave relationship exists.

Eg: A Solaris system can be configured to employ dozens of processors, all running Solaris.

(iii) Clustered systems: Another type of multiple-CPU systems is the clustered system, in which multiple CPUs gathered to accomplish a computational task. clustered systems are composed of two or more individual systems coupled together, via a local-area network. clustering is usually used to provide high-availability i.e service is still provided if one or more systems in the cluster fails.

→ Operating system Functions:

- An OS is a program that controls the execution of application programs and acts as an interface between applications and the computer hardware. Three objectives of operating system are as:

(a) Convenience: An OS makes a computer more convenience to use.

(b) Efficiency: An OS allows the computer system resources to be used in an efficient manner.

(c) Ability to evolve: An OS should be designed in a way to permit the effective development, testing and introducing new system functions.

- Three main functions of an OS are as follows:

① The operating System as a User/Computer Interface:

- The hardware and software used in providing application to a user can be viewed as hierarchical structure as

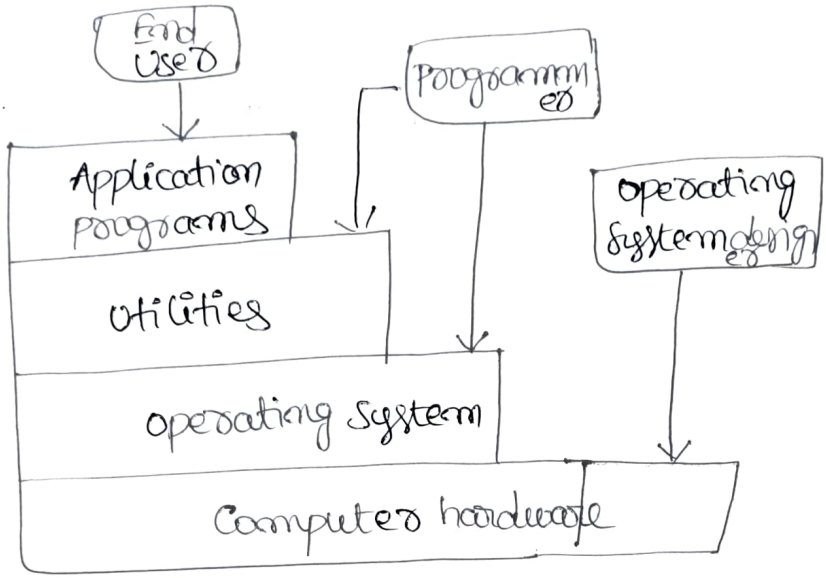


fig: Layers and views of a Computer System

- The end user is not concerned with the details of hardware and views a computer in terms of a set of applications.
- A programmer will use of utilities in developing an application. OS masks the details of the hardware from the programmer and provides a way to use it. The OS provides various services as follows:

(a) Program development: The OS provides a variety of facilities and services such as editors and debuggers to assist the programmer in creating programs.

(b) Program Execution: A number of steps to be needed to execute a program. Instructions and data must be loaded into main memory, I/O devices and files must be initialized

(c) Access to I/O devices: The OS provides a uniform interface that hides I/O control signals so that programmers can access such devices using simple read and writes.

(d) Controlled access to files: For file access the OS must reflect the detailed structure of the data in the files.

(e) System access: For shared systems, the OS controls access to the system as a whole and to specific system resources.

(f) Error detection and response: A variety of errors can occur while a computer is running. These include internal and external hardware errors and software errors. In each case the OS must provide a response that clears the error condition with the least impact on running applications.

(g) Accounting: A good OS will collect usage statistics for various resources and monitor performance parameters such as response time.

(2) The operating System as Resource manager:

- A computer is a set of resources and is responsible for managing these resources. The OS directs the processor in the use of the other system resources and in the timing of its execution of other programs.
- A portion of the OS is in main memory and it includes kernel or nucleus which contains most frequently used functions in the OS. The remainder of main memory contains user programs and data.

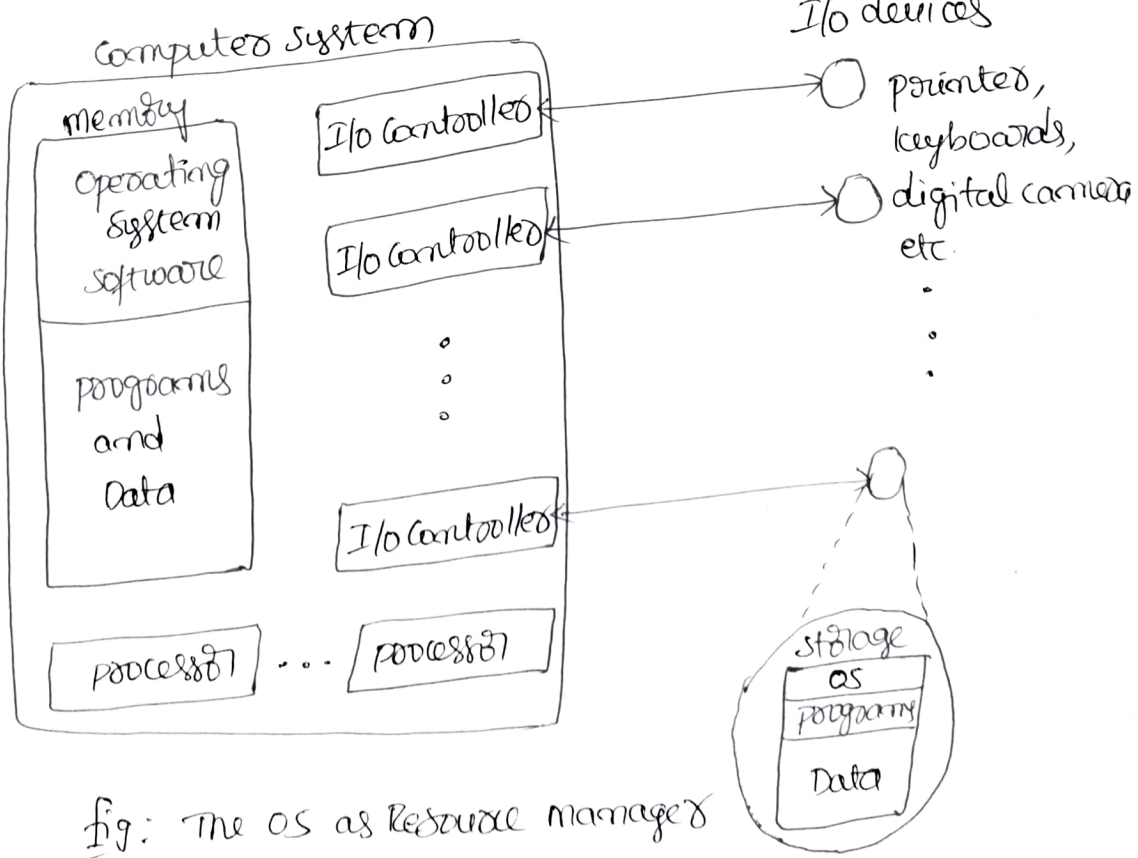


fig: The OS as Resource manager

- The allocation of main memory is controlled by both the OS and memory management hardware in processor.

(5) Ease of Evolution of an operating System:

- A major operating system will evolve over time for the following reasons:

(i) Hardware Upgrades: The OS should adopt to the new types of hardwares as early versions of UNIX OS did't employ a paging mechanism because they were run on processors without paging hardware.

(ii) New services: In response to user demands the OS expand to offer new services.

(iii) Fixes: Any OS has faults. These are discovered over the course of time and fixes are made.

→ Operating System (operations / functionalities) Components

- operating system perform the following functions such as:

(i) Process management:

- A program in execution is a process. A time-shared user program such as a compiler is a process, A word-processing program being run by an user is a process.
- A process needs resources such as CPU time, memory, files, and I/O devices to accomplish its task.
- The OS responsible for the following activities
  - (a) Creating and deleting both user and system processes.
  - (b) suspending and resuming processes.
  - (c) Process synchronization mechanism
  - (d) Providing deadlock handling mechanisms.
  - (e) Providing process communication mechanisms.

(ii) Memory management:

- Main memory is a large array of words & bytes and each word & byte has its own address. The processor reads

instructions from main memory during instruction fetch cycle and both reads and writes data from memory during the data fetch cycle. The OS is responsible for the following activities:

(a) Keeping track of which parts of memory are currently used.
(b) Deciding which processes and data to move into and out of memory.
(c) Allocating and deallocating memory space.

(iii) Storage management:

- To make the computer system convenient for users, the OS provides a uniform, logical view of information storage. The OS maps files onto physical media and accesses these files via storage devices.

(a) File-System management: File management is one of the most visible components of an operating system. Computers store information on magnetic disks, optical disks and magnetic tapes and physical organization. A file is a collection of related information. The OS is responsible for the following activities

1. Creating and deleting files.
2. Creating and deleting directories to organize files.
3. Supporting primitives for manipulating files & directories
4. Mapping files onto secondary storage.
5. Backing up files on stable storage media.

(b) mass-storage management: Main memory is too small to accomodate all data and programs and so computer systems use disks as the principal on-line storage medium for both programs and data. Most programs such as compilers, assemblers, word processors, editors and formatters stored on a disk. The OS is responsible for the following activities

1. Free space management.

(c) Caching: The frequently used information is kept in some main memory area on a temporary basic for faster access which is known as cache memory. When a user needs that information processor checks first in cache, if it is not use the information from the source. Internal programmable registers such as index registers provide a high speed cache for main memory.

(iv) I/O system management:

- One of the major purposes of an os is to hide the specific hardware devices from the user. The I/O sub system consists of several components:

1. A memory management component that includes buffering, caching and spooling.
2. A general device driver interface.
3. Drivers for specific hardware devices.

→ Protection and security:

- If a computer system has multiple users and allow the concurrent execution of multiple processes. For that purpose files, memory segments, CPU and other resources can be operated by those processes that have authorization from the operating system.
- Protection is a mechanism for controlling the access of processes & users to the resources defined by computer system. Protection can improve reliability by detecting errors at the interfaces between component subsystems.
- A computer system can have adequate protection but still be prone to failure so security defend a system from external and internal attacks. The attacks include viruses worms, denial-of-service attacks, identity theft etc.
- Most of the systems maintain a list of users and associated user identifiers. In windows NT this is a security ID (SID). These numerical IDs are unique, one per user.

when a user logs in to the system, the authentication determines the appropriate user ID for the user. That user ID is associated with all of the other user's processes and threads. In some cases rather than individual user ID we define group of users. i.e group ID.

Eg: In UNIX system may allow to define group of users.

- A user can be in one or more groups, depending the OS.

→ Distributed systems:

- A distributed system is a collection of physically separate, heterogeneous computer systems that are networked to provide the users with access to the various resources. Access to the shared resources will increase the computation speed, functionality, data availability and reliability etc.
- A network 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 transport medium. TCP/IP is the most common network protocol used by UNIX, windows.
- Networks are categorized based on the distances between their nodes.

(a) A local-area network (LAN) connects computers within a room or floor or a building.

(b) A wide-area network (WAN) links buildings, cities or countries

(c) A metropolitan-area network could link buildings within a city.

(d) A small area network would be wireless communication over a distance of several feet. Eg: Bluetooth

- The media include copper wires, fiber strands, and wireless transmissions between satellites, microwave dishes, radio
- The networks also vary in their performance and reliability.

- Some operating systems designed for networking such as a network operating system is an operating system that provides features such as file sharing, across the network and includes a communication scheme that allows to exchange messages.

→ Special-Purpose Systems:

- There are various special-purpose systems whose functions are more limited and objective is to deal with limited computation domains. They are as follows:

(i) Real-Time Embedded systems: Embedded computers are the most relevant form of computers such as car engines and manufacturing robots and microwave ovens etc. They tend to have very specific tasks. The systems they run on are primitive and so the operating systems provide limited features. Some general-purpose computers running standard operating systems with special-purpose applications.

- A real-time system requires that results be produced within a specified deadline period. There are two real time systems
  (a) A hard real time system guarantees that real time tasks be completed within their required deadlines.
  (b) A soft real time system provides priority of real time tasks over non-real time tasks.

- Embedded systems almost always run real-time operating systems which are known as Real-time Embedded systems. The characteristics are (a) Single purpose
  (b) Small size
  (c) Specific timing requirements.

- many real time systems are designed using system-on-a-chip strategy, which allows the CPU, memory management unit and attached peripheral ports.

- A real time system has well defined, fixed time constraints Processing must be done within the defined constraints.

(ii) Multimedia Systems:

- most operating systems are designed to handle data such as text files, programs, word-processing documents and spreadsheets etc. Recent technology includes multimedia data into computer systems.
- Multimedia data consists of audio and video files as well as conventional files, such as MP3 and MPEG files.
- Multimedia describes a wide range of applications such as MP3 DVD movies, video conferencing and short video clips
- Multimedia data must be accessed within specific timing requirements i.e video must be displayed at 24-30 frames per second.
- Multimedia data may be delivered to desktop PCs, PDAs, smart phones etc.
- Streaming is delivering a multimedia data file from a server to client over a Go network connection. There are 2 types

(a) Progressive streaming: the client begins playback of the multimedia data as it is delivered. The file is stored.

(b) Real-time streaming: the multimedia file is delivered to but not stored on the client's computer.

(iii) Handheld Systems:

- Handheld systems include personal digital assistants (PDAs) such as Palm and Pocket-PCs and cellular telephones many of these use special-purpose embedded operating systems. Developers of handheld systems and applications have many challenges due to the limited size of such devices. Eg: PDA is about 5 inches in height & 3 inches width
- most handheld devices have a small amount of memory, slow processors and small display screens.
- There are some issues of handheld system such as

(a) The amount of physical memory is between 512KB to 128MB As a result, the operating system and applications must use

memory efficiently. This include returning all allocated memory back to the memory manager when the memory is not being used.

(b) The speed of handheld devices is slow because processors used will run at a fraction of the speed. Faster processors require more power.

(c) The another issue is I/O i.e lack of physical space limits input methods to small keyboards, handwriting output options.

→ Operating System Services:

- An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of those programs.

(i) User Interface: Almost all operating systems have a user interface that have several forms. One is a command-line interface, which uses text commands and a method for entering them. Another is a batch interface, in which commands and directives to control those commands commonly a graphical user interface is used.

(ii) Program Execution: The system must be able to load a program into memory and run that program. The program may terminated normally or abnormally.

(iii) I/O operations: A running program may require I/O which may involve file or I/O device.

(iv) File System manipulation: The programs need to read and write files and directories. They also need to create and delete files and some programs include permission management to allow or deny access to files or directories

(v) Communication: There are some circumstances in which one process needs to execute others process information. Communications may be implemented via shared

memory or through message passing, in which packets are moved between processes by an operating system.

(vi) Error Detection: The operating system needs to aware of possible errors. Errors may occur in the CPU and memory hardware, in I/O devices and in the user program. For each type of error, the operating sho system should take an appropriate action to ensure correct and consistent computing.

(vii) Resource allocation: when there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them. many different types of resources are managed by operating system. There are many resources to allocate such as printers, modems, USB storage devices and other peripheral devices.

(viii) Accounting: we want to keep track of the resources accessed by the users so that providing more no. of units to accomplish all the tasks. operating system has to take care about performance parameters such as response time of a resource.

(ix) Protection and security: The owners of information stored in a multiuser or networked computer system want to control use of that information. when several processes execute concurrently then no process interfere into the others. Protection involves ensuring that all access to system resources is controlled. security ensures each user to authenticate to the system.

→ Operating System Structure:

A common approach to design a system is to partition the task into small components and each of these modules should be well-defined with defined inputs, outputs and functions. According to the modeling of these components into the kernel there are several operating system structures such as:

(i) Simple Structure: The computer systems do not have

well-defined structures mostly. Such operating systems started as simple, small and limited systems and then grew beyond their original scope.

Eg: MS-DOS is an example of a such system.

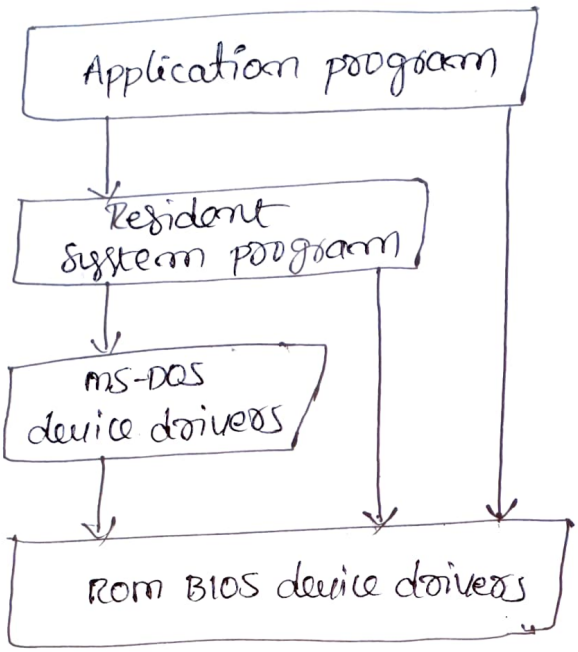


fig: MS-DOS layer structure

- In MS-DOS the interfaces and levels of functionality are not well-separated i.e application programs are able to access the basic I/O routines to write directly to the display and disk drivers.
- Another example is the original UNIX operating system. UNIX is initially limited by hardware functionality. It consists of two parts: the kernel
  the system programs.

The kernel is further separated into a set of interfaces and device drivers, which have been added and expanded over the years. The kernel provides the file system, CPU scheduling, memory management and os functions through system calls.

(ii) Layered Approach:

With proper hardware support, operating systems can be broken into pieces that are smaller and more appropriate. The

operating system can then retain much greater control over the computer and over the applications that make use of that computer. In top-down approach the overall functionality and features are determined and are separated into components.

- A system can be made modular in many ways, one such method is the layered approach.

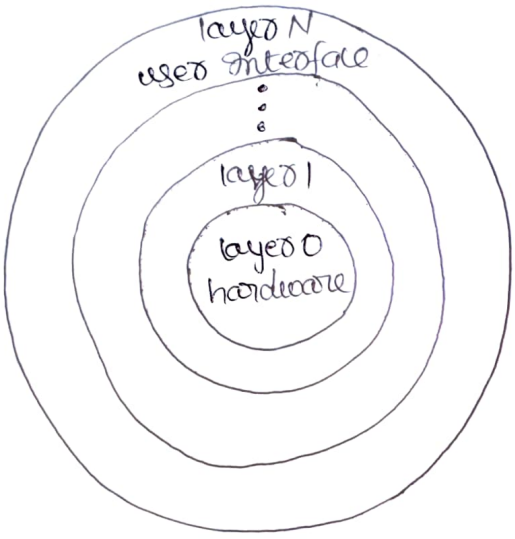


fig: A layered operating system

- In layered approach, the operating system is broken up into a number of layers. The bottom layer is the hardware, the highest layer is the user interface. An operating system layer is an implementation of an abstract object made up of data and operations that can manipulate those data.
- The advantage of the layered approach is simplicity of construction and debugging. The first layer can be debugged without any concern for the rest of the system, because it uses only the basic hardware. once the first layer is debugged, second layer is debugged and so on. If an error is found during the debugging of a particular layer, the error must be on that layer.
- The major difficulty with the layered approach involves defining the various layers appropriately.
- The layered approach is tend to be less efficient than other types.

(iii) Microkernels:

- In the mid-1980's an operating system called mach that modularized the kernel using the microkernel approach. This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs. The result is a smaller kernel. Microkernels provide minimal process and memory management and communication.
- The main function of the microkernel is to provide communication facility between the client program and the various services that are running in user space. Communication is provided by message passing i.e if the client program need to access a file, it must interact with the file server. The client program and service never interact directly and communicate indirectly by exchanging messages with the microkernel
- Adv: The microkernel approach is ease of extending the operating system. All new services are added to user space and consequently do not require modification of the kernel.

Eg: True64 UNIX & Digital UNIX has used the microkernel approach
QNX is a real time operating system also used this approach

- Disadv: The microkernels can suffer from performance decreases due to the increased system function overhead.

(iv) modules:

- The better methodology for operating system design involves using object-oriented programming techniques to create a modular kernel. In this the kernel has a set of core components and dynamically links during boot time & run time. Such a strategy implements in UNIX, Solaris, mac OS X.

Eg: The Solaris operating system structure is organized around a core kernel with seven loadable kernel modules

1. Scheduling classes
2. File systems
3. Loadable system calls

4. Executable formats
5. STREAMS modules
6. Miscellaneous
7. Device and bus drivers.

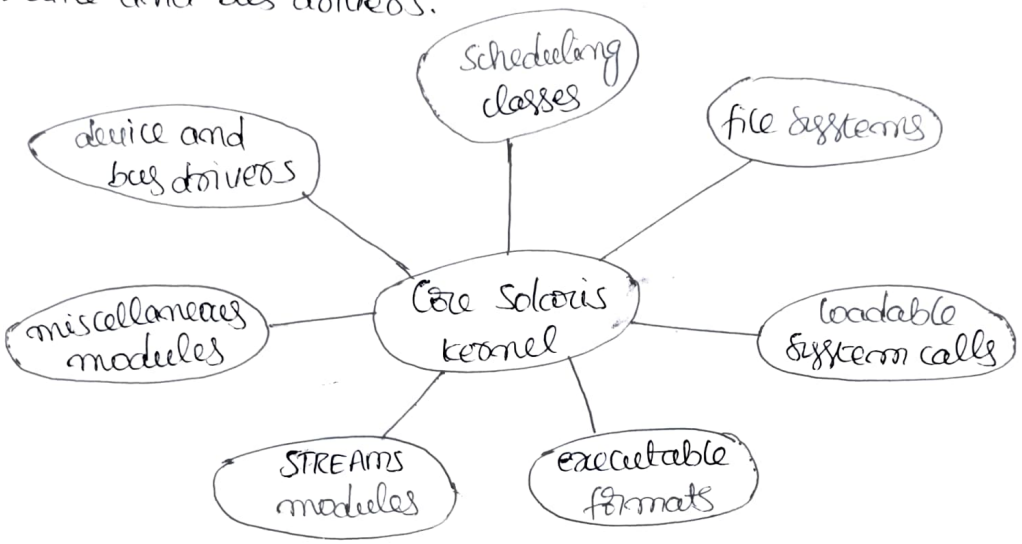


fig: Solaris loadable modules

- Such a design allows the kernel to provide core services that allows certain features to be implemented dynamically.

Eg: Device and bus drivers for specific hardware can be added to the kernel and support for different file systems can be added as loadable modules.

→ System calls:

- System calls provide an interface to the services made available by an operating system. These system calls are available as routines written C and C++ etc.

Eg: Consider a program to read data from one file and copy them to another file.

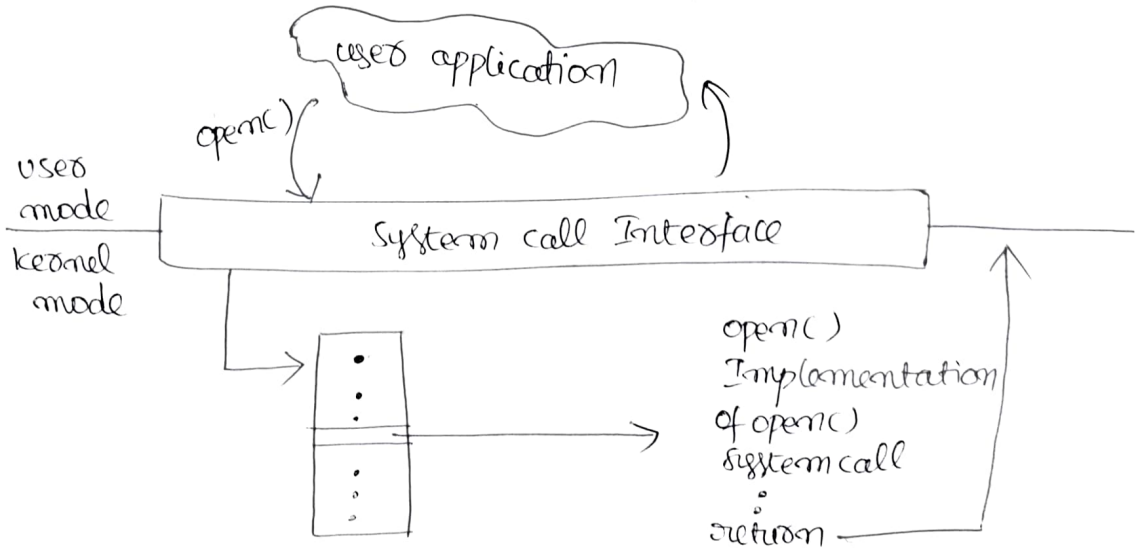
The following procedure to be followed:

Example system call sequence

Acquire input file name
    write prompt to screen
    Accept input
Acquire output File name
    write prompt to screen
    Accept input

```
open the input file
if file doesn't exist, abort
create output file
  if file exists, abort
loop
   Read from Input file
   write to output file
until read fails
close output file
write completion message
                on screen
Terminate normally
```

- Application developers design programs according to an application programming interface. The API specifies a set of functions that are available to an application programmer, including parameters that are passed to each function and the return values. There are three common APIs available as
  (i) Win32 API for windows systems
  (ii) POSIX API for POSIX based systems
  (iii) Java API for designing programs that run on the Java.
- Most programming languages provides a system-call interface that serves as the link to the system calls made available by the operating system.
- The system call interface maintains a table that contains the numbers associated with each system call.



- when a user program invokes a system call, it enters from user mode to kernel mode in which its operations executed. In the above the user program invokes an open() function/system call.

- Types of system calls: System calls can be grouped into five major categories: Process control
  file manipulation
  device manipulation
  information maintenance
  communication.

(i) Process Control: A running program needs to halt its execution either normally or abnormally i.e end or ~~halt~~abort. The system calls used to control the process execution are
  (a) end, abort
  (b) load, execute
  (c) create process, terminate process
  (d) get process attributes, set process attributes
  (e) wait for time
  (f) wait for event, signal event
  (g) allocate and free memory.

(ii) File management: These are used to create and delete files and open, close files etc, (read, write).
  (a) Create file, delete file
  (b) open, close
  (c) read, write, reposition
  (d) get file attributes, set file attributes.

(iii) Device management: A process may need several resources to execute i.e main memory, disk drivers, access to files and so and control over the resources needed:
  (a) request device, release device
  (b) read, write, reposition
  (c) get device attributes, set device attributes
  (d) logically attach or detach devices

(iv) Information maintenance: many system calls simply for the purpose of transferring information between the user program and the operating system. Such system calls are:

(a) get time or date, set time or date.
(b) get system data, set system data.
(c) get process, file or device attributes.
(d) set process, file or device attributes.

(v) Communications: There are two models of interprocess communication: the message passing model
the shared memory model.

In message passing model before communication a connection must be opened.

(a) create, delete communication connection
(b) send, receive messages
(c) transfer status information
(d) attach or detach remote devices.

→ Operating Systems Generation:

- Operating systems are designed to run on any of a class of machines at a different sites and at a different of peripheral configurations.
- The system must be configured or generated for each specific computer site, that process known as system generation (SYSGEN).
- The SYSGEN process runs as a series of jobs under the control of the operating system. There are various kinds of information must be determined while configured:

(a) The CPU used, options in that CPU like extended instruction sets, floating point arithmetic and so on.
(b) The memory available
(c) The devices available i.e device number, the device interrupt number, the device's type and model and any special device
(d) The operating system parameters & values.

Once the above information is determined, it can be used in several ways.

(a) At one extreme, a system administrator can use it to modify a copy of the source code of the operating system. The operating system then is a compiled one. Data declarations, initializations and constants, along with conditional computation produce an output object version of the OS that is tailored on a system described.

(b) At the another extreme, it is possible to construct a system that is completely table driven. All the code is available as part of the system, and selection occurs at execution time. System generation involves simply creating the tables to describe the system.

→ Generation of operating systems:

operating system like hardware have undergone a series of revolutionary changes called generations. In computer hardware the generations have been marked advances in components from vacuum tubes, to transistors, to integrated circuits, to large scale and very large scale integrated circuits.

① Serial processing / Zeroth generation:

Early computer systems from the late 1940s to mid 1950s the programmer directly interacted with the computer hardware, there was no OS. Programs in machine code were loaded via a input device. If an error occured the error condition was indicated by the lights. If the program executed normally, the output appeared on the printer.

Disadv: Scheduling, a user may sign up and finish in 45 mins.
setup time, a single job could involve loading the compiled program and linking together the object program and common functions.

This mode of operation is known as serial processing.

② Simple Batch Systems / the first generation:

The operating systems of the 1950's were designed to smooth transition between jobs. The wasted time due to scheduling and setup time was not say acceptable in this. To improve processor utilization the batch operating system was developed. The first batch operating systems was developed in the mid 1950's in which jobs were gathered in groups or batches. The basic idea in batch processing is the use of a software known as monitor. The monitor performs a scheduling function. A batch of jobs is queued up and jobs are executed with no interleaving idle time. There is job Control language (JCL) which is used to provide instructio to the monitor.

Eg: one such language is FORTRAN. The overall format of the job is as:

```
$JOB
$FTN
  :  } FORTRAN instructions
  :
$LOAD
$RUN
  :  } Data
  :
$END.
```

③ multiprogrammed Batch Systems / the second generation:

The second generation of operating systems was characterized by the development of shared systems with multiprogramming and beginnings of multiprocessing. In multiprogramming several user programs are in main memory at once and the processor is switched between the jobs. In multi processing systems, several processors are used in a single computer system to increase the processing ability. These systems are more expensive and large in size.