OSY-PT-2-SOLUTION-SET-2

---

1. Difference between Preemptive Scheduling and Non-Preemptive Scheduling

Feature	Preemptive Scheduling	Non-Preemptive Scheduling
Definition	CPU can be taken away from a process before completion	CPU is not taken away once assigned until completion
Response Time	Faster; better for real-time applications	Slower; may cause longer waiting times
Examples	Round Robin, Priority Scheduling	FCFS, SJF without preemption

---

2. Describe Linked File Allocation Method

In **Linked File Allocation**, each file is stored as a linked list of disk blocks. Each block contains the address of the next block in the sequence, enabling easy sequential access.

- **Advantages**:
  - Efficient for sequential access.
  - No need for contiguous memory blocks.
- **Disadvantages**:
  - Access speed is slower for random access.
  - Additional storage needed for pointers.

---

3. Define Turnaround Time and Waiting Time

- **Turnaround Time (TAT)**: Total time taken from submission of a process to its completion.
  - Formula: `Turnaround Time = Completion Time - Arrival Time`
- **Waiting Time (WT)**: Total time a process spends waiting in the queue.
  - Formula: `Waiting Time = Turnaround Time - Burst Time`

---

4. Difference between Process and Thread

Feature	Process	Thread
Definition	An independent executing unit	A smaller unit of a process
Memory	Each has its own memory space	Threads share memory within a process
Creation	Slower and requires more resources	Faster and requires fewer resources
Communication	Interprocess Communication (IPC) needed	Direct communication within shared memory
Execution	Executed independently by the OS	Run within the context of a process
Overhead	Higher, as processes have separate memory	Lower, as threads share memory and resources
Crash Impact	If a process crashes, it doesn't affect others	A thread crash can potentially crash the entire process
Scheduling	Processes are independently scheduled by OS	Threads are scheduled together under one process
Context Switching	Slower due to separate memory handling	Faster since threads share memory
Example Use	Running separate applications (e.g., Chrome, Word)	Tasks within a program (e.g., a browser with multiple tabs)
Resource Allocation	Each process has its own resources (e.g., memory, files)	Threads share resources allocated to their parent process

---

5. Difference between Long Term and Short Term Scheduler

Feature	Long Term Scheduler	Short Term Scheduler
Frequency	Less frequent	Highly frequent
Purpose	Decides which processes to add to the pool	Decides which process to execute next
Speed	Slower than short-term scheduler	Faster, works in real-time
Type	Batch-oriented	Interactive-oriented
Control	Controls degree of multiprogramming	Controls CPU utilization
Example	Load balancing	Process queue management

---

6. State and Describe Two Scheduling Criteria

- **CPU Utilization**: Measures how effectively the CPU is utilized; aim for 100% utilization.
- **Throughput**: Number of processes completed per unit time; higher throughput is generally better.

Q2. Attempt Any THREE

1. Describe One-to-One Multithreading Model with Diagram

The **One-to-One Multithreading Model** assigns one kernel thread to each user thread, allowing better performance as each thread can run independently.

- **Advantages**:
  - Improved concurrency, each thread can run in parallel.
  - Better control over threads.

---

2. Types of Scheduler

- **Long-Term Scheduler**: Manages admission of processes to the system.
- **Short-Term Scheduler**: Decides which process to execute next.
- **Medium-Term Scheduler**: Swaps processes in and out to optimize memory.

---

3. Describe Sequential and Direct Access Methods

- **Sequential Access**: Accesses data in a sequential manner, ideal for reading files from start to finish.
- **Direct Access**: Accesses data at any random location, suitable for databases and similar applications.

---

4. Explain Deadlock and Necessary Conditions

Deadlock is a state where processes are unable to proceed because each is waiting for resources held by others.

Necessary Conditions for Deadlock:

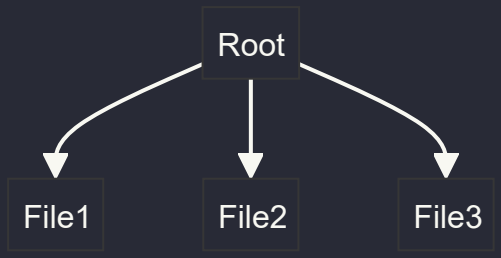
1. **Mutual Exclusion**: Only one process can use a resource at a time.
2. **Hold and Wait**: Processes holding resources can request additional resources.
3. **No Preemption**: Resources cannot be forcibly taken from processes.
4. **Circular Wait**: A circular chain of processes exists, each waiting on the next.

---

5. Directory Structures with Diagrams

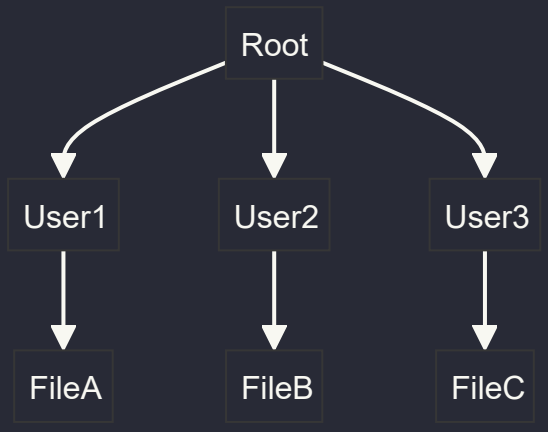
i) Single Level Directory

A **Single Level Directory** is a simple structure where all files are stored in one directory.



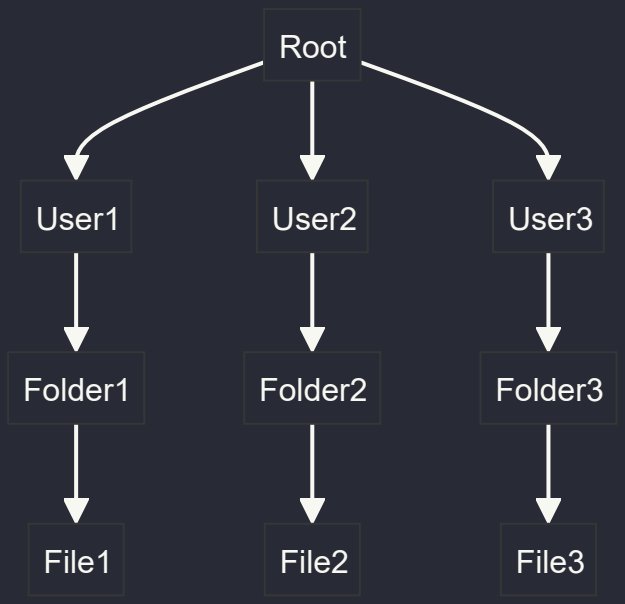
ii) Two-Level Directory

In a **Two-Level Directory**, each user has their own directory.



iii) Tree-Structured Directory

A **Tree-Structured Directory** organizes files hierarchically, allowing nested directories.


Root
User1
User2
User3
Folder1
Folder2
Folder3
File1
File2
File3