Scheduling

Syllabus

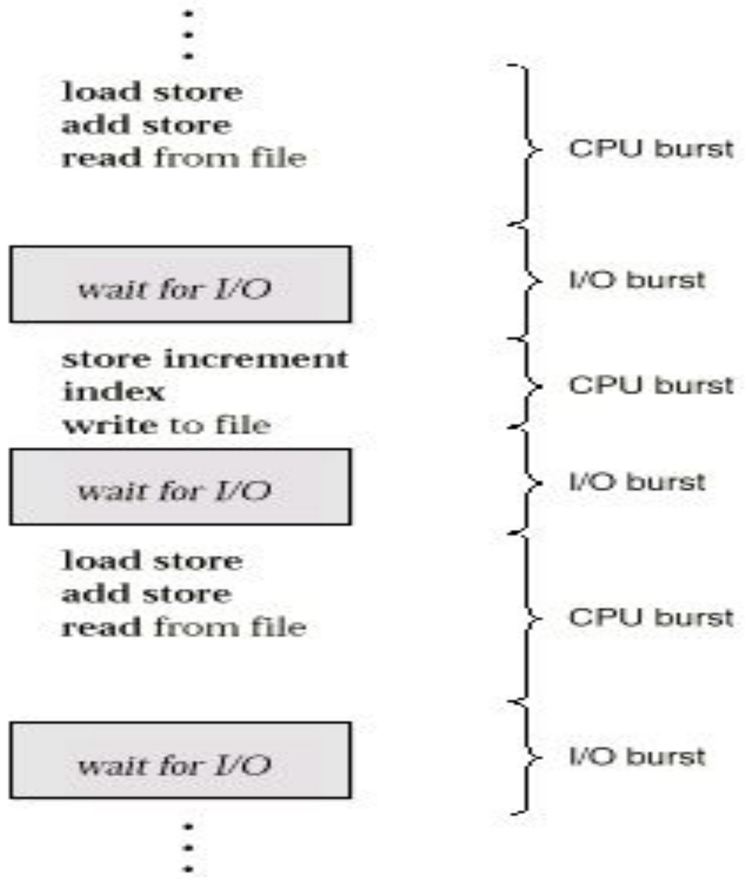
Scheduling: Types of Scheduling, Scheduling Algorithms: FCFS, SJF, Priority, Round Robin.

Scheduling

- Maximum CPU utilization obtained with multiprogramming

- CPU–I/O Burst Cycle – Process execution consists of a *cycle* of CPU execution and I/O wait.

Alternating Sequence of CPU And I/O Bursts



CPU / Process Scheduler

- Selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.

- Scheduling may be *preemptive or nonpreemptive*

Nonpreemptive – If CPU allocated to a process, it keeps the CPU until it releases it by terminating or switching to waiting state

Preemptive - If CPU allocated to a process, it may be released if high priority process needs the CPU

Scheduler

- 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 state
  4.Terminates
- Scheduling under 1 and 4 is *nonpreemptive*
- All other scheduling is *preemptive*

Scheduling Algorithms

They deal with the problem of deciding which of the processes in ready queue is to be allocated the CPU

Algo. compared based on following criteria

CPU utilization – keep the CPU as busy as possible

Throughput – # of processes completed or amount of work done per unit time

Turnaround time – time of submission of a process to the time of completion

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

Optimization Criteria

Max CPU utilization

Max throughput

Min turnaround time

Min waiting time

Min response time

Types of process schedulers / Scheduling categories

- Scheduling is broken down into three categories:

1. Long term scheduling:

- Is performed when a new process is created.
- Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue.
- Long-term scheduler is invoked very infrequently (seconds, minutes) $\Rightarrow$ (may be slow)
- The long-term scheduler controls the *degree of multiprogramming*.

Types of process schedulers / Scheduling categories

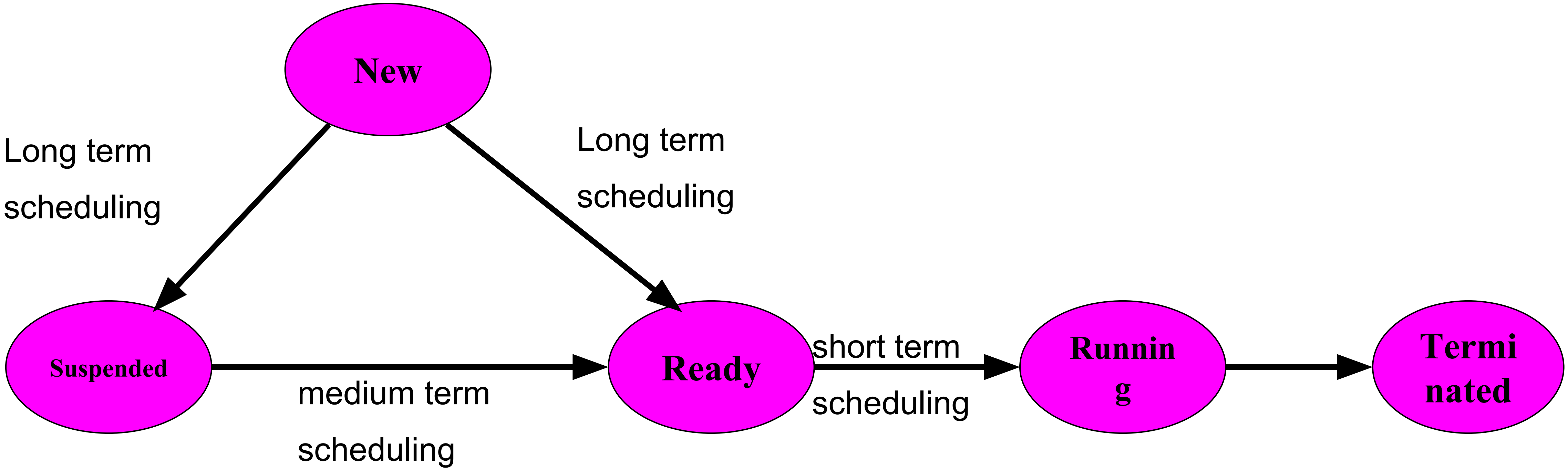
2. Medium term scheduling:

- Part of the swapping function
- Swapping-in decisions are taken by medium term scheduler
- Based on the need to manage the degree of multiprogramming

3.Short term scheduling:

- Determines which ready process will be assigned the CPU when it next becomes available and actually assign the CPU to this process .
- Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU.
- Short-term scheduler is invoked very frequently (milliseconds)$\Rightarrow$ (must be fast).

Types of process schedulers / Scheduling categories



Scheduling Algorithms

- FCFS (First Come First Serve)
- SJF (Shortest Job First)
- Priority scheduling
- Round Robin scheduling

FCFS Scheduling: Characteristics

Selection Function: max(w), selects the process which is waiting in the ready queue for maximum time**.**

Decision Mode : Non_preemptive

Throughput: Not emphasized

Response Time: May be high, especially if there is a large variance in the process execution times.

Overhead: Minimum

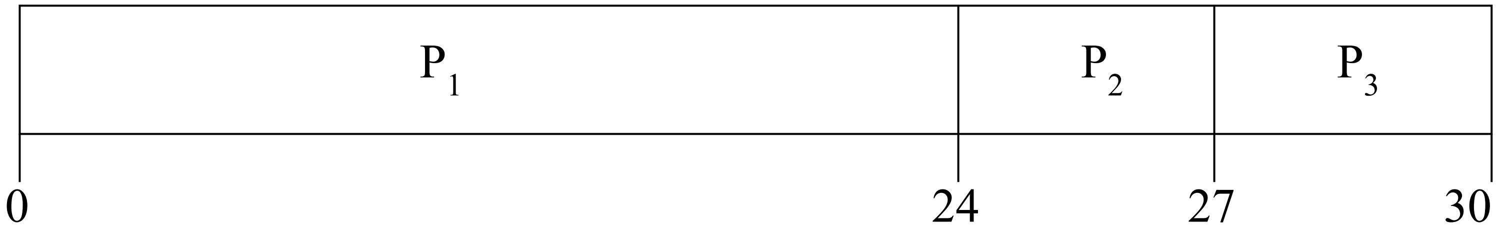
Effect on Processes: Penalizes short processes

Starvation: No

Process	**Burst Time**
P1	24
P2	3
P3	3

- Suppose that the processes arrive in the order: *P1* , *P2* , *P3*

The Gantt Chart for the schedule is:

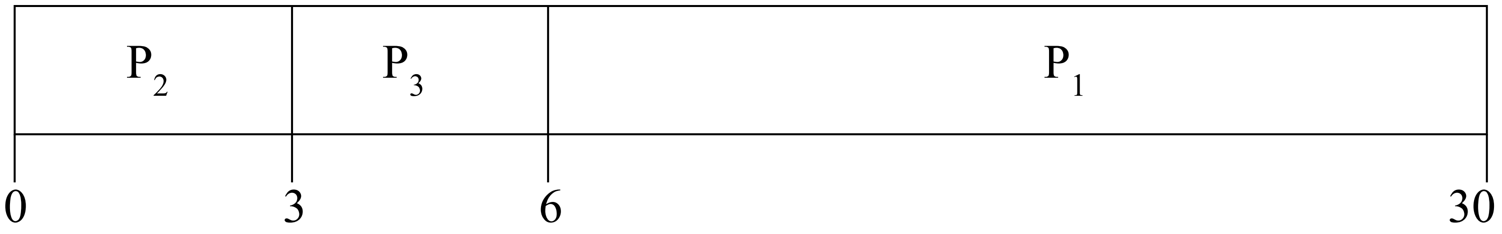


Waiting time for *P1* = 0; *P2* = 24; *P3* = 27

Average waiting time: (0 + 24 + 27)/3 = 17

FCFS Scheduling (Cont.)

- Suppose that the processes arrive in the order : *P2* , *P3* , *P1*



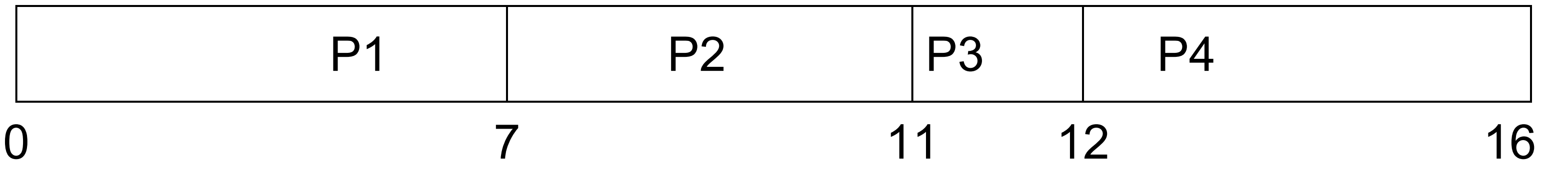
- Waiting time for $P1 = 6$; $P2 = 0$; $P3 = 3$
- Average waiting time: $(6 + 0 + 3)/3 = 3$

Example-2 of FCFS

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4

Example-2 of FCFS

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



Average waiting time = (0 + 5 + 7 + 7)/4 = 4.75
Average Turnaround Time = (7+9+8+11)/4=8.75

Shortest-Job-First (SJR) Scheduling

- Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time

- Two schemes:
  - **nonpreemptive** – once CPU given to the process it cannot be preempted until completes its CPU burst
  - **preemptive** – if a new process arrives with CPU burst length less than remaining time of current executing process, preempt. This scheme is know as the **Shortest-Remaining-Time-First (SRTF)**

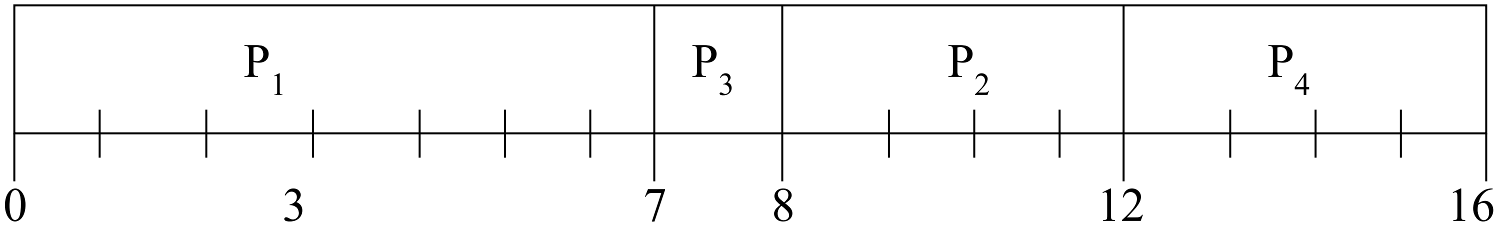
- **SJF is optimal** – gives minimum average waiting time for a given set of processes

Example of Non-Preemptive SJF

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4

Example of Non-Preemptive SJF

Process	Arrival Time	Burst Time
P1	0.0	7
P2	2.0	4
P3	4.0	1
P4	5.0	4



- Average waiting time = (0 + 6 + 3 + 7)/4 = 4

Shortest Job First Preemptive or Shortest Remaining Time

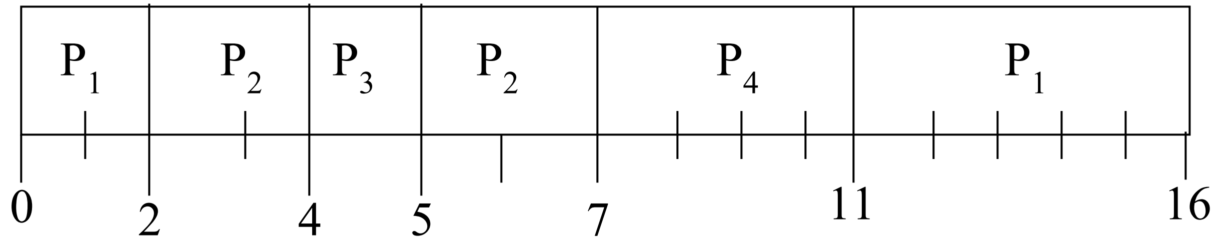
- It is a preemptive version of SJF. In this policy, scheduler always chooses the process that has the shortest expected remaining processing time.
- When a new process arrives in the ready queue, it may in fact have a shorter remaining time than the currently running process.
- Accordingly, the scheduler may preempt whenever a new process becomes ready.
- Scheduler must have an estimate of processing time to perform the selection function.

Shortest Job First Preemptive or Shortest Remaining Time: characteristics

- **Selection Function**: minimum total service time required by the process, minus time spent in execution so far.
- **Decision Mode : Preemptive (At arrival time)**
- **Throughput: High**
- **Response Time:** Provides good response time
- **Overhead: Can be high**
- **Effect on Processes:** Penalizes long processes.
- **Starvation:** Possible

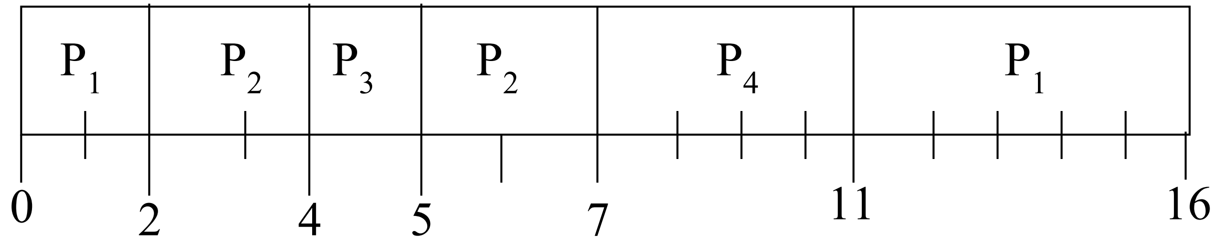
Example of Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4



Example of Preemptive SJF

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4



- Average waiting time = (9 + 1 + 0 +2)/4 = 3
- Average turnaround time= (16+5+1+6)/4=7

Priority Scheduling

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer ≡ highest priority)
  - Preemptive
  - Nonpreemptive
- SJF is a priority scheduling where priority is the predicted next CPU burst time
- Problem ≡ Starvation – low priority processes may never execute
- Solution ≡ Aging – as time progresses increase the priority of the process

Round Robin (RR)

- Each process gets a small unit of CPU time (*time quantum*)
- After this time has elapsed, the process is preempted and added to the end of the ready queue.
- If there are n processes in the ready queue and the time quantum is q, then each process gets $1/n$ of the CPU time in chunks of at most q time units at once.
- No process waits more than $(n-1)q$ time units.

RR: characteristics

- **Selection Function**: constant
- **Decision Mode : Preemptive (At time quantum)**
- **Throughput: May be low if time quantum is too small**
- **Response Time:** Provides good response time for short processes
- **Overhead: Minimum**
- **Effect on Processes:** Fair treatment
- **Starvation:** No

Example of RR with Time Quantum = 20

Process Burst Time

P1 53

P2 17

P3 68

P4 24

P_1	P_2	P_3	P_4	P_1	P_3	P_4	P_1	P_3	P_3

0 20 37 57 77 97 117 121 134 154 162

- Typically, higher average turnaround than SJF, but better *response*

Example: Round Robin (By Default preemptive: Time Quantum 2)

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

Example: Round Robin (By Default preemptive: Time Quantum 2)

Process	**Arrival Time**	**Burst Time**
P_1	**0.0**	**7**
P_2	**2.0**	**4**
P_3	**4.0**	**1**
P_4	**5.0**	**4**

P1	P2	P1	P3	P2	P4	P1	P4	P1

0 2 4 6 7 9 11 13 15 16

Average waiting time = (9 + 3 + 2 +6)/4 =5
Average turnaround time =(16+7+3+10)/4=9

References

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, ISBN-10: 0-13-380591-3, ISBN-13: 978-0-13-380591-8, 8th Edition
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, WILEY,ISBN 978-1-118-06333-0, 9th Edition