CPU Scheduling

CPU Scheduling

- Basic Concepts
- Scheduling Criteria
- Scheduling Algorithms
- Thread Scheduling
- Multiple-Processor Scheduling
- Operating Systems Examples
- Algorithm Evaluation

Objectives

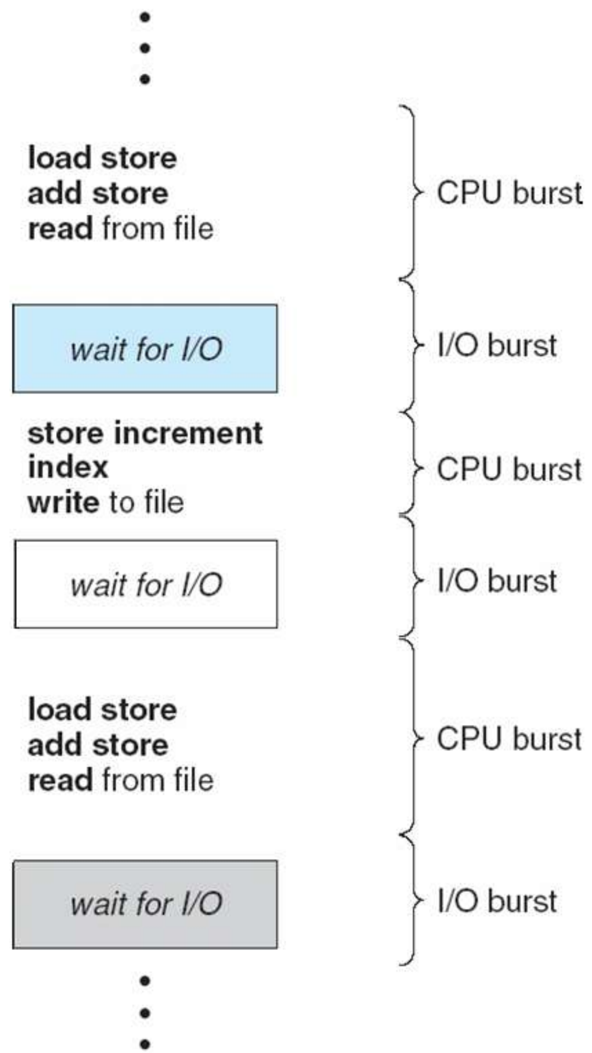
- To introduce CPU scheduling, which is the basis for multiprogrammed operating systems
- To describe various CPU-scheduling algorithms
- To discuss evaluation criteria for selecting a CPU-scheduling algorithm for a particular system

Basic Concepts

- Maximum CPU utilization obtained with multiprogramming
- A CPU bursts when it is executing instructions; an I/O system bursts when it services requests to fetch information.
- **CPU burst** distribution
- Turnaround Time= Turnaround time (TAT) is the time interval from the time of submission of a process to the time of the completion of the process
- **TURNAROUND TIME=WAITING TIME + SERVICE TIME**

- **Burst Time** = The slice that it gets, is called the CPU **burst**. In simple terms, the duration for which a process gets control of the CPU is the CPU **burst time**, and the concept of gaining control of the CPU is the CPU **burst**.
- ***Waiting Time*** *= Starting Time - Arrival Time*

Alternating Sequence of CPU and I/O Bursts



CPU Scheduler

- Selects from among the processes in ready queue, and allocates the CPU to one of them
  - Queue may be ordered in various ways
- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- Scheduling under 1 and 4 is **nonpreemptive**
- All other scheduling is **preemptive**
  - Consider access to shared data
  - Consider preemption while in kernel mode
  - Consider interrupts occurring during crucial OS activities

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – Number of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

Non-Preemptive Scheduling

- once if a process enters into running state, it continues to execute until it terminates or blocks itself to wait for Input/Output or by requesting some operating system service. **First-come, first-served scheduling (FCFS) algorithm**

Preemptive Scheduling :

- currently running process may be interrupted and moved to the ready State by the operating system.
  When a new process arrives or when an interrupt occurs, preemptive policies may incur greater overhead than non-preemptive version but preemptive version may provide better service.
- Round Robin Scheduling

Scheduling Algorithm Optimization Criteria

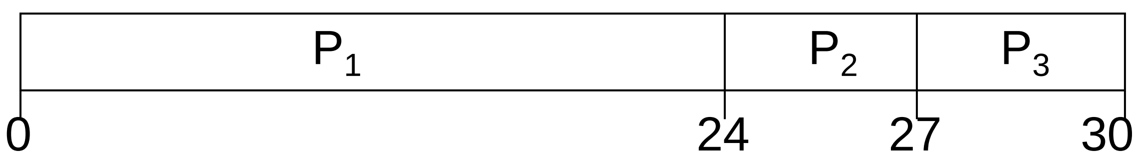
- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

First Come First Serve Scheduling

- In the "First come first serve" scheduling algorithm, as the name suggests, the process which arrives first, gets executed first, or we can say that the process which requests the CPU first, gets the CPU allocated first.
- First Come First Serve, is just like **FIFO**(First in First out) Queue data structure, where the data element which is added to the queue first, is the one who leaves the queue first.
- This is used in Batch Systems.
- It's **easy to understand and implement** programmatically, using a Queue data structure, where a new process enters through the **tail** of the queue, and the scheduler selects process from the **head** of the queue.
- A perfect real life example of FCFS scheduling is **buying tickets at ticket counter**.

First-Come, First-Served (FCFS) Scheduling

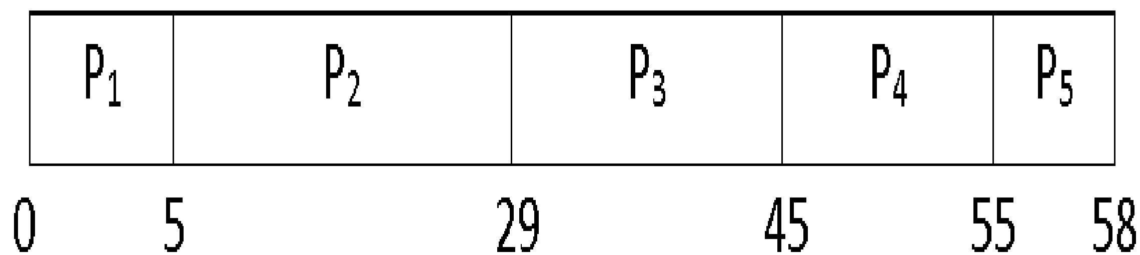
Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3
  The Gantt Chart for the schedule is:
- 

- Waiting time for P_1 = 0; P_2 = 24; P_3 = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

FCFS Algorithm

Process	Burst Time(ms)
P_1	5
P_2	24
P_3	16
P_4	10
P_5	3



- Consider the above set of processes that arrive at time zero. The length of the CPU **burst time** given in millisecond. Now we calculate the average waiting time, average turnaround time.

- **Average Waiting Time and Turnaround Time**
- **Average Waiting Time**
- First of all, we have to calculate the waiting time of each process.
  Waiting Time = Starting Time - Arrival Time
  Waiting time of
  P1 = 0
  P2 = 5 - 0 = 5 ms
  P3 = 29 - 0 = 29 ms
  P4 = 45 - 0 = 45 ms
  P5 = 55 - 0 = 55 ms
  Average Waiting Time = Waiting Time of all Processes / Total Number of Process
  Therefore, average waiting time = (0 + 5 + 29 + 45 + 55) / 5 = 25 ms

- **Average Turnaround Time**
- *Turnaround Time = Waiting time in the ready queue + executing time + waiting time in waiting-queue for I/O*

  Turnaround time of

  P1 = 0 + 5 + 0 = 5ms

  P2 = 5 + 24 + 0 = 29ms

  P3 = 29 + 16 + 0 = 45ms

  P4 = 45 + 10 + 0 = 55ms

  P5 = 55 + 3 + 0 = 58ms

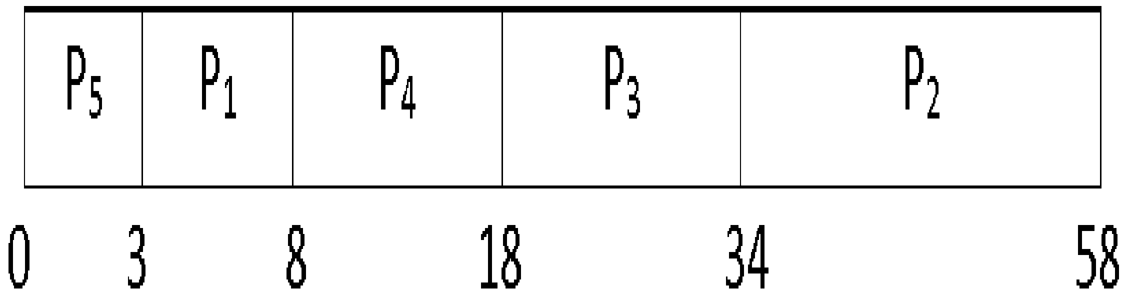
  Total Turnaround Time = (5 + 29 + 45 + 55 + 58)ms = 192ms

  Average Turnaround Time = (Total Turnaround Time / Total Number of Process) = (192 / 5)ms = 38.4ms

- Draw back
- What is Convoy Effect?
- Convoy Effect is a situation where many processes, who need to use a resource for short time are blocked by one process holding that resource for a long time.
- This essentially leads to poort utilization of resources and hence poor performance.

SJF (Shortest Job First) Scheduling

Process	Burst Time(ms)
P_1	5
P_2	24
P_3	16
P_4	10
P_5	3



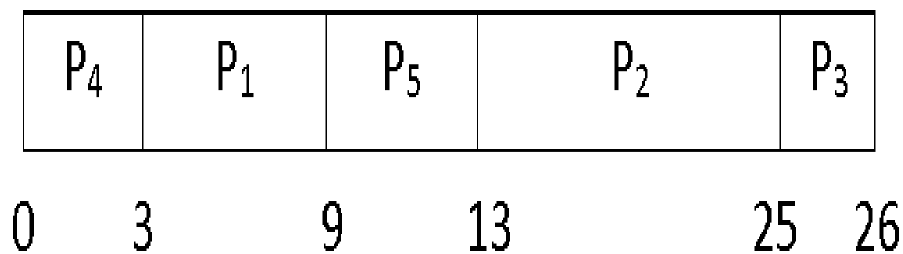
- **Average Waiting Time**
- We will apply the same formula to find average waiting time in this problem. Here arrival time is common to all processes(i.e., zero).
  Waiting Time for
  P1 = 3 - 0 = 3ms
  P2 = 34 - 0 = 34ms
  P3 = 18 - 0 = 18ms
  P4 = 8 - 0 = 8ms
  P5 = 0ms
  Now, Average Waiting Time = (3 + 34 + 18 + 8 + 0) / 5 = 12.6ms

- **Average Turnaround Time**
- According to the SJF Gantt chart and the turnaround time formulae,
  Turnaround Time of
  P1 = 3 + 5 = 8ms
  P2 = 34 + 24 = 58ms
  P3 = 18 + 16 = 34ms
  P4 = 8 + 10 = 18ms
  P5 = 0 + 3 = 3ms
  Therefore, Average Turnaround Time = (8 + 58 + 34 + 18 + 3) / 5 = 24.2ms

Priority Scheduling Example

Process	CPU Burst Time	Priority
P_1	6	2
P_2	12	4
P_3	1	5
P_4	3	1
P_5	4	3

Gantt Chart



- Processes with same priority are executed in FCFS manner.
- **Average Waiting Time**
- First of all, we have to find out the waiting time of each process.
  Waiting Time of process
  P1 = 3ms
  P2 = 13ms
  P3 = 25ms
  P4 = 0ms
  P5 = 9ms
  Therefore, Average Waiting Time = (3 + 13 + 25 + 0 + 9) / 5 = 10ms

- **Average Turnaround Time**
- First finding Turnaround Time of each process.
  Turnaround Time of process
  P1 = (3 + 6) = 9ms
  P2 = (13 + 12) = 25ms
  P3 = (25 + 1) = 26ms
  P4 = (0 + 3) = 3ms
  P5 = (9 + 4) = 13ms
  Therefore, Average Turnaround Time = (9 + 25 + 26 + 3 + 13) / 5 = 15.2ms

Round Robin Scheduling Example

Process	CPU Burst Time
P_1	30
P_2	6
P_3	8

Gantt Chart

P_1	P_2	P_3	P_1	P_2	P_3	P_1	P_1	P_1	P_1

0 5 10 15 20 21 24 29 34 39 44

- Time Quantum is **5ms**.
- **Average Waiting Time**
- For finding Average Waiting Time, we have to find out the waiting time of each process.
  Waiting Time of
  P1 = 0 + (15 - 5) + (24 - 20) = 14ms
  P2 = 5 + (20 - 10) = 15ms
  P3 = 10 + (21 - 15) = 16ms
  Therefore, Average Waiting Time = (14 + 15 + 16) / 3 = 15ms

- **Average Turnaround Time**
- Same concept for finding the Turnaround Time.
  Turnaround Time of
  P1 = 14 + 30 = 44ms
  P2 = 15 + 6 = 21ms
  P3 = 16 + 8 = 24ms
  Therefore, Average Turnaround Time = (44 + 21 + 24) / 3 = 29.66ms