OPERATING SYSTEMS

Module2_Part4

Textbook : Operating Systems Concepts by Silberschatz

Scheduling algorithms

CPU scheduling deals with the problem of deciding which of the processes in the ready queue

is to be allocated the CPU. There are many different CPU-scheduling algorithms. Some of

them are

First-Come, First-Served Scheduling

Shortest-Job-First Scheduling

shortest-remaining-time-first

Priority Scheduling

Round-Robin Scheduling

First-Come, First-Served Scheduling

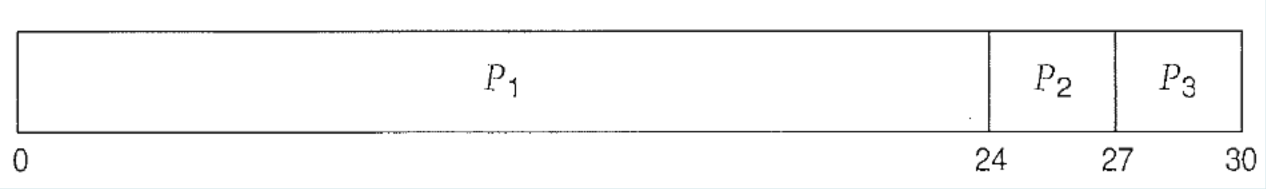
- the simplest CPU-scheduling algorithm
- the process that requests the CPU first is allocated the CPU first.
- The implementation of the FCFS policy is easily managed with a FIFO queue.
- When a process enters the ready queue, its PCB is linked onto the tail of the queue.
- When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue.
- The code for FCFS scheduling is simple to write and understand.

- Consider the following set of processes that arrive at time 0, with the length of the CPU

  burst given in milliseconds:
- Example with 3 processes

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 above schedule is:(Gantt chart, which is a bar chart that illustrates a particular schedule, including the start and finish times of each of the participating processes)



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

First-Come, First-Served Scheduling

If the processes arrive in the order P2, *P3* , P1, however, the results will be as shown in the following Gantt chart:

The average waiting time is now (6 + 0 + 3)/3 = 3 milliseconds. This reduction is substantial. Thus, the average waiting time under an FCFS policy is generally not minimal and may vary substantially if the processes CPU burst times vary greatly.

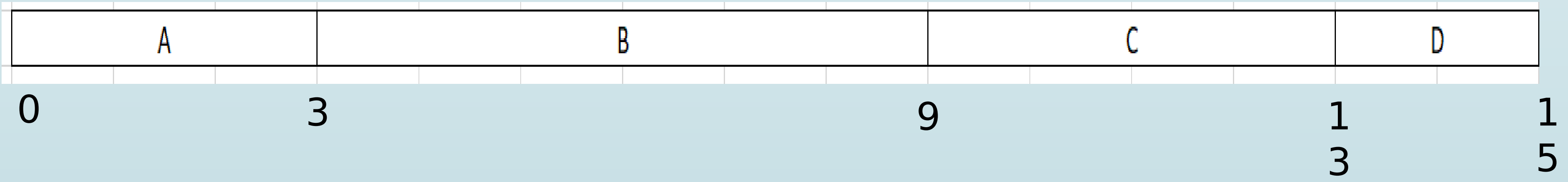
First-Come, First-Served Scheduling

- the FCFS scheduling algorithm is nonpreemptive.
- Once the CPU has been allocated to a process, that process keeps the CPU until it releases the CPU, either by terminating or by requesting I/O.
- The FCFS algorithm is thus particularly troublesome for time-sharing systems, where it is important that each user get a share of the CPU at regular intervals. It would be disastrous to allow one process to keep the CPU for an extended period.
- The processes with higher burst time arrived before processes with smaller burst time, then smaller processes have to wait for a long time for longer processes to release cpu (Convoy effect)

First-Come, First-Served Scheduling Example

For the processes listed draw gantt chart illustrating their execution

process	Arrival time	Processing time
A	0	3
B	1	6
C	4	4
D	6	2



First-Come, First-Served Scheduling Example

- For the process listed what is the average turn around time?

Process	Arrival time	Processing time	Completion time	Turn around time
A	0	3	3	3
B	1	6	9	8
C	4	4	13	9
D	6	2	15	9

Turn around time=completion time –arrival time

Average turn around time=((3-0)+(9-1)+(13-4)+(15-6))/4 = 7.25

First-Come, First-Served Scheduling Example

- For the processes listed what is the waiting time for each process?

Process	Arrival time	Processing time	Completion time	Turn around time	Waiting time
A	0	3	3	3	0
B	1	6	9	8	2
C	4	4	13	9	5
D	6	2	15	9	7

Waiting time=turn around time – execution time

A:(3-3)=0   B:(8-6)=2   C:(9-4)=5   D:(9-2)=7