SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

23ITB203 – PRINCIPLES OF OPERATING SYSTEMS

UNIT – 2 PROCESS MANAGEMENT

Processes - Process Concept - Process Scheduling - Operations on Processes - Inter-process Communication; CPU Scheduling - Scheduling criteria - Scheduling algorithms: Threads - Multithread Models – Threading issues;

PART A

1. **Define thread.**

   Ans : A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.

2. **Write the four situations under which CPU scheduling decisions take place?**

   Ans : 1. When a process switches from the running state to the waiting state (for example, I/O request, or invocation of wait for the termination of one of the child processes)

   2. When a process switches from the running state to the ready state (for example, when an interrupt occurs)

   3. When a process switches from the waiting state to the ready state (for example, completion of I/O)

   4. When a process terminates

3. **What is meant by Dispatch latency?**

   Ans : The time taken by the dispatcher to stop one process and start another running is known as Dispatch Latency.

4. **What is meant by Dispatcher?**

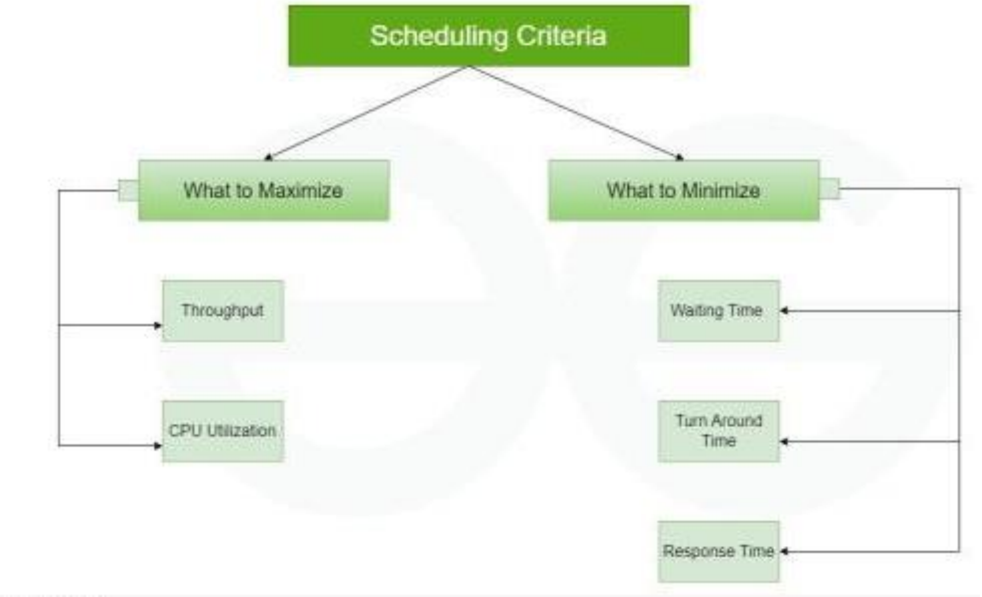
   Ans : It is a module that gives control of the CPU to the process selected by the short-term scheduler. This function involves

   - Switching Context
   - Switching to User Mode
   - Jumping to the proper location in the user program to restart that program

5. **What is meant by Context Switching?**

   Ans : When an interrupt occurs, the operating system saves the status of the interrupted process routes control to the appropriate first level interrupt handler.

6. **What are the various scheduling criteria for CPU scheduling?**



7. Define process and the process states.

A **process** is a program in execution. It consists of the program code, data, and system resources required for execution. A process goes through multiple states during its lifecycle.

Process States:

A process moves through different states during execution. The main states are:

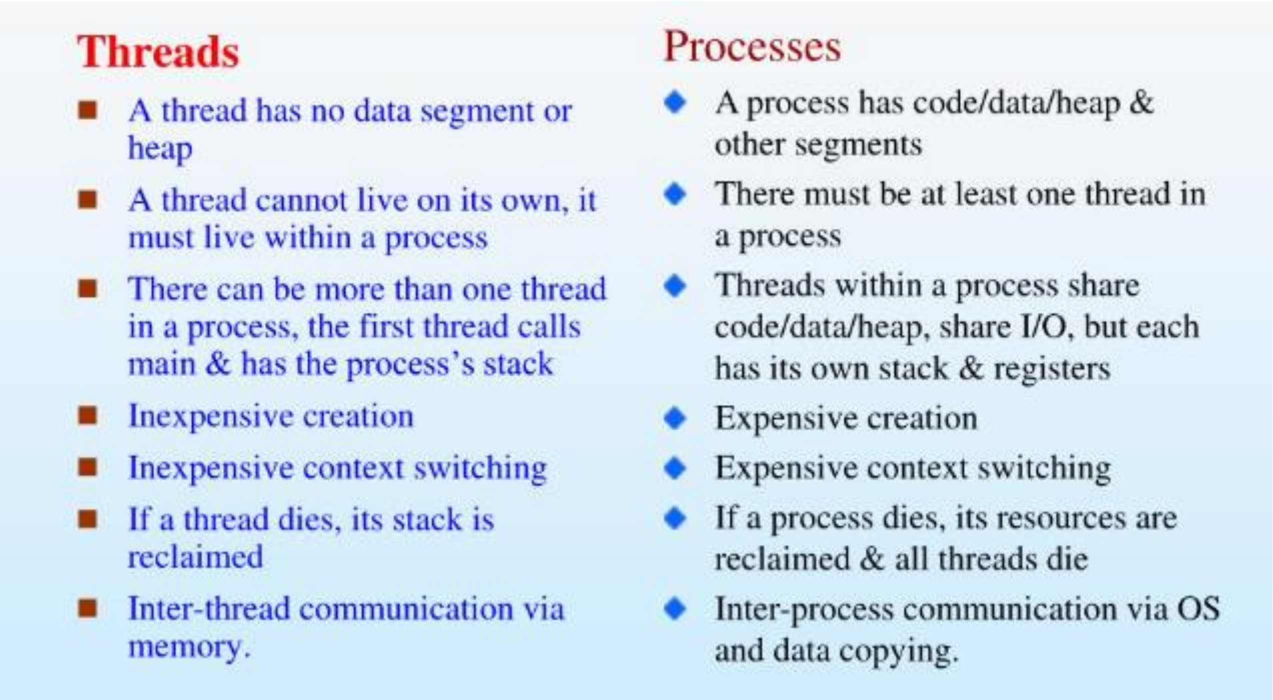
1. **New** – The process is being created.
2. **Ready** – The process is waiting for CPU allocation.
3. **Running** – The process is currently being executed by the CPU.
4. **Waiting** – The process is waiting for an I/O operation or an event to occur.
5. **Terminated** – The process has completed execution or is forcefully stopped.

8. **Differentiate preemptive and non-preemptive scheduling?**

Ans :

Feature	**Preemptive Scheduling**	**Non-Preemptive Scheduling**
Definition	The CPU can be taken from a running process before it completes.	A running process keeps the CPU until it completes or enters a waiting state.
Interruption	A process can be interrupted and moved back to the ready queue.	Once a process starts, it cannot be interrupted.
CPU Utilization	Higher, as the CPU can switch between processes efficiently.	May lead to lower CPU utilization if long processes block shorter ones.
Response Time	Generally better for interactive systems.	May lead to longer response times for some processes.
Complexity	More complex due to context switching.	Simpler to implement.
Example Algorithms	Round Robin (RR), Shortest Remaining Time First (SRTF), Priority Scheduling (preemptive)	First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling (non-preemptive)

9. Differentiate between Threads and Processes.



Threads

- A thread has no data segment or heap
- A thread cannot live on its own, it must live within a process
- There can be more than one thread in a process, the first thread calls main & has the process's stack
- Inexpensive creation
- Inexpensive context switching
- If a thread dies, its stack is reclaimed
- Inter-thread communication via memory.

Processes

- A process has code/data/heap & other segments
- There must be at least one thread in a process
- Threads within a process share code/data/heap, share I/O, but each has its own stack & registers
- Expensive creation
- Expensive context switching
- If a process dies, its resources are reclaimed & all threads die
- Inter-process communication via OS and data copying.

10. **What is meant by CPU-bound process?**

Ans : A CPU-bound process generates I/O requests infrequently using more of its time doing computation than an I/O processes. If all processes are CPU-bound, the I/O waiting queue will almost be empty and the devices will go unused and the system will be unbalanced.

11. **What is meant by I/O-bound process?**

Ans : An I/O-bound process spends more of its time doing I/O than it spends doing computations .If all processes are I/O-bound, the ready queue will almost be empty.

12. **What are benefits of Multiprogramming?**

- Responsiveness
- Resource Sharing
- Economy
- Utilization of multiprocessor architectures

13. **Under what circumstances is user level threads is better than the kernel level threads?**

Ans : User threads are supported above the kernel and are managed without kernel support, whereas kernel threads are supported and managed directly by the operating system.

14. **Differentiate single threaded and multi threaded processes.**

Ans : In Multithreaded environment, multiple threads run at the same time, CPU time is never wasted, Idle time is minimum and throughput increases. In Single thread environment, single thread runs at a time, CPU time is wasted, Idle time is more and throughput is less

15. **What is meant by Co-operating process?**

Ans : If a process can affect or be affected by the other processes executing in the system, that process which shares data with other process is called as Co-operating process.

16. **What is meant by Interrupt?**

Ans :An Interrupt is an event that alters the sequence in which a processor executes instructions. It is generated by the hardware of the computer System.

17. **What is meant by Degree of Multiprogramming? And when it is said to be Stable?**

Ans : Degree of Multiprogramming means the number of processes in memory, and it is said to be stable when the average rate of the number of process creation is equal to the average departure rate of processes leaving the system.

18. **What is meant by direct Communication?**

Ans : In Direct communication, each process that wants to communicate must explicitly name the recipient or sender of the communication. In this scheme, the Send & Receive primitives are defined as:

- send (p , message) - Send a message to process P
- receive (p , message) - Receive a message to process p

19. What is meant by Indirect Communication?

Ans : In Indirect Communication, the messages are sent to and received from mailboxes or ports. A mailbox is an object into which messages can be placed by processes and from which messages can be removed. In this scheme, the Send & Receive primitives are defined as:

- send (A , message) - Send a message to mailbox A.
- receive (A , message) - Receive a message from mailbox A.

20. **Compare user threads and kernel threads.**

Ans: **User threads:-**

User threads are supported above the kernel and are implemented by a thread library at the user level. Thread creation & scheduling are done in the user space, without kernel intervention. Therefore they are fast to Creating and manage blocking system call will cause the entire process to block

Kernel threads:-

Kernel threads are supported directly by the operating system .Thread creation, scheduling and management are done by the operating system. Therefore they are slower to Creating & manage compared to user threads. If the thread performs a blocking system call, the kernel can schedule another thread in the application for execution.

21. **Define thread cancellation and target thread.**

Ans: The thread cancellation is the task of terminating a thread before it has completed. A thread that is to be cancelled is often referred to as the target thread. For example, if multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be cancelled.

22. What are the different ways in which a thread can be cancelled?

Ans:Cancellation of a target thread may occur in two different scenarios:

Asynchronous cancellation: One thread immediately terminates the target thread is called asynchronous cancellation.

Deferred cancellation: The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.

23. What are the 3 different types of scheduling queues?
Ans: **Job Queue:** As process enters the system they are put into job queue.
Ready Queue: The processes that are residing in the main memory and are ready and waiting to execute are kept in the queue.
Device Queue: The list of processes waiting for particular I/O device is called a device queue.

24. **Define schedulers?**
Ans: A process migrates between the various scheduling throughout its lifetime. The operating system must select, for scheduling purposes, processes from these queues in some fashion. The selection process is carried out by the appropriate scheduler.

25. **What are the types of scheduler?**
Ans**: Long term scheduler or job scheduler** selects processes from the pool and load them into the memory for execution.
Short term scheduler or CPU scheduler, select among the processes that are ready to execute and allocates the CPU to one of them.
The **medium-term scheduler** is responsible for temporarily removing processes from the main memory (RAM) and moving them to secondary storage (swap space). This is known as **swapping**. It helps manage the level of multiprogramming by reducing the number of processes in the system, allowing better CPU and memory utilization.

26. **Priority inversion is a condition that occurs in real time systems – Analyze on this statement.**
Ans: Priority inversion is a problem that occurs in concurrent processes when low-priority threads hold shared resources required by some high-priority threads, causing the high priority-threads to block indefinitely. This problem is enlarged when the concurrent processes are in a real time system where high- priority threads must be served on time.
Priority inversion occurs when task interdependency exists among tasks with different priorities.

27. **Define CPU Scheduling.**
Ans: CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multi programmed operating systems. By switching the CPU among processes, the operating system can make the computer more productive.

PART – B

1. Write about different types of scheduling queues. Give the queuing diagram representation of process scheduling.
2. Explain about different types of CPU Schedulers in detail.\
3. Explain Context switching.
4. Elaborate the different operations that can be performed on a process in an operating system.
5. PCB in detail.
6. Threading issues in detail

7. IPC in detail.

8. Write about the various CPU scheduling algorithms.

9. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, A non preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling.

b) What is the turnaround time of each process for each of the scheduling algorithms in part a?

c) What is the waiting time of each process for each of the scheduling algorithms in Part a?

d) Which of the schedules in part a results in the minimal average waiting time (over all processes)?

10. What is meant by process? Explain states of process with neat sketch and discuss the process state transition with a neat diagram.

11. Explain the difference in the degree to which FCFS, RR and non-preemptive SJF scheduling algorithms, discriminate in favor of short process

12. Discuss how following pairs of scheduling criteria conflicts certain settings.

a. CPU utilization and response time

b. Average turnaround time and maximum waiting time

c. I/O device utilization and CPU utilization

13.: You are given a list of tasks represented by uppercase English letters (A to Z) and a non-negative integer n that represents the cooldown period between two identical tasks.

Each task takes one unit of time to execute, and the CPU must wait n units of time before executing the same task again. The CPU can remain idle if necessary.

Write a function in C that determines the minimum number of intervals required to complete all tasks.

Input:

tasks = {'A', 'A', 'A', 'B', 'B', 'B'}

n = 2

Output:

8

14. You are given n tasks. Each task has:

Enqueue time (when it becomes available).
Processing time (time required to complete).
The CPU processes one task at a time and follows these rules:

If multiple tasks are available, the CPU picks the one with the shortest processing time.
If two tasks have the same processing time, the CPU picks the one that arrived first.
The CPU cannot stop a running task until it is finished.
Write a C program to print the order of execution of tasks.
Input:
n = 3
enqueue_time = {1, 2, 3}
processing_time = {3, 2, 1}
Output:

2 1 0

15. Write a bash script to calculate the frequency of each word in a text file words.txt. For simplicity's sake, you may assume:

- words.txt contains only lowercase characters and space ' ' characters.
- Each word must consist of lowercase characters only.
- Words are separated by one or more whitespace characters. (Leetcode 192)

16. Given a text file file.txt that contains a list of phone numbers (one per line), write a one-liner bash script to print all valid phone numbers.
You may assume that a valid phone number must appear in one of the following two formats: (xxx) xxx-xxxx or xxx-xxx-xxxx. (x means a digit)
You may also assume each line in the text file must not contain leading or trailing white spaces. (Leetcode 193)

17. Given a text file file.txt, print just the 10th line of the file (handle edge cases also). (Leetcode 195)