UNIT-I

INTRODUCTION

A computer system has many resources (hardware and software), which may be require to complete a task. The commonly required resources are input/output devices, memory, file storage space, CPU etc. The operating system acts as a manager of the above resources and allocates them to specific programs and users as necessary for their task. Therefore operating system is the resource manager i.e. it can manage the resource of a computer system internally. The resourcesare processor, memory, files, and I/O devices.

An Operating System (OS) is an interface between computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Examples of operating systems

There are many different operating systems. Each do the same thing: they control all input, processing and output.

Some popular Operating Systems include

1. **DOS -** Disk Operating System
   One of the first operating systems for the personal computer.
   DOS is an old (early 1980s) command-line operating system that requires the user to memorise and enter all commands from a command prompt.
   It was not "user friendly"
2. **Microsoft Windows**: A widely used operating system for desktop and laptop computers. Various versions, such as Windows 10 and Windows 11, have been popular among users. The Windows operating system, a product of Microsoft, is a GUI (graphical user interface) Operating system. This type of "user friendly" operating system is said to have WIMP features:

   Windows
   Icons
   Menus
   Pointing device (mouse)

3. macOS: Macintosh, a product of Apple, has its own operating system with a GUI and WIMP features.

4. Linux: An open-source operating system that comes in various distributions, such as Ubuntu, Fedora, Debian, and many others. Linux is popular among developers, server administrators

5. Android: An operating system developed by Google for mobile devices, such as smartphones and tablets. Android is the most widely used mobile OS globally.

6. iOS: Another operating system developed by Apple Inc., specifically for their mobile devices like iPhones and iPads.

7. Chrome OS: An operating system developed by Google, primarily designed for use with Chromebooks and other devices that rely heavily on cloud-based applications.

Definition:

An operating is system software which controls or manages the resources available at the computer system.

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Goals of OPERATING SYSTEMS

- **Convenience:** An OS makes a computer more convenient to use.
- **Efficiency:** An OS allows the computer system resources to be used in an efficient manner.

Barebones Computer System

A barebones computer system, also known as a barebones PC or barebones kit, is a partially assembled computer that includes the essential components required to build a functional system. However, it typically lacks certain key components like a central processing unit (CPU), memory (RAM), storage drives, and sometimes even an operating system. The purpose of a barebones system is to provide a basic foundation for users to customize and complete the computer according to their specific needs and preferences.

A typical barebones computer system usually includes the following core components:

1. Chassis/Casing: The enclosure that holds and protects the internal components of the computer.
2. Motherboard: The main circuit board that houses various components, provides connectivity, and determines the compatibility of other hardware.
3. Power Supply Unit (PSU): Supplies electrical power to all the components in the computer.

Optional components that might be included in some barebones kits:

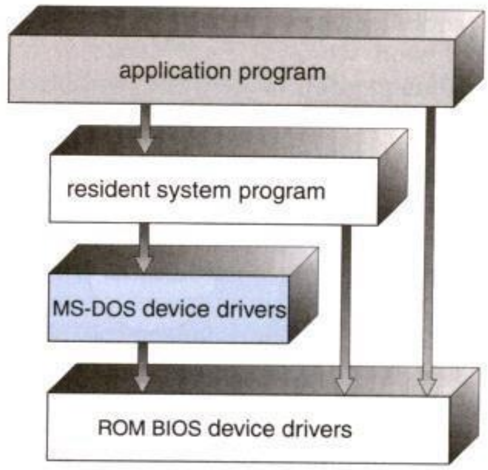
4. Central Processing Unit (CPU): The processor is responsible for executing instructions and performing calculations.
5. Memory (RAM): Provides temporary storage for data and programs that the CPU is actively using.
6. Storage Drive: Such as a hard disk drive (HDD) or solid-state drive (SSD) for permanent data storage.
7. Optical Drive: Used for reading or writing CDs, DVDs, or Blu-ray discs (becoming less common in modern systems).
8. Graphics Card: Responsible for rendering images and graphics on the display.
9. Cooling System: Ensures the components don't overheat during operation.

Users need to purchase the missing components separately and assemble them into the barebones system to create a fully functional computer. This allows individuals to have more control over the selection of components, ensuring that the computer meets their specific requirements in terms of performance, storage capacity, and budget.

Barebones systems are popular among computer enthusiasts, DIY builders, and small businesses because they offer a cost-effective and customizable approach to building a computer. However, they require some technical knowledge and experience in computer assembly to complete the setup successfully.

OPERATING SYSTEM STRUCTURE

SIMPLE STRUCTURE (MS-DOS)



- Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.

- There was no CPU Execution Mode (user and kernel), and so errors in applications couldcause the whole system to crash.

SIMPLE STRUCTURE (UNIX)

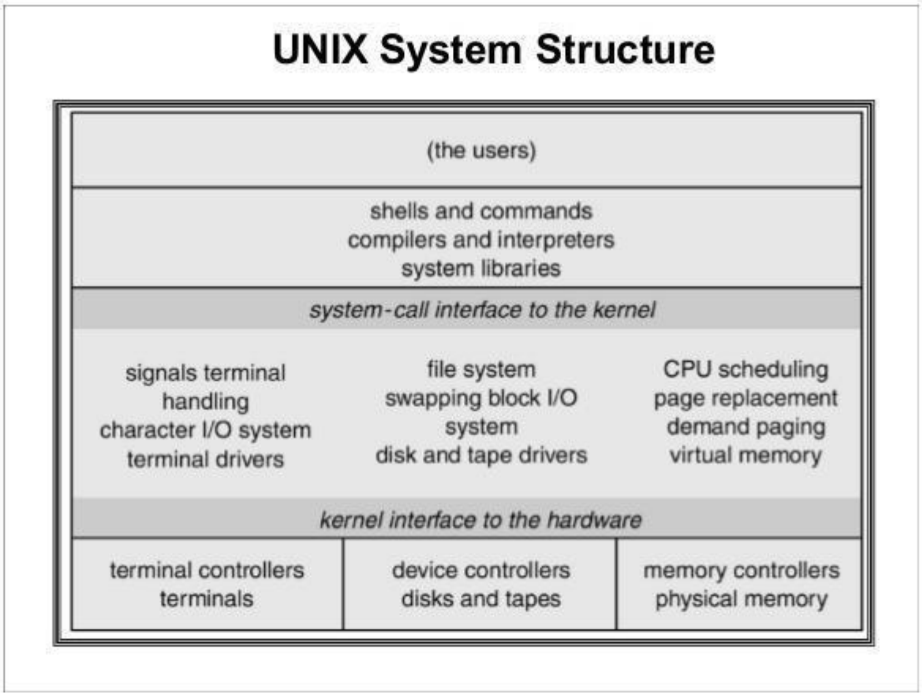
UNIX- Limited by hardware functionality, the original UNIX operating system had limited structuring. The UNIX OS consists of two separable parts:

System programs: system programs are provided to define the user interface

The kernel: consists of everything below the system-call interface and above the physicalhardware.

Separated into a series of interfaces and device drivers.

Different components (filesystem, CPU Scheduling, memory management and other OSfunctions)were layered in its design.

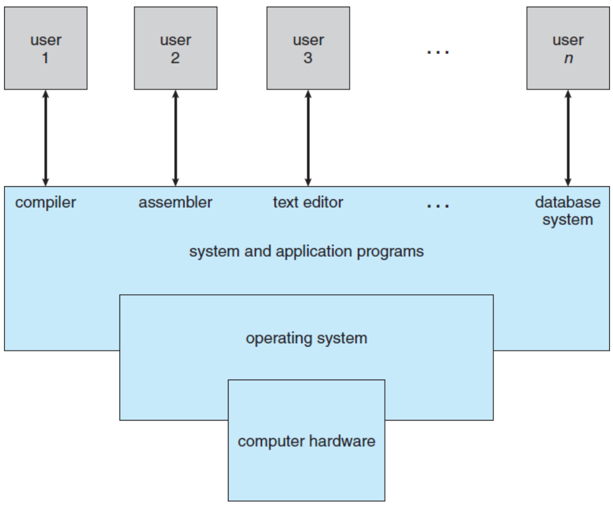


- Functionality of the OS is invoked with simple function calls within the kernel, which isone large program.
- Device drivers are loaded into the running kernel and become part of the kernel.

System components

A computer system can be divided roughly into four components: the **hardware,** the **operating system,** the **application programs,** and the **users.**

- **Hardware**: The hardware the central processing unit (CPU), the memory, and the input/output (I/O) devices—provide the basic computing resources for the system.
- **Application programs**: **The application programs**, such as word processors, spreadsheets, compilers, and Web browsers—define the ways in which these resources are used to solve users’ computing problems.
- **Operating system**: The operating system controls the hardware and coordinates its use among the various application programs for the various users.



We can also view a computer system as consisting of hardware, software, and data. The operating system provides the means for proper use of these resources in the operation of the computer system. An operating system is similar to a government. Like a government, it performs no useful function by itself. It simply provides an ***environment*** within which other programs can do useful work.

Operating System Definition

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

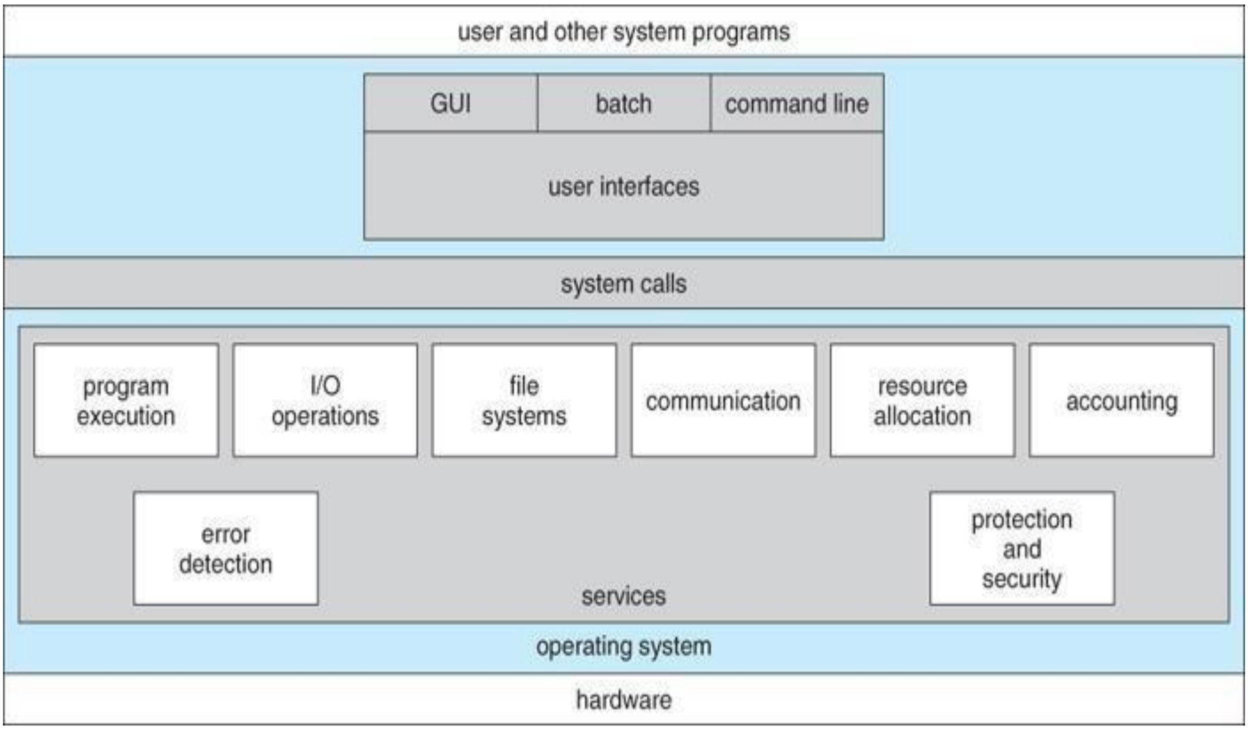
or

An Operating system is one of the programs running at all times on the computer, usually called kernel.

OPERATING SYSTEM SERVICES

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.



Following are a few common services provided by an operating system −

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection

Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated asa process.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management −

- Loads a program into memory.
- Executes the program.
- Handles program's execution.
- Provides a mechanism for process synchronization.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/OOperation

An I/O subsystem comprises of I/O devices and their corresponding driver software.Drivers hide the peculiarities of specific hardware devices from the users.

An Operating System manages the communication between user and device drivers.

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

Filesystemmanipulation

A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. Following are the major activities of an operating system with respect to file management −

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

Communication

In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes. Multiple processes communicate with one another throughcommunication lines in the network.

The OS handles routing and connection strategies, and the problems of contention and security. Following are the major activities of an operating system with respect to communication −

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or byMessage Passing.

Errorhandling

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling −

- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management −

- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.

Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection −

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

Facilites:
Operating system facilities refer to the various features, services, and functionalities provided by an operating system to manage and control computer hardware, software, and resources. These facilities enable users and applications to interact with the computer efficiently and securely. Here are some of the key operating system facilities:

1. Process Management: The OS manages processes, which are instances of running programs. It schedules processes, allocates CPU time, and provides mechanisms for inter-process communication and synchronization.

2. Memory Management: This facility manages the computer's memory by allocating and deallocating memory to processes, ensuring efficient memory utilization, and protecting processes from accessing each other's memory.

3. File System: The file system provides a structured way to store and organize data on storage devices, such as hard drives, solid-state drives, or network storage. It manages files, directories, and permissions.

4. Device Drivers: Operating systems use device drivers to communicate with hardware devices such as printers, scanners, graphics cards, and network cards. Device drivers allow the OS to abstract hardware specifics and provide a unified interface for applications.

5. Input/Output (I/O) Management: The OS handles input and output operations for devices like keyboards, mice, monitors, and disks, ensuring that data is transmitted correctly and efficiently.

6. Security and Protection: Operating systems enforce security measures to protect data and prevent unauthorized access. This includes user authentication, access control, and encryption.

7. Networking: The OS provides networking facilities to enable communication between computers, including protocols, socket management, and network stack implementations.

8. Virtualization: Some operating systems support virtualization, which allows multiple virtual machines to run on a single physical machine, enabling efficient resource utilization and isolation.

9. System Calls: System calls are interfaces provided by the OS for applications to request services from the kernel, such as creating processes, reading files, and managing memory.

10. Inter-Process Communication (IPC): The OS facilitates communication between processes through mechanisms like pipes, sockets, shared memory, and message queues.

11. Error Handling: The OS handles various types of errors and exceptions that may occur during program execution to prevent system crashes and data corruption.

12. Scheduling: The OS employs scheduling algorithms to manage the allocation of CPU time to different processes, optimizing performance and responsiveness.

These are just some of the essential facilities provided by modern operating systems. The exact set of features may vary depending on the OS type (e.g., Windows, macOS, Linux) and its version. Operating system facilities play a crucial role in maintaining stability, security, and usability of the computer system.

OS Organization:

Operating system organization refers to the internal structure and architecture of an operating system. It involves how different components and modules of the operating system are designed, implemented, and interact with each other to provide the various facilities and services that the OS offers. The organization of an operating system can vary depending on its type (e.g., Windows, macOS, Linux) and its intended use (e.g., general-purpose, real-time, embedded).

The main components of an operating system and their organization typically include:

1. Kernel: The kernel is the core component of the operating system that directly interacts with the hardware and provides essential services, such as process scheduling, memory management, and device drivers. It runs in a privileged mode to access hardware resources directly and ensure the protection and isolation of user processes.

2. System Calls: System calls are interfaces provided by the kernel that allow applications to request services from the OS. When an application needs to perform privileged operations like file I/O or process creation, it makes a system call, and the kernel handles the request on behalf of the application.

3. Device Drivers: Device drivers are modules that allow the kernel to communicate with hardware devices. Each hardware device typically requires its specific driver, which acts as an intermediary between the device and the operating system.

4. Process Management: This module is responsible for creating, scheduling, and managing processes. It includes components like the process scheduler, which determines which process gets CPU time, and the process control block (PCB), which stores process-specific information.

5. Memory Management: The memory management module is responsible for allocating and deallocating memory to processes, ensuring memory protection and managing virtual memory if supported.

6. File System: The file system component manages the organization, storage, and retrieval of files on secondary storage devices. It provides an interface for applications to read, write, and manipulate files and directories.

7. Input/Output Management: This module handles the interaction between the operating system and input/output devices. It manages buffering, caching, and data transfer between memory and peripherals.

8. Security and Protection: The security module ensures the enforcement of access control policies, user authentication, and data protection to prevent unauthorized access and maintain system integrity.

9. User Interface: The user interface may consist of a command-line interface (CLI) or a graphical user interface (GUI) that allows users to interact with the operating system and run applications.

10. Networking: If the operating system supports networking, it includes components for managing network connections, protocols, and data transmission.

The exact organization and design of these components can vary significantly between different operating systems, as each system may prioritize different features or emphasize specific use cases. Moreover, modern operating systems often modularize their code to improve maintainability, scalability, and portability.

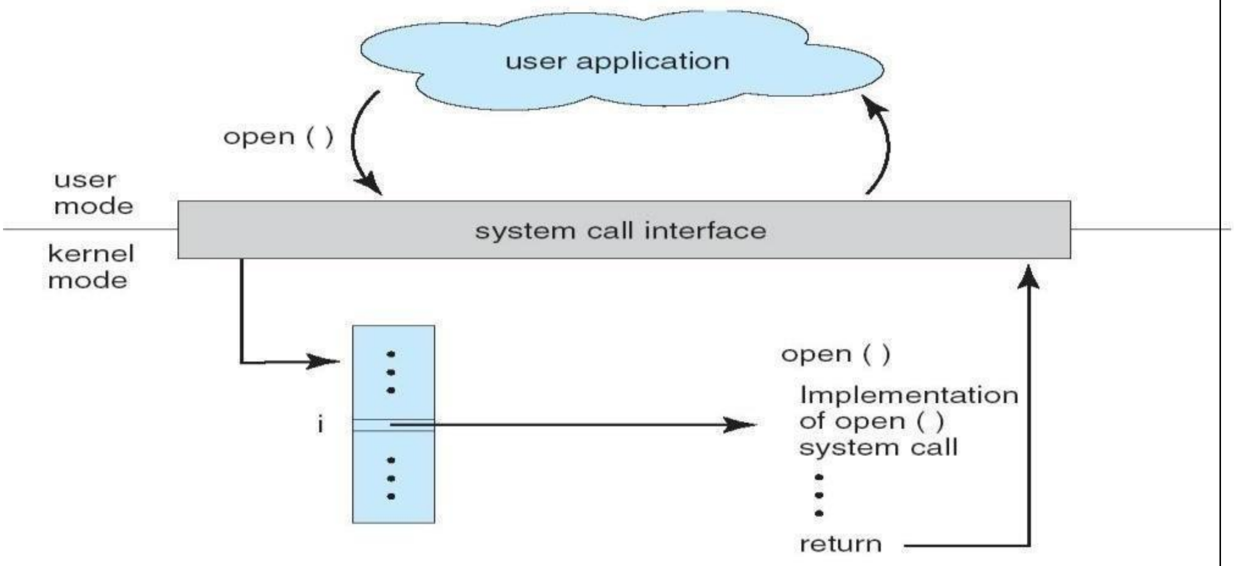
Overall, the organization of an operating system is a critical aspect of its functionality and performance, determining how efficiently it can manage resources and provide services to applications and users.

SYSTEM CALLS

Def:

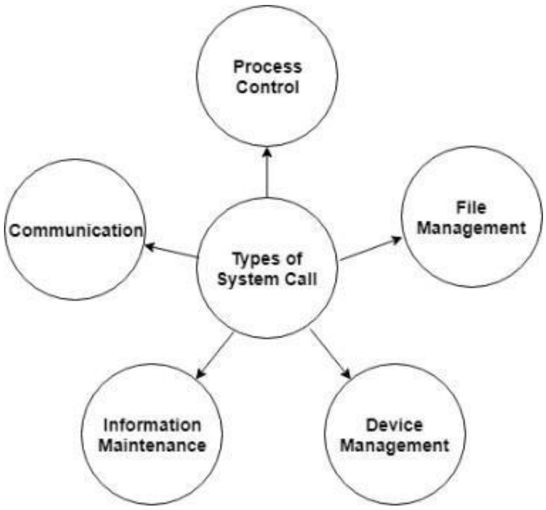
The interface between a process and an operating system is provided by system calls. In general, system calls are available as assembly language instructions.

They are also included in the manuals used by the assembly level programmers.
System calls are usually made when a process in user mode requires access to a resource.Then it requests the kernel to provide the resource via a system call.



Types of System Calls

There are mainly five types of system calls. These are explained in detail as follows:



Process Control

These system calls deal with processes such as process creation, process termination etc.

File Management

These system calls are responsible for file manipulation such as creating a file, reading afile, writing into a file etc.

Device Management

These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.

Information Maintenance

These system calls handle information and its transfer between the operating system and the user program.

Communication

These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

Some of the examples of all the above types of system calls in Windows and Unix are given as follows:

Types of System Calls	**Windows**	**Linux**
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	Create File() ReadFi le() WriteFile() CloseHandle()	open() read() write() close()
Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()

TYPES OF COMPUTER SYSTEMS

SERIAL PROCESSING

Users access the computer in series. From the late 1940's to mid-1950's, the programmer interacted directly with computer hardware i.e., no operating system.
These machines were run with a console consisting of display lights, toggle switches, some form of input device and a printer.
Programs in machine code are loaded with the input device like card reader.
If an error occur the program was halted and the error condition was indicated by lights. Programmers examine the registers and main memory to determine error.
if the program is success, then output will appear on the printer.

Disadvantages:
Main problem here is the setup time. That is single program needs to load source program into memory, saving the compiled (object) program and then loading and linking together.

SIMPLE BATCH SYSTEMS

The first computers used batch operating systems, in which the computer ran batches of jobs without stop.
Programs were punched into cards that were usually copied to tape for processing. When the computer finished one job, it would immediately start the next one on the tape.

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards or tapes and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows −

Lack of interaction between the user and the job.

- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

MULTIPROGRAMMING

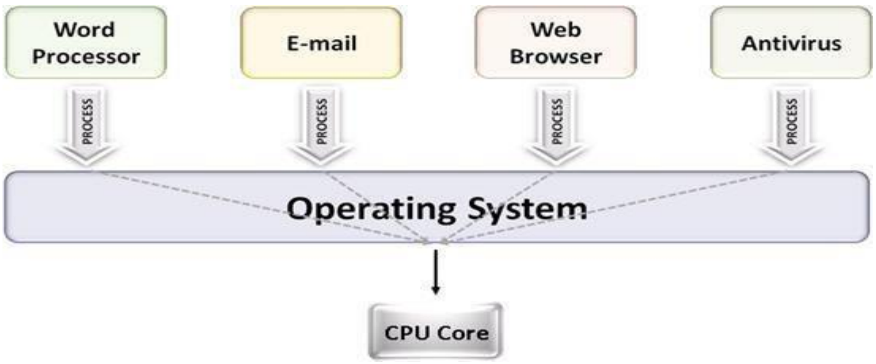
Multiprogramming is also the ability of an operating system to execute more than one program on a single processor machine. More than one task/program/job/process can resideinto the main memory at one point of time. A computer running excels and Firefox browser simultaneously is an example of multiprogramming.

Memory layout for Multiprogramming System

MULTI TASKING OR TIME-SHARING SYSTEMS

Time-sharing or multitasking is a logical extension of multiprogramming.
Multitasking is the ability of an operating system to execute more than one task simultaneously on a single processor machine. Though we say so but in reality no two tasks on a single processor machine can be executed at the same time.

Actually CPU switches from one task to the next task so quickly that appears as if all the tasks are executing at the same time. More than one task/program/job/process can reside into the same CPU at one point of time.



Advantages of Timesharing operating systems are as follows −

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows −

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

For example IBM's OS/360.

Time-sharing operating systems are even more complex than multiprogrammed operating systems. As in multiprogramming, several jobs must be kept simultaneously in memory.

PERSONAL COMPUTERS

A computers computer system is dedicated to a single user is called personal computer, appeared inthe 1970s. Micro are considerably smaller and less expensive than mainframecomputers.

The goals of the operating system have changed with time; instead of maximizing CPU and peripheral utilization, the systems developed for maximizing user convenience and responsiveness.
For eg., MS-DOS,MicrosoftWindows and Apple Macintosh.

DISTRIBUTEDSYSTEMS
Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.
The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as loosely coupled systems or distributed systems. Processors in a distributed system may vary in size and function. Theseprocessors are referred as sites, nodes, computers, and so on.

The advantages of distributed systems are as follows −

- With resource sharing facility, a user at one site may be able to use the resources availableat another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

REALTIMESYSTEMS

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the response time. So in this method, the response time is very less as compared to online processing.

Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Real time system is divided into two systems

- Hard Real Time Systems.
- Soft Real Time Systems.

Hard Real Time Systems:

Hard real time system is purely deterministic and time constraint system for example users expected the output for the given input in 10sec then system should process the input data and give the output exactly by 10th second. Here in the above example 10 sec. is the deadline to complete process for given data. Hard real systems should complete the process and give the output by 10th second. It should not give the output by 11th second or by 9th second, exactly by 10th second it should give the output. In the hard real time system meeting the deadline is very important if deadline is not met the system performance will fail. Another example is defense system if a country launched a missile to another country the missile system should reach the destiny at 4:00 to touch the ground what if missile is launched at correct time but it reached the destination ground by 4:05 because of performance of the system, with 5 minutes of difference destination is changed from one place to another place or even to another country. Here system should meet the deadline.

Soft Real Time System:

In soft real time system, the meeting of deadline is not compulsory for every time for everytask but process should get processed and give the result. Even the soft real time systems cannot miss the deadline for every task or process according to the priority it should meet the deadline or can miss the deadline. If system is missing the deadline for every time the performance of the system will be worse and cannot be used by the users. Best example for soft real time system is personal computer, audio and video systems, etc.

PARALLEL SYSTEMS

In general, Most systems are single-processor systems; that is they have only one main CPU. Multiprocessor systems have more than one processor.Multiprocessing operating system or the parallel system supports the use of more than one processor in close communication.

The advantages of the parallel (multiprocessing) system are:

1. Increased Throughput − By increasing the number of processors, more work can be completed in a unit time.

2. Cost Saving − Parallel system shares the memory, buses, peripherals etc. Multiprocessor system thus saves money as compared to multiple single systems. Also, if a number of programs are to operate on the same data, it is cheaper to store that data on one single disk and shared by all processors instead of using many copies of the same data.

3. Increased Reliability − In this system, as the workload is distributed among severalprocessors which results in increased reliability. If one processor fails then its failure may slightly slow down the speed of the system but system will work smoothly.

Purpose of User Interface

The purpose of a user interface (UI) is to act as an intermediary between users and a computer system or software application. It enables users to interact with the system, issue commands, and receive feedback in a way that is intuitive, efficient, and user-friendly. A well-designed user interface enhances user experience, making it easier for individuals to perform tasks, access information, and achieve their goals while using the software or system.

Types of User Interfaces:

1. Graphical User Interface (GUI): GUI is the most common type of user interface in modern computing. It utilizes visual elements such as icons, windows, menus, buttons, and pointers (e.g., mouse cursor) to enable users to interact with the system. GUIs are generally more intuitive and user-friendly, allowing users to perform tasks with the help of graphical representations.

2. Command-Line Interface (CLI): CLI is a text-based user interface where users interact with the system by typing commands. Users need to be familiar with specific commands and their syntax to perform tasks. While CLI interfaces may appear less user-friendly to some, they are often favored by experienced users and administrators for their efficiency and direct control over the system.

3. Voice User Interface (VUI): VUI allows users to interact with the system using spoken commands and responses. This type of interface is becoming increasingly popular with the rise of voice-activated virtual assistants like Amazon's Alexa, Apple's Siri, or Google Assistant.

4. Touchscreen Interface: Touchscreen interfaces are prevalent in smartphones, tablets, and some modern laptops. They allow users to interact with the system by directly touching and manipulating elements on the screen.

5. Menu-Driven Interface: A menu-driven interface presents users with a set of options organized in menus. Users navigate through these menus to perform specific tasks or access particular features.

6. Natural Language Interface: This type of interface allows users to interact with the system using natural language, as if they were communicating with another person. Natural language processing technology enables the system to understand and respond appropriately to user input.

7. Gesture-Based Interface: Gesture-based interfaces utilize physical gestures, such as swiping, pinching, or tapping, to control the system or perform specific actions. These interfaces are common in touch-based devices and virtual reality systems.

8. Web User Interface (Web UI): Web UI is specifically designed for web applications and websites. Users interact with the system through web browsers using clickable links, buttons, forms, and other web elements.

The choice of user interface type depends on the context, target audience, and the nature of the application or system being used. A good user interface should be well-designed, responsive, accessible, and aligned with users' needs and preferences.