100 Years 1920-2020

University of Lucknow | Centenary Year

लखनऊ विश्वविद्यालय | शताब्दी वर्ष

Operating System: Process Scheduling

Dr. Puneet Misra

Department of Computer Science,

University of Lucknow,

Lucknow PIN-226007

puneetmisra@gmail.com

+919415159146

Agenda

- Process
- Process States
- Criteria for scheduling
- Process Scheduling
- Pre-emptive vs Non-pre-emptive scheduling
- Scheduling Algorithms with examples
  - ✓First Come First Serve (FCFS)
  - ✓Shortest Job First (SJF)
  - ✓Priority Scheduling (PS)
  - ✓Shortest Remaining Time Next (SRTN)
  - ✓Round Robin Scheduling
- Comparison

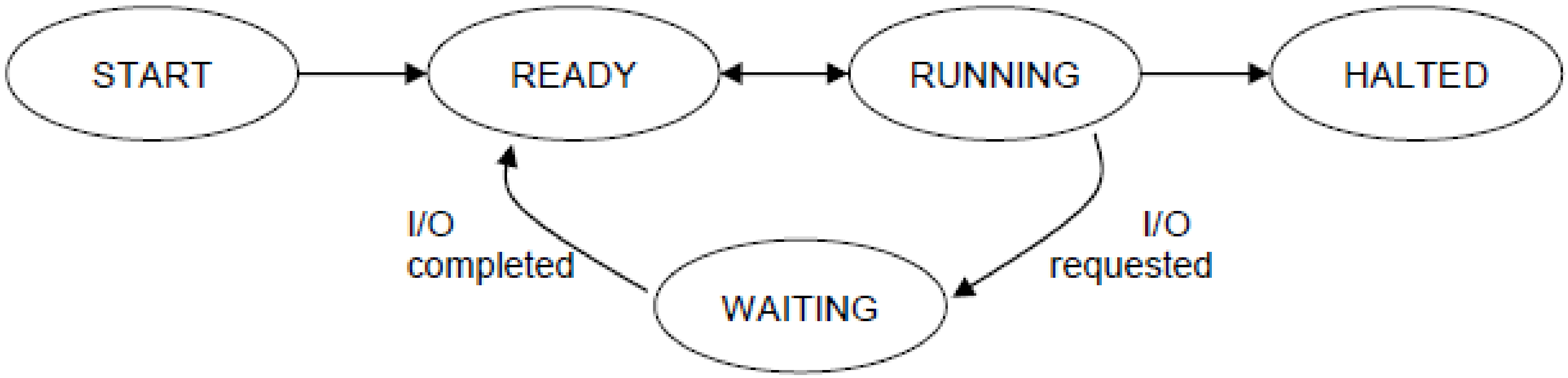
Process

- The term "process" was first used by the designers of the MULTICS in 1960's. Since then, the term process, used somewhat interchangeably with 'task' or 'job'.
- The process has been given many definitions for instance
  - A program in Execution.
  - An asynchronous activity.
  - The 'animated sprit' of a procedure in execution.
  - The entity to which processors are assigned.
  - The 'dispatchable' unit.
- Process is not the same as program. A process is more than a program code. A process is an 'active' entity as oppose to program which is considered to be a 'passive' entity.

Process

- A process is an executing program, which includes the current values of the program counter, registers, and variables.
- The subtle difference between a process and a program is that the program is a group of instructions whereas the process is the activity. In multiprogramming systems, processes are performed in a pseudoparallelism as if each process has its own processor. In fact, there is only one processor but it switches back and forth from process to process.
- Henceforth, execution of a process, means the processor's operations on the process like changing its variables, etc. and I/O work means the interaction of the process with the I/O operations like reading something or writing to somewhere. They may also be named as "processor (CPU) burst" and "I/O burst" respectively. According to these definitions, we classify programs as
  - Processor-bound program: A program having long processor bursts (execution instants)
  - I/O-bound program: A program having short processor bursts.

Process States



- **Start :** The process has just arrived.
- **Ready :** The process is waiting to grab the processor.
- **Running :** The process has been allocated by the processor.
- **Waiting :** The process is waiting for some I/O control or waiting for some event.
- **Halted :** The process has finished and is about to leave the system.

Process Control Block

- A process, in an operating system, is represented by a data structure known as a process control block (PCB) or process descriptor. The PCB contains important information about the specific process including
  - ✓ The current state of the process i.e., whether it is ready, running, waiting, or whatever.
  - ✓ Unique identification of the process in order to track "which is which" information.
  - ✓ A pointer to parent process.
  - ✓ Similarly, a pointer to child process (if it exists).
  - ✓ The priority of process (a part of CPU scheduling information).
  - ✓ Pointers to locate memory of processes.
  - ✓ A register save area.
  - ✓ The processor it is running on.

Criteria for Scheduling

- CPU utilization: keep the CPU as busy as possible
  - ✓Maximize
- Throughput: No of processes that complete their execution per time unit
  - ✓Maximize
- Turnaround time: amount of time to execute a particular process (time from submission to termination)
  - ✓Minimize
- Waiting time: amount of time a process has been waiting in the ready queue (sum of time waiting in ready queue)
  - ✓Minimize
- 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)
  - ✓Minimize
- Fairness: surety that each process gets a fair share of the CPU

Scheduling Algorithms:

Non-preemptive Scheduling

- These are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time.

Preemptive Scheduling

- This is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

Scheduling Algorithms

- First Come First Serve (FCFS)
- Shortest Job First (SJF)
- Priority Scheduling (PS)
- Shortest Remaining Time Next (SRTN)
- Round Robin Scheduling

First Come First Serve Scheduling

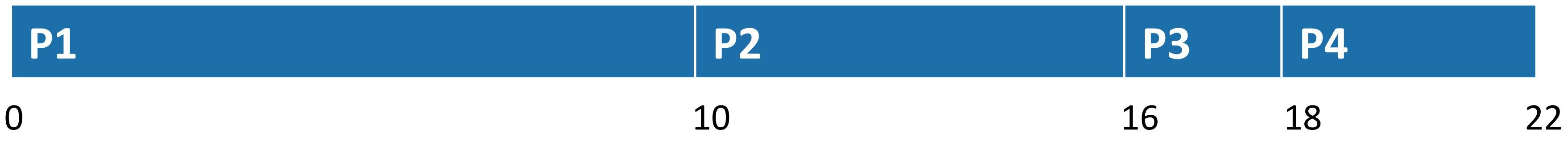
- The processes are scheduled for execution in the order of their arrival i.e. the process which enters the queue first is executed first.
- It is a run to completion policy i.e. each process runs to completion. Therefore, It is non-preemptive (Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be pre-empted until it completes its allotted time)
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- The process leaves CPU in following conditions:
  - The Process has completed its job.
  - The Process has asked for an I/O Operation
  - The process has encountered an exception causing abnormal exit.

FCFS (With no priority):Example 1:

Example 1:

Process	Arrival time (in ms)	Burst time (in ms)
P1	0	10
P2	1	6
P3	3	2
P4	5	4

- Gantt Chart:



Example 1:

Process	Arrival time (in ms) T0	Burst time (in ms) BT	Finish time (in ms) Ti	Turn around time (TAT) (in ms) Ti-T0	Waiting time (in ms) TAT-BT
P1	0	10	10	10	0
P2	1	6	16	15	9
P3	3	2	18	15	13
P4	5	4	22	17	19

- Average Turnaround time = (10+15+15+17)/4 = 57/4 = 14.25ms
- Average waiting time = (0+9+13+19)/4 = 41/4 =10.25ms

FCFS (With no priority):Example 2:

Example2:

Process	Arrival time (in ms)	Burst time (in ms)
P1	0	24
P2	0	3
P3	0	3

- Gantt Chart:

Example 2:

Process	Arrival time (in ms)	Burst time (in ms)	Completion time (in ms)	Waiting time (in ms)	Turn around time (TAT) (in ms)
P1	0	24	24	0	24
P2	0	3	27	24	27
P3	0	3	30	27	30

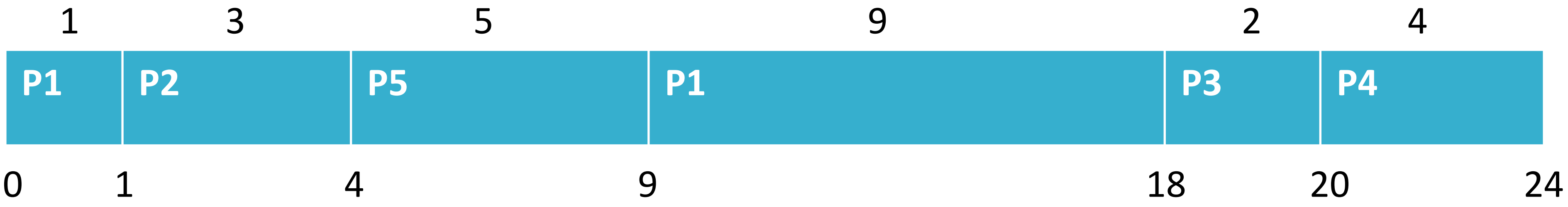
- Average waiting time = (0+24+27)/3 = 51/3 =17ms
- Average Turnaround time = (24+27+30)/3 = 81/3 = 27ms

FCFS (With priority) :

Example:

Process	Arrival time (in ms)	Burst time (in ms)	Priority (1>2>3>4>5)
P1	0	10	3
P2	1	3	1
P3	2	2	3
P4	3	4	4
P5	4	5	2

- Gantt Chart:



Process	Arrival time (in ms)	Burst time (in ms)	Priority	Process wait (in ms)	Completion time (in ms)	Waiting time (in ms)	Turn around time (TAT) (in ms)
P1	0	10	3	9	18	8	18
P2	1	3	1	1	4	0	3
P3	2	2	3	18	20	16	18
P4	3	4	4	20	24	17	21
P5	4	5	2	4	9	0	5

- Average waiting time = (8+0+16+17+0)/5 = 41/5 =8.2ms
- Average Turnaround time = (18+3+18+21+5)/5 = 65/5 = 13.0ms

First Come First Serve

Benefits

- Simple and easy to implement
- Starvation free
- Examples: printer queues, mail queues

Drawbacks

- Response time
- Suffers from Convoy Effect

Shortest Job First Scheduling (SJF)

- The process with the lowest execution time is executed first.
- After completing the job next burst of remaining is considered.
- The next job to be dispatched will be the shortest among pending jobs.
- It can be either preemptive or non preemptive. Thus there are two types of SJFS:-
  - Non pre-emptive SJF
  - Preemptive SJF

Shortest Job Scheduling First (Non-pre-emptive)

Example 1:

Process	Arrival time (in ms)	Burst time (in ms)
P1	0	10
P2	1	6
P3	3	2
P4	5	4

- Gantt Chart:

P1	P3	P4	P2

0 10 12 16 22

Example 1:

Process	Arrival time (in ms) T0	Burst time (in ms) BT	Finish time (in ms) Ti	Turn around time (TAT) (in ms) Ti-T0	Waiting time (in ms) TAT-BT
P1	0	10	10	10	0
P2	1	6	22	21	15
P3	3	2	12	9	7
P4	5	4	16	11	7

- Average Turnaround time = (10+21+9+11)/4 = 51/4 = 12.75ms
- Average waiting time = (0+15+7+7)/4 = 41/4 =7.25ms

Shortest Job Scheduling First {Non-Preemptive)

Example 2:

Process	Arrival time (in ms)	Burst time (in ms)
P1	0	24
P2	0	3
P3	0	3

- Gantt Chart:

Process	Arrival time (in ms)	Burst time (in ms)	Completion time (in ms)	Waiting time (in ms)	Turn around time (TAT) (in ms) WT+BT
P1	0	24	30	6	30
P2	0	3	3	0	3
P3	0	3	6	3	6

- Average waiting time = (6+0+3)/3 = 9/3 =3ms
- Average Turnaround time = (30+3+6)/3 = 39/3 = 13ms

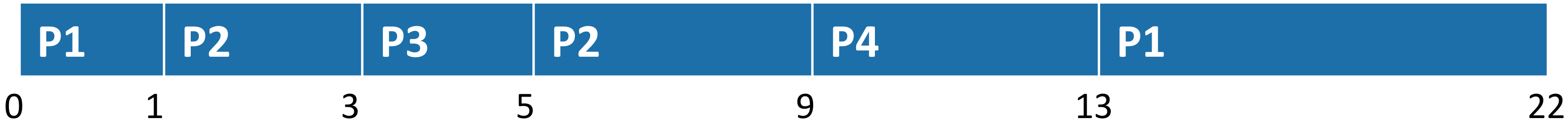
Shortest Remaining Time Next Scheduling OR Job Scheduling First (Pre-emptive)

- Next job dispatched will be the one that happens to be shortest among pending jobs. If any process arrives later whose next burst happens to be shorter than remaining burst time of currently running process will be pre-emptive.

Example 1:

Process	Arrival time (in ms)	Burst time (in ms)
P1	0	10
P2	1	6
P3	3	2
P4	5	4

- Gantt Chart:



Example 1:

Process	Arrival time (in ms) T0	Burst time (in ms) BT	Finish time (in ms) Ti	Turn around time (TAT) (in ms) Ti-T0	Waiting time (in ms) TAT-BT
P1	0	10	22	22	12
P2	1	6	9	8	2
P3	3	2	5	2	0
P4	5	4	13	8	4

- Average Turnaround time = (22+8+2+8)/4 = 40/4 = 10 ms
- Average waiting time = (12+2+0+4)/4 = 18/4 =4.5 ms

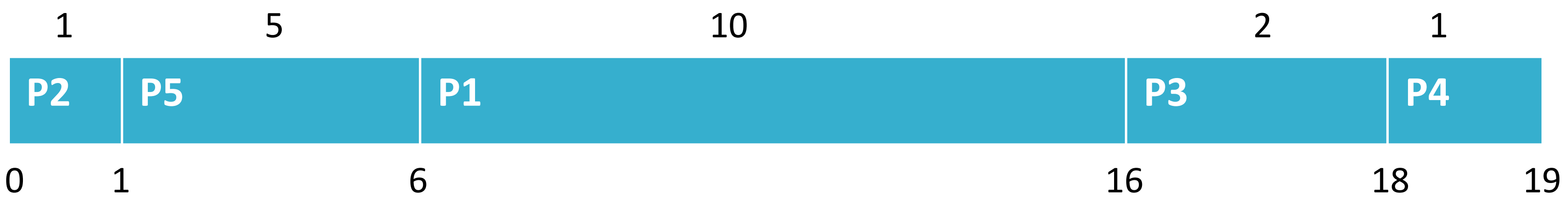
Priority Scheduling:

- In this scheduling more critical processes are assigned priorities other than the less critical one. Each process is associated with priority and the CPU is allocated to each of them based on their priorities.
- Multiple process with same priority are dealt using FCFS.

Example (Non-pre-emptive priority scheduling):

Process	Arrival time (in ms)	Burst time (in ms)	Priority (1>2>3>4>5)
P1	0	10	3
P2	0	1	1
P3	0	2	4
P4	0	1	5
P5	0	5	2

Gantt Chart



Process	Arrival time (in ms)	Burst time (in ms)	Priority	Process wait (in ms)	Completion time (in ms)	Waiting time (in ms)	Turn around time (TAT) (in ms)
P1	0	10	3	6	16	6	16
P2	0	1	1	0	1	0	1
P3	0	2	4	16	18	16	18
P4	0	1	5	18	19	18	19
P5	0	5	2	1	6	1	6

- Average waiting time = (6+0+16+18+1)/5 = 41/5 =8.2ms
- Average Turnaround time = (16+1+18+19+6)/5 = 60/5 = 12.0ms

Priority based non-pre-emptive scheduling:

- At the time of scheduling, a process dispatched will be the one that has the highest priority.
- Once dispatched process is allowed to complete even if higher priority process becomes ready to run during the currently running process
- Multiple process with same priority are dealt using FCFS.

Example (Non-pre-emptive priority scheduling):

Process	Arrival time (in ms)	Burst time (in ms)	Priority (0>1>2>3>4>5)
P1	0	10	5
P2	1	6	4
P3	3	2	2
P4	5	4	0

Gantt Chart

Process	Arrival time (in ms)	Burst time (in ms)	Priority	Process wait (in ms)	Completion time (in ms)	Turn around time (TAT) (in ms)	Waiting time (in ms)
P1	0	10	5	0	10	10	0
P2	1	6	4	16	22	21	15
P3	3	2	2	14	16	13	11
P4	5	4	0	10	14	9	5

- Average Turnaround time = (10+21+13+9)/4 = 53/4 = 13.25ms
- Average waiting time = (0+15+11+5)/4 = 31/5 =7.75ms

Priority based pre-emptive scheduling:

- At the time of scheduling, a process dispatched will be the one that has the highest priority among processes waiting in the ready queue.
- When a process is executing and if another process of higher priority becomes ready to run during the execution of earlier process then this earlier process will be pre-empted by the process in the ready queue.

Example (Pre-emptive priority scheduling):

Process	Arrival time (in ms)	Burst time (in ms)	Priority (0>1>2>3>4>5)
P1	0	10	5
P2	1	6	4
P3	3	2	2
P4	5	4	0

Gantt Chart

P1	P2	P3	P4	P2	P1	
0	1	3	5	9	13	22

Process	Arrival time (in ms)	Burst time (in ms)	Priority	Process wait (in ms)	Finish time (in ms)	Turn around time (TAT) (in ms) Ti-T0	Waiting time (in ms)
P1	0	10	5	13	22	22	12
P2	1	6	4	1	13	12	6
P3	3	2	2	3	5	2	0
P4	5	4	0	5	9	4	0

- Average Turnaround time = (22+12+2+4)/4 = 40/4 = 10ms
- Average waiting time = (12+6+0+0)/4 = 18/4 =4.5ms

Round Robin Scheduling

- For providing quick response time to every user in a multiuser environment Round Robin scheduling is used which is a preemptive scheduling algorithm.
- Each process is provided a fix time to execute, it is called a quantum.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.

EXAMPLE:-

Time quantum is 4ms.

Process	Arrival Time	Burst Time
P1	0	24
P2	0	3
P3	0	3

Gantt Chart

4	3	3	4	4	4	4	4
P1	P2	P3	P1	P1	P1	P1	P1

0 4 7 10 14 18 22 26 30

Process	Arrival time (in ms)	Burst time (in ms)	Priority	Process wait (in ms)	Completion time (in ms)	Waiting time (in ms)	Turn around time (TAT) (in ms)
P1	0	24	0	10	30	10	30
P2	0	3	0	4	7	4	7
P3	0	3	0	7	10	7	10

- Average waiting time = (10+4+7)/3 = 21/3 = 3 ms
- Average Turnaround time = (30+7+10)/3 = 47/3 = 15.67ms

Comparison

- Comparing the average turnaround time and waiting time, the algorithms are ranked as SJF, FCFS and PS.
- Based on average CPU utilization the algorithms are ranked as FCFS, PS and SJF.
- In terms of execution speed, FCFS was found to be consistently faster than PS and SJF.