**Model Question Paper of Operating System**

**MCA Semester I**

1. **What is meant by process scheduling?**

The act of determining which process in the ready state should be moved to the running state is known as Process Scheduling. The prime aim of the process scheduling system is to keep the CPU busy all the time and to deliver minimum response time for all programs. For achieving this, the scheduler must apply appropriate rules for swapping processes IN and OUT of CPU.

Schedulers fell into one of the two general categories:

**Non pre-emptive scheduling.** When the currently executing process gives up the CPU voluntarily.

**Pre-emptive scheduling.** When the operating system decides to favour another process, pre-empting the currently executing process.

**Types of Schedulers**

There are three types of schedulers available :

1. **Long Term Scheduler** :

Long term scheduler runs less frequently. Long Term Schedulers decide which program must get into the job queue. From the job queue, the Job Processor, selects processes and loads them into the memory for execution. Primary aim of the Job Scheduler is to maintain a good degree of Multiprogramming. An optimal degree of Multiprogramming means the average rate of process creation is equal to the average departure rate of processes from the execution memory.

1. **Short Term Scheduler** :

This is also known as CPU Scheduler and runs very frequently. The primary aim of this scheduler is to enhance CPU performance and increase process execution rate.

1. **Medium Term Scheduler** :

This scheduler removes the processes from memory (and from active contention for the CPU), and thus reduces the degree of multiprogramming. At some later time, the process can be reintroduced into memory and its execution van be continued where it left off. This scheme is called swapping. The process is swapped out, and is later swapped in, by the medium term scheduler.

1. **What are the security threats to operating system?**

Security refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system. If a computer program is run by an unauthorized user, then he/she may cause severe damage to computer or data stored in it. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc.

* Authentication
* One Time passwords
* Program Threats
* System Threats

## Authentication

Authentication refers to identifying each user of the system and associating the executing programs with those users. It is the responsibility of the Operating System to create a protection system which ensures that a user who is running a particular program is authentic. Operating Systems generally identifies/authenticates users using following three ways −

* **Username / Password** − User need to enter a registered username and password with Operating system to login into the system.
* **User card/key** − User need to punch card in card slot, or enter key generated by key generator in option provided by operating system to login into the system.
* **User attribute - fingerprint/ eye retina pattern/ signature** − User need to pass his/her attribute via designated input device used by operating system to login into the system.

## One Time passwords

One-time passwords provide additional security along with normal authentication. In One-Time Password system, a unique password is required every time user tries to login into the system. Once a one-time password is used, then it cannot be used again. One-time password are implemented in various ways.

* **Random numbers** − Users are provided cards having numbers printed along with corresponding alphabets. System asks for numbers corresponding to few alphabets randomly chosen.
* **Secret key** − User are provided a hardware device which can create a secret id mapped with user id. System asks for such secret id which is to be generated every time prior to login.
* **Network password** − Some commercial applications send one-time passwords to user on registered mobile/ email which is required to be entered prior to login.

## Program Threats

Operating system's processes and kernel do the designated task as instructed. If a user program made these process do malicious tasks, then it is known as Program Threats. One of the common example of program threat is a program installed in a computer which can store and send user credentials via network to some hacker. Following is the list of some well-known program threats.

* **Trojan Horse** − Such program traps user login credentials and stores them to send to malicious user who can later on login to computer and can access system resources.
* **Trap Door** − If a program which is designed to work as required, have a security hole in its code and perform illegal action without knowledge of user then it is called to have a trap door.
* **Logic Bomb** − Logic bomb is a situation when a program misbehaves only when certain conditions met otherwise it works as a genuine program. It is harder to detect.
* **Virus** − Virus as name suggest can replicate themselves on computer system. They are highly dangerous and can modify/delete user files, crash systems. A virus is generatlly a small code embedded in a program. As user accesses the program, the virus starts getting embedded in other files/ programs and can make system unusable for user

## System Threats

System threats refers to misuse of system services and network connections to put user in trouble. System threats can be used to launch program threats on a complete network called as program attack. System threats creates such an environment that operating system resources/ user files are misused. Following is the list of some well-known system threats.

* **Worm** − Worm is a process which can choked down a system performance by using system resources to extreme levels. A Worm process generates its multiple copies where each copy uses system resources, prevents all other processes to get required resources. Worms processes can even shut down an entire network.
* **Port Scanning** − Port scanning is a mechanism or means by which a hacker can detects system vulnerabilities to make an attack on the system.
* **Denial of Service** − Denial of service attacks normally prevents user to make legitimate use of the system. For example, a user may not be able to use internet if denial of service attacks browser's content settings.

1. **Explain process control Block?**

Process control block (PCB) is a data structure which is associated with any process and provides all the complete information about that process. The process control block is "the manifestation of a process in an operating system". Process control block is important in multiprogramming environment as it captures the information pertaining to the number of processes running simultaneously.

## Role of process control block

The role or work of **process control block** (PCB) in process management is that it can access or modified by most OS utilities including those are involved with memory, scheduling, and input / output resource access. It can be said that the set of the process control blocks give the information of the current state of the operating system. Data structuring for processes is often done in terms of process control blocks. For example, pointers to other process control blocks inside any process control block allows the creation of those queues of processes in various scheduling states.

The following are the various information that is contained by process control block:

* Naming the process
* State of the process
* Resources allocated to the process
* Memory allocated to the process
* Scheduling information
* Input / output devices associated with process

Components of Process control block

The following are the various components that are associated with the process control block PCB:

**1. Process ID:**  
In computer system there are various process running simultaneously and each process has its unique ID. This Id helps system in scheduling the processes. This Id is provided by the process control block.  
In other words, it is an identification number that uniquely identifies the processes of computer system.

**2. Process state:**  
As we know that the process state of any process can be New, running, waiting, executing, blocked, suspended, terminated. Process control block is used to define the process state of any process. In other words, process control block refers the states of the processes.

**3. Program counter:**  
Program counter is used to point to the address of the next instruction to be executed in any process. This is also managed by the process control block.

**4. Register Information:**  
This information is comprising with the various registers, such as index and stack that are associated with the process. This information is also managed by the process control block.

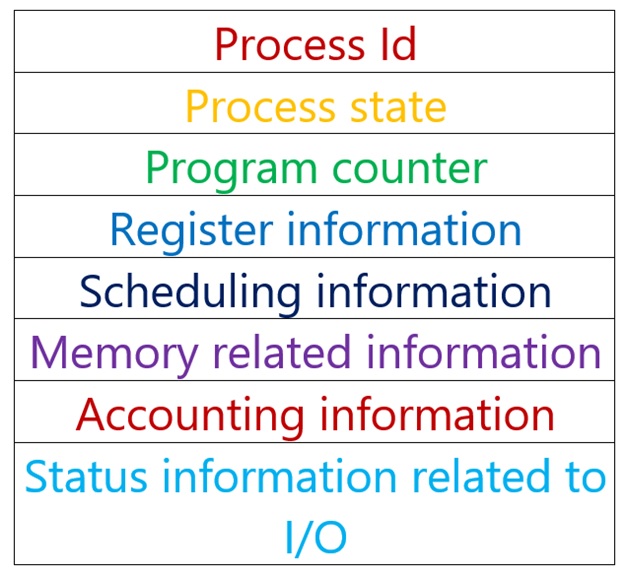
**5. Scheduling information:**  
Scheduling information is used to set the priority of different processes. This is very useful information which is set by the process control block. In computer system there were many processes running simultaneously and each process have its priority. The priority of primary feature of RAM is higher than other secondary features. Scheduling information is very useful in managing any computer system.

**6. Memory related information:**  
This section of the process control block comprises of page and segment tables. It also stores the data contained in base and limit registers.

**7. Accounting information:**  
This section of process control block stores the details relate to central processing unit (CPU) utilization and execution time of a process.

**8. Status information related to input / output:**  
This section of process control block stores the details pertaining to resource utilization and file opened during the process execution.

The operating system maintains a table called process table, which stores the process control blocks related to all the processes.



1. **Explain Paging?**

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

* Logical Address or Virtual Address (represented in bits): An address generated by the CPU
* Logical Address Space or Virtual Address Space( represented in words or bytes): The set of all logical addresses generated by a program
* Physical Address (represented in bits): An address actually available on memory unit
* Physical Address Space (represented in words or bytes): The set of all physical addresses corresponding to the logical addresses

The mapping from virtual to physical address is done by the memory management unit (MMU) which is a hardware device and this mapping is known as paging technique.

* The Physical Address Space is conceptually divided into a number of fixed-size blocks, called frames.
* The Logical address Space is also splitted into fixed-size blocks, called pages.
* Page Size = Frame Size

Let us consider an example:

* Physical Address = 12 bits, then Physical Address Space = 4 K words
* Logical Address = 13 bits, then Logical Address Space = 8 K words
* Page size = frame size = 1 K words (assumption)

Address generated by CPU is divided into

* **Page number(p):** Number of bits required to represent the pages in Logical Address Space or Page number
* **Page offset(d):** Number of bits required to represent particular word in a page or page size of Logical Address Space or word number of a page or page offset.

Physical Address is divided into

* **Frame number(f):** Number of bits required to represent the frame of Physical Address Space or Frame number.
* **Frame offset(d):** Number of bits required to represent particular word in a frame or frame size of Physical Address Space or word number of a frame or frame offset.

1. **Differentiate between Network Operating system and Distributed Operating System?**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Network Operating System** | **Distributed Operating System** |
| 1 | A network operating system is made up of software and associated protocols that allow a set of computer network to be used together. | A distributed operating system is an ordinary centralized operating system but runs on multiple independent CPUs. |
| 2 | Environment users are aware of multiplicity of machines. | Environment users are not aware of multiplicity of machines. |
| 3 | Control over file placement is done manually by the user. | It can be done automatically by the system itself. |
| 4 | Performance is badly affected if certain part of the hardware starts malfunctioning. | It is more reliable or fault tolerant i.e distributed operating system performs even if certain part of the hardware starts malfunctioning. |
| 5 | Remote resources are accessed by either logging into the desired remote machine or transferring data from the remote machine to user's own machines. | Users access remote resources in the same manner as they access local resources. |

1. **What are semaphores?**

Semaphore is a simply a variable. This variable is used to solve critical section problem and to achieve process synchronization in the multi processing environment.  
The two most common kinds of semaphores are counting semaphores and binary semaphores. Counting semaphore can take non-negative integer values and Binary semaphore can take the value 0 & 1. only.

**Some point regarding P and V operation**

1. P operation is also called wait, sleep or down operation and V operation is also called signal, wake-up or up operation.
2. Both operations are atomic and semaphore(s) is always initialized to one.
3. A critical section is surrounded by both operations to implement process synchronization.

Let there be two processes P1 and P2 and a semaphore s is initialized as 1. Now if suppose P1 enters in its critical section then the value of semaphore s becomes 0. Now if P2 wants to enter its critical section then it will wait until s > 0, this can only happen when P1 finishes its critical section and calls V operation on semaphore s. This way mutual exclusion is achieved.

The description above is for binary semaphore which can take only two values 0 and 1. There is one other type of semaphore called counting semaphore which can take values greater than one.

Now suppose there is a resource whose number of instance is 4. Now we initialize S = 4 and rest is same as for binary semaphore. Whenever process wants that resource it calls P or wait function and when it is done it calls V or signal function. If value of S becomes zero than a process has to wait until S becomes positive. For example, Suppose there are 4 process P1, P2, P3, P4 and they all call wait operation on S(initialized with 4). If another process P5 wants the resource then it should wait until one of the four process calls signal function and value of semaphore becomes positive.

**Problem in this implementation of semaphore**

Whenever any process waits then it continuously checks for semaphore value (look at this line while (s==0); in P operation) and waste CPU cycle. To avoid this another implementation is provided below.

|  |
| --- |
| P(Semaphore s)  {      s = s - 1;      if (s < 0) {            // add process to queue          block();      }  }    V(Semaphore s)  {      s = s + 1;      if (s >= 0) {            // remove process p from queue          wakeup(p);      }  } |

In this implementation whenever process waits it is added to a waiting queue of processes associated with that semaphore. This is done through system call block() on that process. When a process is completed it calls signal function and one process in the queue is resumed. It uses wakeup() system call.

1. **Explain Thrashing?**

A process that is spending more time paging than executing is said to be thrashing. In other words it means, that the process doesn't have enough frames to hold all the pages for its execution, so it is swapping pages in and out very frequently to keep executing. Sometimes, the pages which will be required in the near future have to be swapped out.

Initially when the CPU utilization is low, the process scheduling mechanism, to increase the level of multiprogramming loads multiple processes into the memory at the same time, allocating a limited amount of frames to each process. As the memory fills up, process starts to spend a lot of time for the required pages to be swapped in, again leading to low CPU utilization because most of the processes are waiting for pages. Hence the scheduler loads more processes to increase CPU utilization, as this continues at a point of time the complete system comes to a stop.

1. **Explain disk scheduling algorithms?**

Disk schedulingis is done by operating systems to schedule I/O requests arriving for disk. Disk scheduling is also known as I/O scheduling.

Disk scheduling is important because:

* Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by disk controller. Thus other I/O requests need to wait in waiting queue and need to be scheduled.
* Two or more request may be far from each other so can result in greater disk arm movement.
* Hard drives are one of the slowest parts of computer system and thus need to be accessed in an efficient manner.

**Disk Scheduling Algorithms**

1. **FCFS:**FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.

Advantages:

* Every request gets a fair chance
* No indefinite postponement

Disadvantages:

* Does not try to optimize seek time
* May not provide the best possible service

1. **SSTF:** In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system.

Advantages:

* Average Response Time decreases
* Throughput increases

Disadvantages:

* Overhead to calculate seek time in advance
* Can cause Starvation for a request if it has higher seek time as compared to incoming requests
* High variance of response time as SSTF favours only some requests

1. **SCAN:**In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works like an elevator and hence also known as **elevator algorithm.**As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

Advantages:

* High throughput
* Low variance of response time
* Average response time

Disadvantages:

* Long waiting time for requests for locations just visited by disk arm

1. **CSCAN**: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

These situations are avoided in *CSAN*algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Advantages:

* Provides more uniform wait time compared to SCAN

1. **LOOK:** It is similar to the SCAN disk scheduling algorithm except the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.
2. **CLOOK:** As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm inspite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end’s last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.
3. **What are deadlocks? Explain methods of handling deadlock?**

Deadlock - A condition that arises when two or more processes are waiting indefinitely for an event that can only be caused by one of the waiting processes - The events that the processes are waiting for will never happen .

Deadlock can arise if the four conditions hold simultaneously

– Mutual exclusion - only one process at a time can use a resource

- Hold and wait - a process holding at least one resource is waiting to acquire additional resources held by other processes

- No preemption of resources - a resource can be released only voluntarily by the process holding it, after that process has completed its task

- Circular wait - there exists a set {P0, P1, …, Pn} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, …, Pn–1 is waiting for a resource that is held by Pn, and Pn is waiting for a resource that is held by P0

**Methods for Handling Deadlocks**

• To ensure that deadlocks never occur, a system can use either deadlock prevention or deadlock-avoidance

• Deadlock Prevention - ensure that at least one of the four necessary conditions for deadlock cannot hold

• Deadlock Avoidance - requires that the operating system be given comprehensive information about which resources a process will request during its lifetime - Operating system can then make intelligent decisions about when a process should be allocated a resource

1. **Differentiate between threads and process?**

|  |  |  |
| --- | --- | --- |
|  | **Process** | **Thread** |
| Definition | An executing instance of a program is called a process. | A thread is a subset of the process. |
| Process | It has its own copy of the data segment of the parent process. | It has direct access to the data segment of its process. |
| Communication | Processes must use inter-process communication to communicate with sibling processes. | Threads can directly communicate with other threads of its process. |
| Overheads | Processes have considerable overhead. | Threads have almost no overhead. |
| Creation | New processes require duplication of the parent process. | New threads are easily created. |
| Control | Processes can only exercise control over child processes. | Threads can exercise considerable control over threads of the same process. |
| Changes | Any change in the parent process does not affect child processes. | Any change in the main thread may affect the behavior of the other threads of the process. |
| Memory | Run in separate memory spaces. | Run in shared memory spaces. |
| File descriptors | Most file descriptors are not shared. | It shares file descriptors. |
| File system | There is no sharing of file system context. | It shares file system context. |
| Signal | It does not share signal handling. | It shares signal handling. |
| Controlled by | Process is controlled by the operating system. | Threads are controlled by programmer in a program. |
| Dependence | Processes are independent. | Threads are dependent. |

**Prepared By**

**Kavita Lal**

**Guest Faculty**

**MCA Dept**