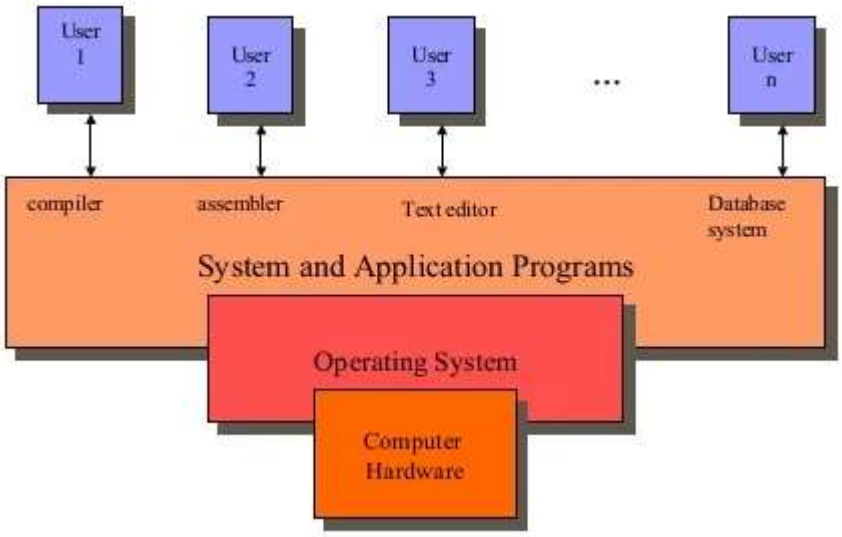
www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

UNIT–1 INTRODUCTION TO OPERATING SYSTEM

Introduction of computer system

- Computer consists of the hardware, Operating System, system programs, application programs.
- The hardware consists of memory, CPU, ALU, I/O device, storage device and peripheral device.
- System program consists of compilers, loaders, editors, OS etc.
- Application program consists of database programs, business programs.
- Every computer must have an OS to run other programs.
- The OS controls & coordinates the use of the hardware among the various system programs and application programs for various tasks.
- It simply provides an environment within which other programs can do useful work.



OPERATING SYSTEM

Definition

- In the 1960's one might have defined OS as **"The software that controls the hardware"**.
- Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. **resource manager**.
- Operating system becomes an interface between the user and the machine. It is one of the most required software that is present in the device.
- Operating System is a type of software that works as an interface between the system program and the hardware.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Concept of OS

- The OS is a set of special programs that run on a computer system that allow it to work properly.
- It performs basic task as recognizing input from the keyboard, keeping track of files and directories on the disk, sending output to the display screen and controlling a peripheral device.
- The OS must support the following tasks. They are,
  - Provides the facilities to create, modification of program and data file using an editor.
  - Access to the compiler for translating the user program from high level language to machine language.
  - Provide a loader program to move the compiled program code to the computer memory for execution.

OS as a resource allocator

- OS keeps track of the status of each resource and decides who gets a resource, for how long and when.
- OS makes sure that different programs and users running at the same time do not interfere with each other.
- It is also responsible for security, ensuring that unauthorized users do not access the system.
- The primary objective of OS is to increase productivity of a processing resource such as computer hardware or user.
- The OS is the first program run on a computer when the computer boots up.

The OS acts as a manager of these resources and allocates them to specific programs and user as necessary for tasks.

OS can be explored from two view points:

1. **The user view**

The user view depends on the system interface that is used by the users. The different types of user view experiences can be explained as follows −

- If the user is using a personal computer, the operating system is largely designed to make the interaction easy. Some attention is also paid to the performance of the system, but there is no need for the operating system to worry about resource utilization. This is because the personal computer uses all the resources available and there is no sharing.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- If the user is using a system connected to a mainframe or a minicomputer, the operating system is largely concerned with resource utilization. This is because there may be multiple terminals connected to the mainframe and the operating system makes sure that all the resources such as CPU, memory, I/O devices etc. are divided uniformly between them.
- If the user is sitting on a workstation connected to other workstations through networks, then the operating system needs to focus on both individual usage of resources and sharing though the network. This happens because the workstation exclusively uses its own resources but it also needs to share files etc. with other workstations across the network.
- If the user is using a handheld computer such as a mobile, then the operating system handles the usability of the device including a few remote operations. The battery level of the device is also taken into account.

There are some devices that contain very less or no user view because there is no interaction with the users. Examples are embedded computers in home devices, automobiles etc.

2. The system view

According to the computer system, the operating system is the bridge between applications and hardware. It is most intimate with the hardware and is used to control it as required.

The different types of system view for operating system can be explained as follows:

- The system views the operating system as a resource allocator. There are many resources such as CPU time, memory space, file storage space, I/O devices etc. that are required by processes for execution. It is the duty of the operating system to allocate these resources judiciously to the processes so that the computer system can run as smoothly as possible.
- The operating system can also work as a control program. It manages all the processes and I/O devices so that the computer system works smoothly and there are no errors. It makes sure that the I/O devices work in a proper manner without creating problems.
- Operating systems can also be viewed as a way to make using hardware easier.
- Computers were required to easily solve user problems. However it is not easy to work directly with the computer hardware. So, operating systems were developed to easily communicate with the hardware.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- An operating system can also be considered as a program running at all times in the background of a computer system (known as the kernel) and handling all the application programs. This is the definition of the operating system that is generally followed.

Names of OS

DOS, windows 3,windows 95/98, windows NT/2000, Unix, Linux etc.

Classification of OS

OS classified into two types:

1. **Character user interface (CUI) / Single user OS**
   - The user can interact with computer through commands.
   - We cannot use mouse here,
   - It is not user friendly,
   - If user knows the command only the user can interact with computer.
   - For example: DOS, Unix, Linux.

2. **Graphical user interface (GUI) / Multi user OS**
   - It is user friendly.
   - For example: Windows.

GOALS OF OS

- Simplify the execution of user programs and make solving user problems easier.
- Use computer hardware efficiently.
- Make application software portable and versatile.
- Provide isolation, security and protection among user programs.
- Improve overall system reliability.

WHY SHOULD WE STUDY OPERATING SYSTEMS

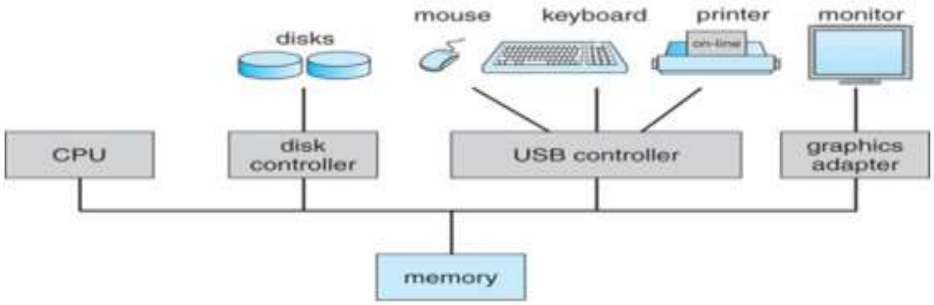
- Need to understand interaction between the hardware and application.
- Need to understand basic principles in the design of computer systems.
- Increasing need for specialized operating systems. For example:
  - Real-time operating systems – aircraft control, multimedia services.
  - Embedded operating systems for devices – cell phones, sensors and controllers.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Computer System Organisation

The computer system is a combination of many parts such as peripheral devices, secondary memory, CPU etc. This can be explained more clearly using a diagram.



The salient points about the above figure displaying Computer System Organisation is −

- Computer consists of processor, memory, and I/O components, with one or more modules of each type. These modules are connected through interconnection network.
- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers.
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an interrupt.

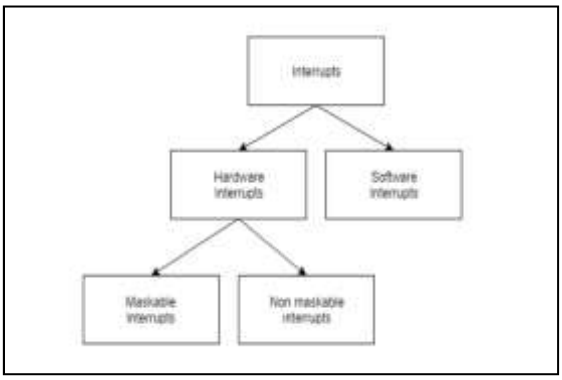
Interrupt Handling

An interrupt is a necessary part of Computer System Organisation as it is triggered by hardware and software parts when they need immediate attention.

An interrupt can be generated by a device or a program to inform the operating system to halt its current activities and focus on something else.

Types of interrupts

Hardware and software interrupts are two types of interrupts. Hardware interrupts are triggered by hardware peripherals while software interrupts are triggered by software function calls.



Hardware interrupts are of further two types. Maskable interrupts can be ignored or disabled by the CPU while this is not possible for non-maskable interrupts.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

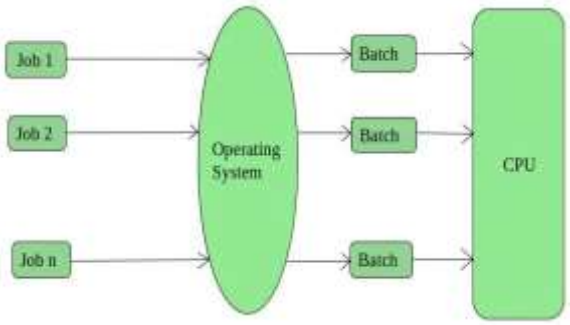
Types of Operating Systems

There are several types of Operating Systems which are mentioned below.

- Batch Operating System
- Multi-Programming System
- Multi-Processing System
- Multi-Tasking Operating System
- Time-Sharing Operating System
- Personal Computers
- Parallel Operating System
- Distributed Operating System
- Network Operating System
- Real-Time Operating System

1. Batch Operating System

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and groups them into batches. It is the responsibility of the operator to sort jobs with similar needs.



Advantages

- Processors of the batch systems know how long the job would be when it is in the queue.
- Multiple users can share the batch systems.
- The idle time for the batch system is very less.
- It is easy to manage large work repeatedly in batch systems.

Disadvantages

- The computer operators should be well known with batch systems.
- Batch systems are hard to debug.
- It is sometimes costly.
- The other jobs will have to wait for an unknown time if any job fails.
- It is very difficult to guess or know the time required for any job to complete.

Examples

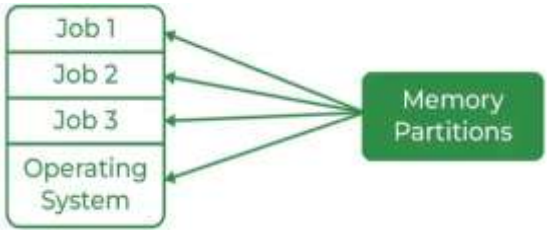
Payroll Systems, Bank Statements, etc.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

2. Multi-Programming Operating System

Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution. This is basically used for better execution of resources.



Advantages of Multi-Programming Operating System

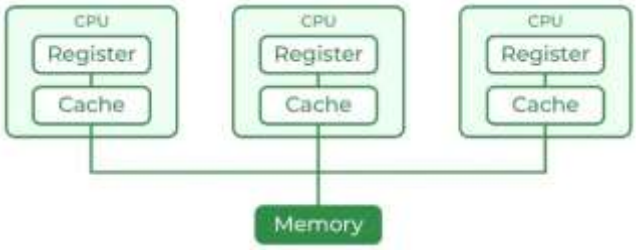
- Multi Programming increases the Throughput of the System.
- It helps in reducing the response time.

Disadvantages of Multi-Programming Operating System

- There is not any facility for user interaction of system resources with the system.

3. Multi-Processing Operating System

It is a type of Operating System in which more than one CPU is used for the execution of resources. It betters the throughput of the System.



Advantages

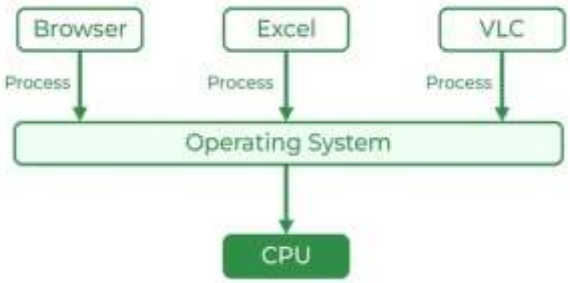
- It increases the throughput of the system.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

Disadvantages of Multi-Processing Operating System

- Due to the multiple CPU, it can be more complex and somehow difficult to understand.

4. Multi-Tasking Operating System

Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling Algorithm. It can run multiple programs simultaneously. There are two types of Multi-Tasking Systems which are listed below.



www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Preemptive Multi-Tasking

The operating system can initiate a context switching from the running process to another process. In other words, the operating system allows stopping the execution of the currently running process and allocating the CPU to some other process.

Cooperative Multi-Tasking

The operating system never initiates context switching from the running process to another process. A context switch occurs only when the processes voluntarily yield control periodically or when idle or logically blocked to allow multiple applications to execute simultaneously. Also, in this multitasking, all the processes cooperate for the scheduling scheme to work.

Advantages

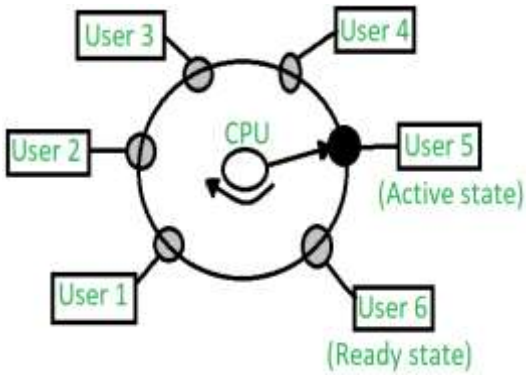
- Multiple Programs can be executed simultaneously in Multi-Tasking Operating System.
- It comes with proper memory management.

Disadvantages of Multi-Tasking Operating System

- The system gets heated in case of heavy programs multiple times.

5. Time-Sharing Operating Systems

Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of the CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



Advantages

- Each task gets an equal opportunity.
- Fewer chances of duplication of software.
- CPU idle time can be reduced.
- **Resource Sharing:** Time-sharing systems allow multiple users to share hardware resources such as the CPU, memory, and peripherals, reducing the cost of hardware and increasing efficiency.
- **Improved Productivity:** Time-sharing allows users to work concurrently, thereby reducing the waiting time for their turn to use the computer. This increased productivity translates to more work getting done in less time.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- **Improved User Experience:** Time-sharing provides an interactive environment that allows users to communicate with the computer in real time, providing a better user experience than batch processing.

Disadvantages

- Reliability problem.
- One must have to take care of the security and integrity of user programs and data.
- Data communication problem.
- **High Overhead:** Time-sharing systems have a higher overhead than other operating systems due to the need for scheduling, context switching, and other overheads that come with supporting multiple users.
- **Complexity:** Time-sharing systems are complex and require advanced software to manage multiple users simultaneously. This complexity increases the chance of bugs and errors.
- **Security Risks:** With multiple users sharing resources, the risk of security breaches increases. Time-sharing systems require careful management of user access, authentication, and authorization to ensure the security of data and software.

Examples

- **IBM VM/CMS**: IBM VM/CMS is a time-sharing operating system that was first introduced in 1972. It is still in use today, providing a virtual machine environment that allows multiple users to run their own instances of operating systems and applications.
- **TSO (Time Sharing Option)**: TSO is a time-sharing operating system that was first introduced in the 1960s by IBM for the IBM System/360 mainframe computer. It allowed multiple users to access the same computer simultaneously, running their own applications.
- **Windows Terminal Services**: Windows Terminal Services is a time-sharing operating system that allows multiple users to access a Windows server remotely. Users can run their own applications and access shared resources, such as printers and network storage, in real-time.

6. Personal Computer

A personal computer (PC) is a microcomputer designed for use by one person at a time.

Prior to the PC, computers were designed for -- and only affordable for -- companies that attached terminals for multiple users to a single large mainframe computer whose resources were shared among all users. By the

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

1980s, technological advances made it feasible to build a small computer that an individual could own and use as a word processor and for other computing functions.

Whether they are home computers or business ones, PCs can be used to store, retrieve and process data of all kinds. A PC runs firmware that supports an operating system (OS), which supports a spectrum of other software. This software lets consumers and business users perform a range of general-purpose tasks, such as the following:

- word processing
- spreadsheets
- email
- instant messaging
- accounting
- database management
- internet access
- listening to music
- network-attached storage
- graphic design
- music composition
- video gaming
- software development
- network reconnaissance
- multimedia servers
- wireless network access hotspots
- video conferencing

Types

Personal computers fall into various categories, such as the following:

- **Desktop computers** usually have a tower, monitor, keyboard and mouse.
- **Tablets** are mobile devices with a touchscreen display.
- **Smartphones** are phones with computing capabilities.
- **Wearables** are devices users wear, such as smartwatches and various types of smart clothing.
- **Laptop computers** are portable personal computers that usually come with an attached keyboard and trackpad.
- **Notebook computers** are lightweight laptops.
- **Handheld computers** include advanced calculators and various gaming devices.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

7. Parallel Operating System

Parallel Systems are designed to speed up the execution of programs by dividing the programs into multiple fragments and processing these fragments at the same time.

Advantages

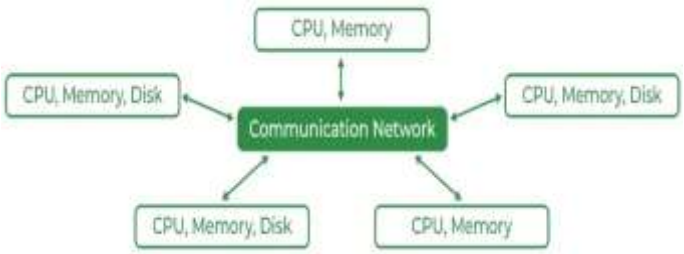
- **High Performance:** Parallel systems can execute computationally intensive tasks more quickly compared to single processor systems.
- **Cost Effective:** Parallel systems can be more cost-effective compared to distributed systems, as they do not require additional hardware for communication.

Disadvantages

- **Limited Scalability:** Parallel systems have limited scalability as the number of processors or cores in a single computer is finite.
- **Complexity:** Parallel systems are more complex to program and debug compared to single processor systems.
- **Synchronization Overhead:** Synchronization between processors in a parallel system can add overhead and impact performance.

8. Distributed Operating System

These types of operating systems are a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, at a great pace.



Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as loosely coupled systems or distributed systems. These systems' processors differ in size and function.

The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.

Types of Distributed Systems

The nodes in the distributed systems can be arranged in the form of client/server systems or peer to peer systems. Details about these are as follows −

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Client/Server Systems

In client server systems, the client requests a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server. Both the client and server usually communicate via a computer network and so they are a part of distributed systems.

Peer to Peer Systems

The peer to peer systems contains nodes that are equal participants in data sharing. All the tasks are equally divided between all the nodes. The nodes interact with each other as required as share resources. This is done with the help of a network.

Advantages

- Failure of one will not affect the other network communication, as all systems are independent of each other.
- Electronic mail increases the data exchange speed.
- Since resources are being shared, computation is highly fast and durable.
- Load on host computer reduces.
- These systems are easily scalable as many systems can be easily added to the network.
- Delay in data processing reduces.

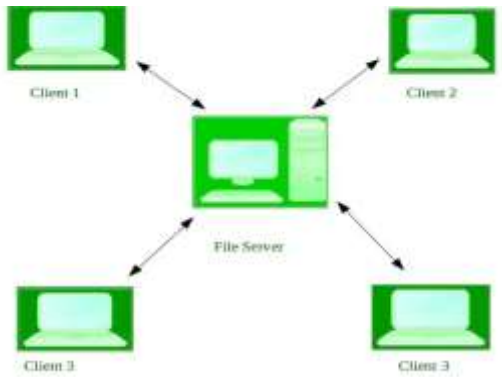
Disadvantages

- Failure of the main network will stop the entire communication.
- To establish distributed systems the language is used not well-defined yet.
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet.

Example: LOCUS

9. Network Operating System

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access to files, printers, security, applications, and other networking functions over a small private network.



www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that's why these computers are popularly known as tightly coupled systems.

Advantages

- Highly stable centralized servers.
- Security concerns are handled through servers.
- New technologies and hardware up-gradation are easily integrated into the system.
- Server access is possible remotely from different locations and types of systems.

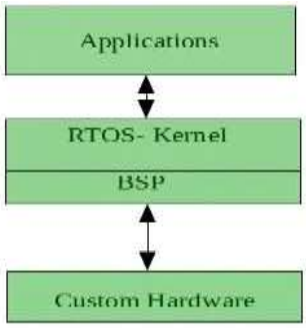
Disadvantages

- Servers are costly.
- User has to depend on a central location for most operations.
- Maintenance and updates are required regularly.

Examples

Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, BSD, etc.

10. Real-Time Operating System



These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.

Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Types:

1. **Hard Real-Time Systems**

   Hard Real-Time OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of an accident. Virtual memory is rarely found in these systems.

2. **Soft Real-Time Systems**

   These OSs are for applications where time-constraint is less strict.

Advantages

- **Maximum Consumption:** Maximum utilization of devices and systems, thus more output from all the resources.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- **Task Shifting:** The time assigned for shifting tasks in these systems is very less. For example, in older systems, it takes about 10 microseconds in shifting from one task to another, and in the latest systems, it takes 3 microseconds.
- **Focus on Application:** Focus on running applications and less importance on applications that are in the queue.
- Real-time **operating system in** the **embedded system:** Since the size of programs is small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

Disadvantages

- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on a few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupts signal to respond earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples

Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Mainframe Systems

- First commercial systems: Enormous, expensive and slow.
- I/O: Punch cards and line printers.
- Single operator/programmer/user runs and debugs interactively:
- Standard library with no resource coordination
- Monitor that is always resident
- Inefficient use of hardware: poor *throughput* and poor *utilization*
- They initially executed one program at a time and were known as batch systems.

Throughput: Amount of useful work done per hour
Utilization: keeping all devices busy

Operating System Services

- **User Interface -** User interface is essential and all operating systems provide it. Users either interface with the operating system through command-line interface (CUI) or graphical user interface (GUI). Command interpreter executes next user-specified command. A GUI offers the user a mouse-based window and menu system as an interface.
- **Program execution** - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
- **I/O operations** - A running program may require I/O, which may involve a file or an I/O device.
- **File-system manipulation** - The file system is of particular interest. Obviously, programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.
- **Communications** – Processes may exchange information, on the same computer or between computers over a network. Communications may be via shared memory or through message passing (packets moved by the OS)
- **Error detection** – OS needs to be constantly aware of possible errors may occur in the CPU and memory hardware, in I/O devices, in user program. For each type of error, OS should take the appropriate action to ensure correct and consistent computing. Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system.

Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them. Many types of resources such as CPU cycles, main memory, and file storage may have special allocation code, others such as I/O devices may have general request and release code.
- **Accounting -** To keep track of which users use how much and what kinds of computer resources
- **Protection and security -** The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other. **Protection** involves ensuring that all access to system resources is controlled. **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts. If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

System Calls

- A system call is a way for a user program to interface with the operating system. The program requests several services, and the OS responds by invoking a series of system calls to satisfy the request.
- A system call can be written in assembly language or a high-level language like **C, C++** or **Pascal**.
- System calls are predefined functions that the operating system may directly invoke if a high-level language is used.
- A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running.
- A system call is a method of interacting with the operating system via programs.
- A system call is a request from computer software to an operating system's kernel.
- A simple system call may take few nanoseconds to provide the result, like retrieving the system date and time. A more complicated system call, such as connecting to a network device, may take a few seconds. Most operating systems launch a distinct kernel thread for each system call to avoid bottlenecks. Modern operating systems are multi-threaded, which means they can handle various system calls at the same time.
- The **Application Program Interface (API)** connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services. The kernel system can only be accessed using system calls. System calls are required for any programs that use resources.

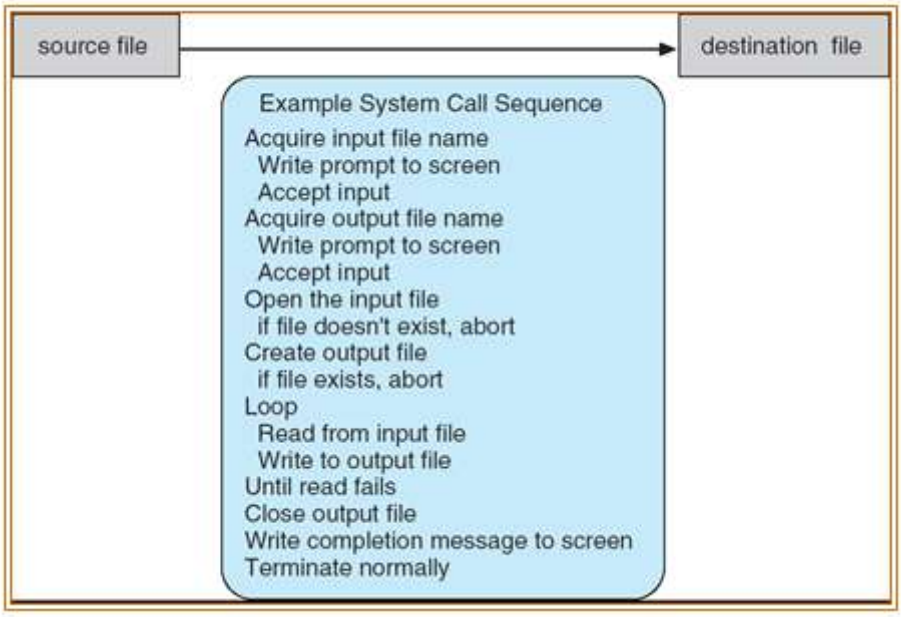
www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

- When computer software needs to access the operating system's kernel, it makes a system call. The system call uses an API to expose the operating system's services to user programs. It is the only method to access the kernel system. All programs or processes that require resources for execution must use system calls, as they serve as an interface between the operating system and user programs.

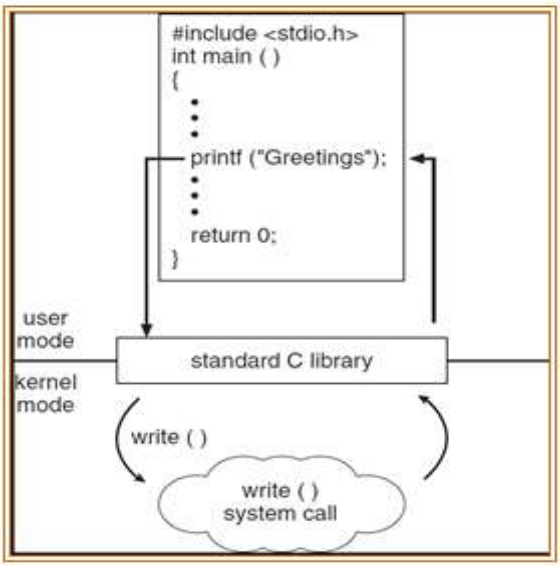
Example of System Calls

System call sequence to copy the contents of one file to another file



Standard C Library Example

C program invoking printf() library call, which call write() system call



www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

There are various situations where we must require system calls in the operating system. Following of the situations are as follows:

1. It is must require when a file system wants to create or delete a file.
2. Network connections require the system calls to sending and receiving data packets.
3. If you want to read or write a file, you need to system calls.
4. If you want to access hardware devices, including a printer, scanner, you need a system call.
5. System calls are used to create and manage new processes.

Types of System Calls

There are commonly five types of system calls. These are as follows:

1. **Process Control**
2. **File Management**
3. **Device Management**
4. **Information Maintenance**
5. **Communication**

Process Control

Process control is the system call that is used to direct the processes. Some process control examples include creating, load, abort, end, execute, process, terminate the process, etc.

File Management

File management is a system call that is used to handle the files. Some file management examples include creating files, delete files, open, close, read, write, etc.

Device Management

Device management is a system call that is used to deal with devices. Some examples of device management include read, device, write, get device attributes, release device, etc.

Information Maintenance

Information maintenance is a system call that is used to maintain information. There are some examples of information maintenance, including getting system data, set time or date, get time or date, set system data, etc.

Communication

Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

Examples of Windows and Unix system calls

Process	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	Fork() Exit() Wait()
File Manipulation	CreateFile() ReadFile() 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()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorgroup()	Chmod() Umask() Chown()

open()

The **open()** system call allows you to access a file on a file system. It allocates resources to the file and provides a handle that the process may refer to. Many processes can open a file at once or by a single process only. It's all based on the file system and structure.

read()

It is used to obtain data from a file on the file system. It accepts three arguments in general:

- A file descriptor.
- A buffer to store read data.
- The number of bytes to read from the file.

The file descriptor of the file to be read could be used to identify it and open it using **open()** before reading.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh

www.android.universityupdates.in | www.universityupdates.in | https://telegram.me/jntuh

wait()

In some systems, a process may have to wait for another process to complete its execution before proceeding. When a parent process makes a child process, the parent process execution is suspended until the child process is finished. The **wait()** system call is used to suspend the parent process. Once the child process has completed its execution, control is returned to the parent process.

write()

It is used to write data from a user buffer to a device like a file. This system call is one way for a program to generate data. It takes three arguments in general:

- A file descriptor.
- A pointer to the buffer in which data is saved.
- The number of bytes to be written from the buffer.

fork()

Processes generate clones of themselves using the **fork()** system call. It is one of the most common ways to create processes in operating systems. When a parent process spawns a child process, execution of the parent process is interrupted until the child process completes. Once the child process has completed its execution, control is returned to the parent process.

close()

It is used to end file system access. When this system call is invoked, it signifies that the program no longer requires the file, and the buffers are flushed, the file information is altered, and the file resources are de-allocated as a result.

exec()

When an executable file replaces an earlier executable file in an already executing process, this system function is invoked. As a new process is not built, the old process identification stays, but the new process replaces data, stack, data, head, etc.

exit()

The **exit()** is a system call that is used to end program execution. This call indicates that the thread execution is complete, which is especially useful in multi-threaded environments. The operating system reclaims resources spent by the process following the use of the **exit()** system function.

www.android.previousquestionpapers.com | www.previousquestionpapers.com | https://telegram.me/jntuh