What is Swapping?

Swapping is a memory management scheme that temporarily swaps out an idle or blocked process from the main memory to secondary memory which ensures proper memory utilization and memory availability for those processes which are ready to be executed.

Swapping in **OS** is done to get access to data present in secondary memory and transfer it to the main memory so that it can be used by the application programs.

The two important concepts in the process of swapping are:

Swap In: The method of removing a process from secondary memory (**Hard Drive**) and restoring it to the main memory (**RAM**) for execution is known as the Swap In method.

Swap Out: The method of bringing out a process from the main memory(**RAM**) and sending it to the secondary memory(**hard drive**) so that the processes with higher priority will be executed is known as the Swap Out method.

Swapping	Paging
It is the procedure of copying out the entire process.	It is a technique of memory allocation.
Swapping occurs when whole process is transferred to the disk.	Paging occurs when some part of the process is transferred to the disk.
In this, a process is swapped temporarily from main memory to secondary memory.	In this. the contiguous block of memory is made non-contiguous but of fixed size called frame or pages.
Swapping can be performed without any memory management.	It is a concept used in Non-contiguous Memory Management.
Swapping is done by inactive processes.	The only active process can perform paging.
Swapping is relatively slower.	Paging is relatively faster than swapping.

What is a process?

A process is defined as a sequence of instructions are executed in a **predefined order.** A process is managed by the Process Control Block (PCB).

A process in OS consists of 4 important sections:

1. **Stack** - Temporary data such as method/function arguments, return address, and local variables are stored in the stack.

2. **Heap** - Memory that is dynamically allocated to a process during its execution. Contains dynamically allocated content.
3. **Data** - This section contains the global and static variables.
4. **Text** - Consists of the current activity as reflected by the Program Counter value and the processor's registers contents.

Explain FIFO and LIFO

FIFO (First In First Out) page replacement algorithm is a simple page replacement algorithm that replaces the page that has been in memory for the longest time.

In FIFO, the page that was brought into the memory first will be the first one to be replaced when a page fault occurs and a new page needs to be loaded. The operating system keeps track of all pages that are currently loaded in memory and adds new pages to the end of the queue as they are loaded. When a page fault occurs, the operating system simply replaces the oldest page in the queue.

LIFO (Last In First Out) page replacement algorithm is a page replacement algorithm that is similar to the FIFO page replacement algorithm, with the difference being that it replaces the page that was brought into memory most recently, instead of the page that has been in memory the longest.

In LIFO, the operating system keeps a stack of the pages that are currently loaded in memory, and adds new pages to the top of the stack as they are loaded. When a page fault occurs and a new page needs to be loaded, the operating system simply replaces the page at the top of the stack, which is the page that was brought into memory most recently.

List various functions of OS.

- Memory Management
- Processor Management
- Device Management
- File Management
- Network Management
- Acts as user interface
- Booting
- Provides Security
- Controls System Performance
- Job Accounting
- Error Detection
- Coordinates other software

What is Context Switching?

Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier. This is a feature of a multitasking operating system and allows a single CPU to be shared by multiple processes.

The three major triggers for context switching are:

- **Multitasking:** In a multitasking environment, a process is switched out of the CPU so another process can be run. The state of the old process is saved and the state of the new process is loaded.

- **Interrupt Handling:** The hardware switches a part of the context when an interrupt occurs.
- **User and Kernel Mode Switching:** A context switch may take place when a transition between the user mode and kernel mode is required in the operating system.

What is Multitasking?

Multitasking: Multitasking allows a user to perform more than one process simultaneously. These processes share similar processing resources like a **CPU**. The operating system keeps track of where you are in each of these jobs and allows you to transition between them without losing data.

Multitasking operating systems can be divided into three categories:

1. **Preemptive Multitasking:** It determines how much time one job spends on the operating system before assigning another process to use it.
2. **Cooperative Multitasking:** It's goal is to complete the current work while allowing another process to execute.
3. **True Multitasking:** A true Multitasking operating system has the ability to conduct several tasks simultaneously while underlying the hardware or software.

What are Real Time Systems?

A real-time system is a time-bound system with well-defined and fixed time constraints, and processing must be done within the defined constraints; otherwise, the system will fail. Real-Time System is used at those places where we require higher and timely responses. A **real-time operating system (RTOS)** is an operating system that needs strict completion deadlines for all the tasks that need to be performed on it.

The Real-Time Operating System is of three types:

1. **Hard Real-Time Operating system:** A hard real-time operating system is used when we need to complete tasks by a given deadline. If the task is not completed on time, then the system is considered to be failed. e.g. pacemaker
2. **Soft Real-Time Operating System:** A soft real-time operating system is used where few delays in time duration are acceptable. e.g. telephone switches
3. **Firm Real-Time Operating System:** A firm real-time system is one in which a few missed deadlines will not lead to total failure, but missing more than a few may lead to complete or catastrophic system failure. e.g. Video Conferencing

State in brief the four key features of each of the following types of OS: (i) Real-time (ii) Distributed (iii) Multiprogramming (iv) Time-sharing.

Real Time OS

- The response time of RTOS is highly predictable. That is we can guess the time system will take to complete the task.
- All the interrupt requests go to the Kernel. The kernel is responsible for taking all the decisions.
- RTOS occupies very less space. Since the size of programs used in RTOS is small.

- RTOS consumes very less resources. Since maximum utilization of resources happens in RTOS, we can also start our system with fewer resources.

Distributed OS

1. **Concurrency:** A distributed operating system allows multiple processes to execute simultaneously on different machines.
2. **Resource sharing:** Resources such as memory, storage, and processing power can be shared across multiple machines.
3. **Communication:** Communication plays a crucial role in distributed operating systems as it is necessary to coordinate the actions of multiple machines.
4. **Security:** Security is a critical feature of distributed operating systems since they are exposed to a wide range of security risks.

Multiprogramming OS

1. It allows loading multiple processes for concurrent execution.
2. It ensures efficient process memory allocation and deallocation.
3. It implements CPU scheduling algorithms to allocate CPU time to various processes.
4. It is also responsible for Input/output Device Management.

Time Sharing OS

1. Multiple users can use the same computer simultaneously
2. End users believe they have complete control over the computer system.
3. Interaction among users and computers is improved.
4. It can do a large number of tasks quickly.

Discuss the advantages of Multiprocessor System.

1. **More reliable Systems**: Even if one processor fails, the system will not halt.
2. **Enhanced Throughput**: More work could be done in less time as the number of processors increases.
3. **More Economic Systems**: They are cheaper than single processor systems in the long run because they share the data storage, peripheral devices, power supplies etc.

Differentiate between multiprogramming, multitasking, timesharing and multiprocessing systems.

Feature	Multiprogramming	Multitasking	Time Sharing	Multiprocessing
Definition	Running multiple programs on a single CPU	Running multiple tasks (applications) on a single CPU	Multiple users use a particular computer system at the same time.	Running multiple processes on multiple CPUs (or cores)
Resource Sharing	Resources (CPU, memory) are shared among programs	Resources (CPU, memory) are shared among tasks	Resources (CPU, memory) are shared among users	Each process has its own set of resources (CPU, memory)
Context Switching	Requires a context switch to switch between programs	Requires a context switch to switch between tasks	Requires a context switch to switch between users	Requires a context switch to switch between processes
Advantages	Faster response time, Higher CPU utilisation	Virtual Memory, Reliability	Quick response time, User-friendly	Increased throughput, Reliability
Disadvantages	CPU scheduling required, Memory management required	CPU Heat Up, Memory Boundation	Reliability problem, Data communication problem	Increased complexity, Higher power consumption

What is a semaphore? What is its use?

Semaphores are variables used to coordinate the activities of multiple processes. They are used to enforce mutual exclusion, avoid race conditions and implement synchronization between processes.

Semaphores are of two types:

1. **Binary Semaphore:** It is also known as a mutex lock. It can have only two values 0 and 1. Its value is initialized to 1. It is used to implement the solution of critical section problems with multiple processes.
2. **Counting Semaphore:** Its value can range over an unrestricted domain. It is used to control access to a resource that has multiple instances.

Uses

- It is used to implement critical sections in a program.
- It is used to avoid race conditions.
- It is used to implement synchronization between processes.
- It is used to enforce mutual exclusion

- It provides a flexible and robust way to manage shared resources.

What is a Deadlock? How is it prevented? Explain Deadlock avoidance algorithm.

When two or more processes try to access the critical section at the same time and they fail to access simultaneously or stuck while accessing the critical section then this condition is known as Deadlock.

Deadlock Prevention

- **Eliminate Mutual Exclusion:** We cannot prevent deadlocks by denying the mutual exclusion condition because some resources, such as the tape drive and printer, are inherently non-shareable.
- **Eliminate Hold and wait:** Two protocols to prevent the Hold and Wait are:
    1. Allocate all required resources to the process before the beginning of its execution, but it will lead to low device utilization
    2. The process will make a new request for resources after releasing the current set of resources, but this solution may lead to starvation.
- **Eliminate No Preemption:** Preempt resources from the process when other high-priority processes require resources.
- **Eliminate Circular Wait:** A number will be allocated to each resource. A process may request that the number of resources be increased or decreased. For Example, if process P1 is given R5 resources, the next time process P1 requests R4, R3, or any other resource less than R5, the request will be denied. Only requests for resources more than R5 will be granted.

Deadlock Avoidance

Resource Allocation Graph

The resource allocation graph (RAG) is used to visualize the system's current state as a graph. The Graph includes all processes, the resources that are assigned to them, as well as the resources that each Process requests. Sometimes, if there are fewer processes, we can quickly spot a deadlock in the system by looking at the graph.

Banker's Algorithm

Safety Algorithm

The algorithm for finding out whether or not a system is in a safe state can be described as follows:

1. Let *Work* and *Finish* be vectors of length m and n, respectively. Initialize
   $Work = Available$ and $Finish[i] = false$ for $i = 0, 1, \ldots, n - 1$.
2. Find an index i such that both
   a. $Finish[i] == false$
   b. $Need_i \leq Work$

   If no such i exists, go to step 4.
3. $Work = Work + Allocation_i$
   $Finish[i] = true$
   Go to step 2.
4. If $Finish[i] == true$ for all i, then the system is in a safe state.

Resource Request Algorithm

Let $Request_i$ be the request array for process P_i. $Request_i[j] = k$ means process P_i wants k instances of resource type R_j. When a request for resources is made by process P_i, the following actions are taken:

1. If $Request_i \leq Need_i$, go to step 2. Otherwise, raise an error condition, since the process has exceeded its maximum claim.
2. If $Request_i \leq Available$, go to step 3. Otherwise, P_i must wait, since the resources are not available.
3. Have the system pretend to have allocated the requested resources to process P_i by modifying the state as follows:

$$Available = Available - Request_i;$$
$$Allocation_i = Allocation_i + Request_i;$$
$$Need_i = Need_i - Request_i;$$

Explain Segmentation with paging.

Segmentation with Paging (Segmented Paging): It is a memory management technique that divides the physical memory into pages, and then maps each logical address used by a process to a physical page. In this approach, segments are used to map virtual memory addresses to physical memory addresses

Advantages of Segmented Paging

1. The page table size is reduced
2. Reduces external fragmentation
3. Swapping out into virtual memory becomes easier .

Disadvantages of Segmented Paging

1. Internal fragmentation is present.
2. Extra hardware is required

Write short notes on: (a)(i) C-SCAN (ii) FIFO. (b) Virtual Memory. (c) Windows 2000.

C-SCAN Disk Scheduling Algorithm

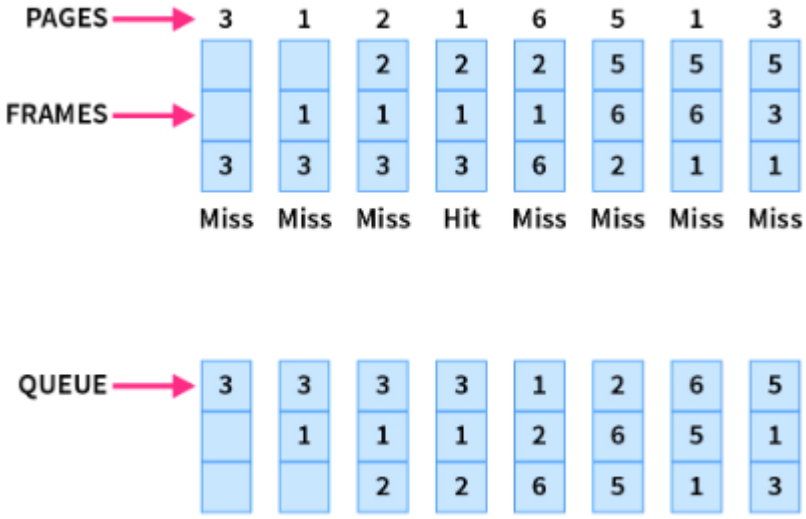
1. Arrange all the I/O requests in ascending order.
2. The head will start moving in the right direction, i.e., from 0 to the size of the block.
3. As soon as a request is encountered, head movement is calculated as the current request - the previous request.
4. This process is repeated until the head reaches the end of the disk and the head movements are added.
5. As the end is reached, the head will go from right to left without processing any requests.
6. As the head reaches the beginning of the disc, i.e., 0, the head again starts moving in the right direction, and head movement keeps on adding.
7. This process is completed as soon as all the requests are processed, and we get total head movement.

FIFO Page Replacement Algorithm

In this, we maintain a queue of all the pages that are in the memory currently. The oldest page in the memory is at the front-end of the queue and the most recent page is at the back or rear-end of the queue.

Whenever a page fault occurs, the operating system looks at the front-end of the queue to know the page to be replaced by the newly requested page. It also adds this newly requested page at the rear-end and removes the oldest page from the front-end of the queue.

Example: Consider the page reference string as 3, 1, 2, 1, 6, 5, 1, 3 with 3-page frames. Let's try to find the number of page faults:



- Initially, all of the slots are empty so page faults occur at 3,1,2.

Page faults = 3

- When page 1 comes, it is in the memory so no page fault occurs.

Page faults = 3

- When page 6 comes, it is not present and a page fault occurs. Since there are no empty slots, we **remove the front of the queue, i.e 3.**

Page faults = 4

- When page 5 comes, it is also not present and hence a page fault occurs. The front of the queue i.e **1 is removed.**

Page faults = 5

- When page 1 comes, it is not found in memory and again a page fault occurs. The front of the queue i.e **2 is removed.**

Page faults = 6

- When page 3 comes, it is again not found in memory, a page fault occurs, and **page 6 is removed** being on top of the queue

Total page faults = 7

Belady's anomaly: Belady's anomaly refers to the phenomena where increasing the number of frames in memory, increases the page faults as well.

Virtual Memory

Virtual memory is a part of the system's secondary memory that acts and gives us a feel as if it is a part of the main memory. Virtual memory allows a system to execute heavier applications or multiple applications simultaneously without exhausting the RAM.

When running multiple heavy applications at once, the system's RAM may get overloaded. To mitigate this issue, some data stored in RAM that isn't being actively used can be temporarily relocated to virtual memory. This frees up RAM space, which may then be utilized to store data that the system will need to access.

A system can continue to run smoothly with significantly less physical RAM than it would normally require. This can be done by exchanging data between RAM and virtual memory based on its need.

Windows 2000

Features of Windows 2000:

- **Multiprocessing:** Windows 2000 can run on systems with multiple processors.
- **Virtual memory:** Windows 2000 supports virtual memory, which allows processes to access more memory than is physically available.
- **Security:** Windows 2000 has a strong security model, which helps to protect data from unauthorized access.
- **Network support:** Windows 2000 supports a wide range of network protocols, including TCP/IP and NetBEUI.

Advantages of Windows 2000:

- **Stability and Reliability:** It was designed with the robustness needed for business environments, making it a popular choice for organizations.
- **Improved Active Directory:** Windows 2000 introduced Active Directory. This feature made it easier to manage large-scale networks and improve security.
- **Enhanced Security:** Windows 2000 offered improved security features compared to previous Windows versions.

Disadvantages of Windows 2000:

- **Limited Hardware and Software Compatibility:** Some software applications were not fully compatible with Windows 2000, leading to potential compatibility issues.
- **Steeper Learning Curve:** Its interface and management tools were more complex compared to Windows XP, requiring a steeper learning curve for novice users.
- **Lack of Multimedia Features:** This is a disadvantage for users who required advanced multimedia capabilities.