1. Define OS. Explain the functions of OS.

Answer

An Operating System (OS) is an interface between a 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. 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.

Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

Functions of OS are:

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address. Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. In multiprogramming, the OS decides which process will get memory when and how much. Allocates the memory when a process requests it to do so. De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called process scheduling. An Operating System does the following activities for processor management −

- Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File management

A file system is organized into directories for efficient or easy navigation and usage. These directories may contain other directories and other files. An Operating System carries out the following file management activities. It keeps track of where information is stored, user access settings and status of every file, and more. These facilities are collectively known as the file system.

Security

By means of password and similar other techniques, it prevents unauthorized access to programs and data. OS makes sure that only authorized users get access to the computer and its data and the users only do things they are authorized to do.

2. List the types of OS and explain all.

Answer

- Batch OS
- Distributed OS
- Multitasking OS
- Network OS
- Real-OS

Batch OS

Batch OS is the first operating system for second-generation computers. This OS does not directly interact with the computer. Instead, an operator takes up similar jobs and groups them together into a batch, and then these batches are executed one by one based on the first-come, first, serve principle. 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 and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group.

Distributed OS

A distributed operating system is a recent advancement in the field of computer technology and is utilized all over the world that too with great pace. In a distributed OS, various computers are connected through a single communication channel. These independent computers have their memory unit and CPU and are known as loosely coupled systems. The system processes can be of different sizes and can perform different functions. The major benefit of such a type of operating system is that a user can access files that are not present on his system but another connected system.

Multitasking OS

The multitasking OS is also known as the time-sharing operating system as each task is given some time so that all the tasks work efficiently. This system provides access to a large number of users, and each user gets the time of CPU as they get in a single system. The tasks performed are given by a single user or by different users. The time allotted to execute one task is called a quantum, and as soon as the time to execute one task is completed, the system switches over to another task.

Network OS

Network operating systems are the systems that run on a server and manage all the networking functions. They allow sharing of various files, applications, printers, security, and other networking functions over a small network of computers like LAN or any other private network. In the network OS, all the users are aware of the configurations of every other user within the network, which is why network operating systems are also known as tightly coupled systems.

Real OS

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. 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.

There are two types of real-time operating systems.

Hard real-time systems

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

Soft real-time systems

Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

3. What are the objectives of OS? Explain.

Answer

The main objectives of an operating system are as follows:

- Convenience**:** This appears to be the most obvious objective of an operating system. Without an operating system, users would have to deal with the hardware directly with not access to pre-configured utility packages that come with an operating system. This would make the use of a computer very inconveniencing. Operating systems allow users to get started with the tasks they want to accomplish in no time without having to deal with the stress of configuring the system first.

- Efficiency**:** The operating systems improves efficiency of output. This is because less time is spent in configuring the system. System activities like granting of resources to processes, as well as conflict resolution among different programs and users are all handled by the operating system by default. This saves the user time thereby making for a more efficient outcome.

- Abstraction of hardware details**:** The operating system does a good job in hiding the complex details about the computer system. The user is able to make full use of the computer hardware without having to deal with the complexities associated with it. The operating system takes care of the communication between the user programs and the computer hardware.

- Management of system resources**:** The operating system acts as a referee. It plays management roles in the computer system because it ensures fair sharing of resources among different processes and users.

- To make the computer system convenient to use in an efficient manner.
- To hide the details of the hardware resources from the users.
- To provide users a convenient interface to use the computer system.
- To act as an intermediary between the hardware and its users, making it easier for the users to access and use other resources.
- To manage the resources of a computer system.

4. OS acts as resource manager. Explain in detail.

Answer

An Operating System is a collection of programs and utilities. It acts as an interface between user and computer. It creates the user-friendly environment. Another main operating system function is resource management. The operating system collects all the resources in the network environment and allocates the resources to requesting process in an efficient manner. A computer has many resources (Hardware and Software), which may be required to complete a task. The commonly required resources are Input/Output devices, Memory file storage space, CPU (Central Processing Unit) time and so on. The operating system acts as the manager of these resources and allocates them to specific programs and users as necessary for their tasks. Therefore, we can say an operating system is a resource allocator. When a number of computers connected through a network more than one computer trying for a computer print or a common resource, then the operating system follows same order and manage the resources in an efficient manner.

Generally, resources sharing in two ways "in time" and "in space". When a resource is a time-sharing resource, first one of the tasks get the resource for some time, then another and so on.

5. What are the advantage and disadvantage of OS? Explain.

Answer

Advantages

Computing Source

An operating system acts as an interface between the user and the hardware. It allows users to input data, process it, and access the output. Besides, through the operating system, users can communicate with computers to perform various functions such as arithmetic calculations and other significant tasks.

User-Friendly Interface

Windows operating system, when it came into existence, also introduces Graphical User Interface (GUI), which made using computers much more natural than earlier Command Line Interface. Moreover, users can quickly understand, interacts, and communicate with computer machines.

Resource Sharing

Operating systems allow the sharing of data and useful information with other users via Printers, Modems, Players, and Fax Machines. Besides, a single user can share the same data with multiple users at the corresponding time via mails. Also, various apps, images, and media files can be transferred from PC to other devices with the help of an operating system.

Safeguard of Data

There’s a lot of user data stored on the computer, and that can only be accessed with the help of an OS. Besides, storing and accessing the data, another important task of an OS is to safely and securely manage the data. For example, Windows Defender in

Microsoft Windows detects malicious and harmful files and removes them. Also, it secures your data by storing them with a bit-to-bit encryption.

Software Update

An operating system is a software which needs to update regularly to control high fleeting features that are continually increasing. With other apps and software hitting updates to improve their functionality, OS must improve their benchmarks and handle all the working of a computer. An OS can easily be updated without any complexity.

Multitasking

An operating system can handle several tasks simultaneously. It allows users to carry out different tasks at the same point in time.

Disadvantages

Expensive

When compared to the open-source platforms like Linux, some operating systems are costly. While users can use a free OS but generally, they are a bit difficult to run than others. Moreover, operating systems such as Microsoft Windows with GUI functionality and other in-built features carry a costly price tag.

System Failure

If the central operating system fails, it will affect the whole system, and the computer will not work. Moreover, an OS is the heart of a computer system without which it cannot function. If the central system crashes, the whole communication will be halted, and there will be no further processing of data.

Highly Complex

Operating systems are highly complex, and the language which used to establish these OS are not clear and well defined. Besides, if there’s an issue with OS users cannot directly understand, and it cannot be resolved quickly.

Virus Threats

Threats to the operating systems are higher as they are open to such virus attacks. Many users download malicious software packages on their system which halts the functioning of OS and slow it down.

Fragmentation

Fragmentation in the computer is a state when storage memory breaks into pieces. Internal fragmentation occurs when the method of process is larger than the memory size. External fragmentation occurs when the method or process eliminates. So, these are the following advantages and disadvantages of the operating system. If you have any other queries or related doubt, then please let us know in the comment box below.

6. Describe the structure of OS with suitable example.

Answer

a. Simple Structure

Such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system. In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails. These started as small systems and rapidly expanded much further than their scope. A common example of this is MS-DOS. It was designed simply for a niche amount for people. There was no indication that it would become so popular.

b. Layered structure

An OS can be broken into pieces and retain much more control on system. In this structure the OS is broken into number of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower-level layers only. This simplifies the debugging process as if lower-level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower-level layers have already been debugged. The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system. Moreover, careful planning of the layers is necessary as a layer can use only lower-level layers. UNIX is an example of this structure.

c. Micro-kernel

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernel called the micro-kernel.

Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus, it is more secure and reliable as if a service fails then rest of the operating system remains untouched. Mac OS is an example of this type of OS.

d. Modular structure

It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module.

7. Define kernel. Describe different types of kernel with suitable example.

Answer

Kernel is central component of an operating system that manages operations of computer and hardware. It basically manages operations of memory and CPU time. It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls. It is responsible for various tasks such as disk management, task management, and memory management.

There are different types of kernel are as follows:

a. Monolithic kernel

It is one of types of kernel where all operating system services operate in kernel space. It has dependencies between systems components. It has huge lines of code which is complex. Example: Linux, Unix, etc.

b. Micro kernel

It is kernel type which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space. Example: Mach, Mimix, K42, etc.

c. Hybrid kernel

It is the combination of both monolithic kernel and microkernel. It has speed and design of monolithic kernel and modularity and stability of microkernel. Example: Windows NT, Netware, etc.

d. Exo kernel

It is the type of kernel which follows end-to-end principle. It has fewest hardware abstractions as possible. It allocates physical resources to applications. Examples: Nemesis, ExOS, etc.

e. Nano kernel

It is also the type of kernel that offers hardware abstraction but without system services. Micro kernel also does not have system services therefore the Micro Kernel and does not have system services therefore the Micro kernel and Nano Kernel have become analogous. Example: EROS, etc.

8. What is the use of kernel in OS? Explain.

Answer

Uses of kernel are:

a. Process Management
The creation, execution, and termination of processes keep on going inside the system whenever a system is in the ON mode. A process contains all the information about the task that needs to be done. So, for executing any task, a process is created inside the systems. At a time, there are many processes which are in live state inside the system. The management of all these processes is very important to avoid deadlocks and for the proper functioning of the system, and it is handled by the Kernel.

b. Memory management

Whenever a process is created and executed, it occupies memory, and when it gets terminated, the memory can be used again. But the memory should be handled by someone so that the released memory can be assigned again to the new processes. This task is also done by the Kernel. The kernel keeps track about which part of the memory is currently allocated and which part is available for being allocated to the other processes.

c. Device Management
The Kernel also manages all the different devices which are connected to the system, like the Input and Output devices, etc.

d. Interrupt Handling
While executing the processes, there are conditions where tasks with more priority need to be handled first. In these cases, the kernel has to interrupt in-between the execution of the current process and handle tasks with more priority which has arrived in between.

e. I/O Communication

As the Kernel manages all the devices connected to it, so it is also responsible for handling all sorts of input and output that is exchanged through these devices. So, all the information that the system receives from the user and all the output that the user is provided with via different applications is handled by the Kernel.

9. Define shell. How is it different from kernel? Explain.

Answer

A shell is a computer program which exposes an operating system's services to a human user or other programs. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation. It is named a shell because it is the outermost layer around the operating system. A shell is software that provides an interface between users and operating system of a computer to access the services of a kernel.

Shell is different from kernel:

A shell is an environment or a special user program which provide an interface to user to use operating system services. It executes programs based on the input provided by the user. It is the interface between kernel and user. Its types are – Bourne Shell, C shell, Korn Shell, etc.

Kernel is the heart and core of an Operating System that manages operations of computer and hardware. It acts as a bridge between the user and the resources of the system by accessing various computer resources like the CPU, I/O devices and other resources. It is the core of the operating system. Its types are – Monolithic Kernel, Micro kernel, Hybrid kernel, etc.

10.Describe Client-Server model and Virtual Machine.

Answer

Client-Server model

Client-server architecture, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

Virtual machine

A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. Virtual Machine abstracts the hardware of our personal computer such as CPU, disk drives, memory, NIC (Network Interface Card) etc. into many different execution environments as per our requirements, hence giving us a feel that each execution environment is a single computer. For example, VirtualBox. When we run different processes on an operating system, it creates an illusion that each process is running on a different processor having its own virtual memory, with the help of CPU scheduling and virtual-memory techniques. There are additional features of a process that cannot be provided by the hardware alone like system calls and a file system. VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.

Unit 2 and 3

1. What is process control block? Explain operations on process.

Answer

Process Control Block is a data structure that contains information of the process related to it. The process control block is also known as a task control block, entry of the process table, etc. It is very important for process management as the data structuring for processes is done in terms of the PCB. It also defines the current state of the operating system. There are many operations that can be performed on processes. Some of these are process creation, process preemption, process blocking, and process termination. These are given in detail as follows:

Process creation: Process creation can be done by many events like; User request for process creation, System initialization, Execution of a process creation system call by a running process, Batch job initialization. Process creation means the construction of a new process for the execution. This might be performed by system, user or old process itself.

Process preemption: An interrupt mechanism is used in preemption that suspends the process executing currently and the next process to execute is determined by the short-term scheduler. Preemption makes sure that all processes get some CPU time for execution.

Process Blocking: The process is blocked if it is waiting for some event to occur. This event may be I/O as the I/O events are executed in the main memory and don't require the processor. After the event is complete, the process again goes to the ready state. This basically is a mode where process waits for input-output.

Termination: After the process has completed the execution of its last instruction, it is terminated. The resources held by a process are released after it is terminated. Its is the activity of ending process. Termination can take place because of following reasons: fully execution of the program, termination due to error, problem in hardware and one process can be terminated by another.

2. Define the term semaphore. How does semaphore help in dining philosopher problem?

Answer

Semaphores are integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

The Dining Philosopher Problem: The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pick up the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both.

Solution to The Dining Philosopher Problem using Semaphore: There are three states of the philosopher: THINKING, HUNGRY, and EATING. Here there are two semaphores: Mutex and a semaphore array for the philosophers. Mutex is used such that no two philosophers may access the pickup or putdown at the same time. The array is used to control the behavior of each philosopher. But semaphores can result in deadlock due to programming errors.

3. What is process? How does it differ from program? Explain.

Answer

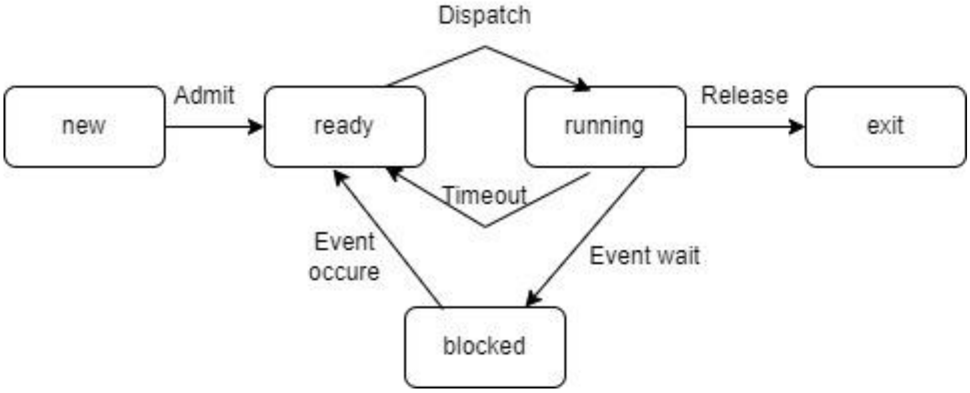
The term process refers to program code that has been loaded into a computer’s memory so that it can be executed by the central processing unit. A process can be described as an instance of a program running on a computer or as an entity that can be assigned to and executed on a processor. A program becomes a process when loaded into memory and thus is an active entity. Difference between process and program:

a. The process and program are both related terms to each other, and the lifespan of the program is longer compared to the process. A batch operating system is referred to as sequential executing instructions/jobs, whereas in a real-time operating system, it is referred to as a program.
b. A program is a collection of sequential and ordered operations that should be executed. In contrast, a process is an example of a program being executed.
c. In the process, the resources need is much higher. It may require processing, memory, input/output resources for the successful execution. In contrast, a program only needs storage memory.
d. A program is passive in nature because it does nothing until it is executed, while a process is dynamic or active because it is an instance of executing a program and performing a specific action.
e. The process has various resources such as disk, printer, memory address, whereas the program requires memory space on the disk to store all instructions.

4. What are different process models? Draw a state(block)diagram of process with different state and explain each briefly.

Answer

In process model, all the runnable software on the computer, is organized into a number of sequential processes. Each process has its own virtual Central Processing Unit. The real Central Processing Unit (CPU) switches back and forth from process to process. This work of switching back and forth is called multiprogramming. A process is basically an activity. It has a program, input, output, and a state.



a. **New:** In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by OS to create the process.
b. **Ready:** New -> Ready to run. After the creation of a process, the process enters the ready state i.e., the process is loaded into the main memory. The process here is ready to run and is waiting to get the CPU time for its execution. Processes that are ready for execution by the CPU are maintained in a queue for ready processes.
c. **Running:** The process is chosen by CPU for execution and the instructions within the process are executed by any one of the available CPU cores.
d. **Blocked:** Whenever the process requests access to I/O or needs input from the user or needs access to a critical region (the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to the ready state.
e. **Exit:** When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.

5. Describe how multithreading improve performance over a single-threaded solution.

Answer

Advantages of Multithreaded Processes A single application can have different threads within the same address space using resource sharing. In a multiprocessor architecture, each thread can run on a different processor in parallel using multithreading. This increases concurrency of the system. All the threads of a process share its resources such as memory, data, files etc. A single application can have different threads within the same address space using resource sharing. It is more economical to use threads as they share the process resources. Comparatively, it is more expensive and time consuming to create processes as they require more memory and resources. Program responsiveness allows a program to run even if part of it is blocked using multithreading. This can also be done if the process is performing a lengthy operation. In a multiprocessor architecture, each thread can run on a different processor in parallel using multithreading. This increases concurrency of the system. This is in direct contrast to a single processor system, where only one process or thread can run on a processor at a time.

6. Differentiate between kernel level thread and user level thread. Which one has a better performance?

Answer

Sn	Kernel level thread	Sn	User level thread
i.	Kernel threads are implemented by OS.	i.	User thread are implemented by users.
ii.	Kernel threads are recognized by OS.	ii.	OS doesn’t recognize user level threads.
iii.	Implementation of Kernel thread is complicated.	iii.	Implementation of User threads is easy.
iv.	Context switch time is more.	iv.	Context switch time is less.
v.	Hardware support is needed.	v.	Context switch requires no hardware support.
vi.	If one kernel thread performs blocking operation then another thread can continue execution.	vi.	If one user level thread perform blocking operation then entire process will be blocked.
vii.	Kernel level threads are designed as independent threads.	vii.	User level threads are designed as dependent threads.
viii.	Example: Window Solaris.	viii.	Example: Java thread, POSIX threads

User level thread have better performance as it is easy to implement, fast, can run on any operating system.

7. Differentiate between process and thread.

Answer

Sn	Process	Sn	Thread
i.	Process means any program is in execution.	i.	Thread means segment of a process.
ii.	Process takes more time to terminate.	ii.	Thread takes less time to terminate.
iii.	It takes more time for creation.	iii.	It takes less time for creation.
iv.	Process is less efficient in term of communication.	iv.	Thread is more efficient in term of communication.
v.	Multi programming holds the concepts of multi process.	v.	We don't need multi programs in action for multiple threads because a single process consists of multiple threads.
vi.	Process is called heavy weight process.	vi.	Thread is lightweight as each thread in a process shares code, data and resources.

8. What resources are used when a thread is created? How do they differ from those used when a process is created?

Answer

A process must have at least one thread. Therefore, creating a process means creating the process and creating a thread. Since a thread is a part of the process, no additional resources are used when a thread is created, instead, it shares the memory space of the process from which this particular thread has been created.

When a thread is created the threads does not require any new resources to execute the thread shares the resources like memory of the process to which they belong to. The benefit of code sharing is that it allows an application to have several different threads of activity all within the same address space. Whereas if a new process creation is very heavyweight because it always requires new address space to be created and even if they share the memory then the inter process communication is expensive when compared to the communication between the threads.

9. Compare the use of monitor and semaphore operations.

Answer

The main difference between Semaphore and Monitor is that Semaphore is an integer variable that performs wait () and signal () operations, while Monitor is an abstract data type that allows only one process to use the shared resource at a time.

a) Semaphore is a variable used to control access to a common resource by multiple processes in a concurrent system such as a multitasking operating system. In contrast, monitor is a synchronization construct that allows threads to have both mutual exclusion and the ability to wait(block) for a certain condition to become true. Thus, this is the main difference between semaphore and monitor.
b) Another difference between semaphore and monitor is that Semaphore is an integer variable while Monitor is an abstract data type.
c) There is no condition variable concept in semaphores whereas monitor has condition variables.
d) When a process requires access to the semaphore, it performs wait () and performs signal () when releasing the resource. On the other hand, a process uses procedures to access the shared variable in the monitor. Hence, this is also a difference between semaphore and monitor.

10.Define mutual exclusion and critical section.

Answer

Mutual exclusion is a property of concurrency control, which is instituted for the purpose of preventing race conditions. Example: It is some way of making sure that if one process is using a shared variable or file, the other process will be excluded from doing the same things. Mutual exclusion can be defined as property for process synchronization which states “no two processes can exist in the critical section at any given point of time”.

Critical Section is the part of a program which tries to access shared resources. That resource may be any resource in a computer like a memory location, Data structure, CPU or any IO device. The critical section cannot be executed by more than one process at the same time; operating system faces the difficulties in allowing and disallowing the processes from entering the critical section. The critical section problem is used to design a set of protocols which can ensure that the Race condition among the processes will never arise.