LECTURE 12: SCHEDULING ALGO-1

Chapter 3 of Operating System
Concepts- Abraham Silberschatz,
Peter Baer Galvin, Greg Gagne

Chapter 7 of OSTEP

Session Outline

❖ **CPU scheduling**

❖ **Categories of scheduling algorithms**

❖ **FCFS scheduling algorithm**

❖ **SJF scheduling algorithm**

❖ **SRTF scheduling algorithm**

❖ **Round Robin scheduling algorithm**

❖ **Priority scheduling algorithm**

Preemptive vs Non-preemptive Scheduling
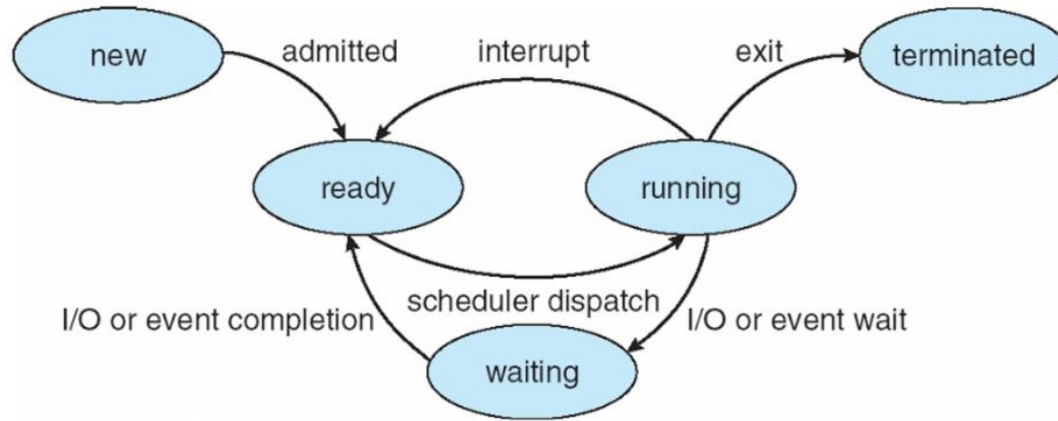
1. When a process switches from the running state to the waiting state

2. When a process switches from the running state to the ready state

3. When a process switches from the waiting state to the ready state

4. When a process terminates

❖ 1 & 4, the scheduling is **non-preemptive** (cooperative)

❖ 2 & 3, the scheduling is **preemptive**

Scheduling Criteria

❖ Different CPU-scheduling algorithms have different properties.

❖ Certain characteristics/criteria are used for comparing various CPU scheduling algorithms.

  ❖ CPU Utilization

  ❖ Throughput

  ❖ Turnaround time

  ❖ Waiting Time

  ❖ Response Time

Categories of Scheduling Algorithms & Goals

❖ **Batch System**

  ❖ Complete maximum number of jobs per unit time.

  ❖ Minimize time between submission and termination

  ❖ Keep CPU busy all time

❖ **Interactive System**

  ❖ Response to requests quickly

  ❖ Reduce waiting time

  ❖ Meet user expectations

❖ **Real time System**

  ❖ Meeting deadlines

  ❖ Ensure quality constraints

CPU Scheduling Algorithms

Batch Systems

- ❖ First-come first-served
- ❖ Shortest job first
- ❖ Shortest remaining Time next

Interactive Systems

- ❖ Round-robin scheduling
- ❖ Priority scheduling
- ❖ Multiple queues
- ❖ Shortest process next
- ❖ Guaranteed scheduling
- ❖ Lottery scheduling
- ❖ Fair-share scheduling

FCFS Scheduling

❖ Simplest CPU-scheduling algorithm

❖ First-come, first-served - process that requests the CPU first is allocated the CPU first

❖ FCFS policy is managed with a FIFO queue

❖ When a process enters the ready queue, its PCB is linked onto the tail of the FIFO queue

❖ When the CPU is free, it is allocated to process at the head of the queue

❖ The running process is then removed from the queue

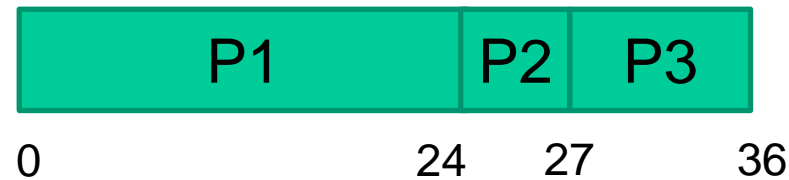❖ It is non-preemptive, once scheduled it will complete

❖ Short jobs wait for long

FCFS Scheduling

❖ Example: Three processes arrive in order P1, P2, P3 all at time 0.

    ❖ P1 burst time: 24

    ❖ P2 burst time: 3

    ❖ P3 burst time: 9

P1	P2	P3

0   24   27   36

❖ Waiting Time

    ❖ P1: 0, P2: 24, P3: 27

❖ Completion Time:

    ❖ P1: 24, 2: 27, P3: 36

❖ Average Waiting Time: (0+24+27)/3 = 17

❖ Average Completion Time: (24+27+36)/3 = 29

SJF Scheduling

❖ Shortest (in terms of CPU time) job available is scheduled first

❖ Shorter processes makes progress

❖ SJF policy is managed with a priority queue with burst time as input

❖ When a process enters the ready queue, its PCB is linked onto the priority queue at the appropriate entry

❖ When the CPU is free, it is allocated to the process at the head of the priority queue

❖ If too many short jobs, long processes will starve

❖ SJF is non-preemptive; once allotted the process will complete

❖ Lowest turnaround time

SJF Scheduling

❖ Consider 3 process P2, P3, P1 all arriving at time T0.

   ❖ P1 burst time: 24

   ❖ P2 burst time: 3

   ❖ P3 burst time: 9

P2	P3	P1	

0   3   12   36

❖ Waiting Time

   ❖ P1: 12,   P2: 0,   P3: 3

❖ Completion Time:

   ❖ P1: 36,   P2: 3,   P3: 12

❖ Average Waiting Time: (12+0+3)/3 = 5 (compared to 17)

❖ Average Completion Time: (36+3+12)/3 = 17 (compared to 29)

SRTF Scheduling

- ❖ Shortest Remaining Time First (SRTF) job is scheduled first

- ❖ Preemptive scheduling algorithm

- ❖ A priority queue with remaining time is used as input

- ❖ When a process enters/re-enters the ready queue, its PCB is linked onto the priority queue at the appropriate entry

- ❖ When the CPU is free, it is allocated to the process at the head of the priority queue

- ❖ Newly arriving short process may forcefully preempt currently running process.

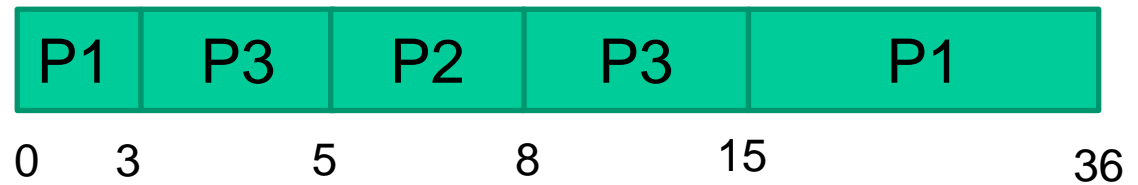- ❖ Longer process may have multiple context switch before completion

SRTF Scheduling

❖ **Consider the following process arriving at different time slots**

  ❖ P1 burst time: 24 arrives at 0

  ❖ P2 burst time: 3 arrives at 5

P1	P3	P2	P3	P1

0   3   5   8   15   36

  ❖ P3 burst time: 9 arrives at 3

❖ Waiting Time

  ❖ P1: (15-3) =12,   P2: 0,   P3: (8-5) = 3

❖ Completion Time:

  ❖ P1: 36,   P2: 8,   P3: 15

❖ Average Waiting Time: (0+3+12)/3 = 5

❖ Average Completion Time: (8+15+36)/3 = 19.6

Round Robin Scheduling

❖ Modified version of preemptive FCFS

❖ Each process gets a small unit of CPU time (time quantum)

❖ FIFO queue is used as input

❖ When a process enters/re-enters the ready queue, its PCB is linked the tail of the queue

❖ After quantum expires, the process is preempted and added to the tail of the ready queue (Hence, preemptive scheduling algorithm)

❖ CPU is allocated to the process at the head of the queue

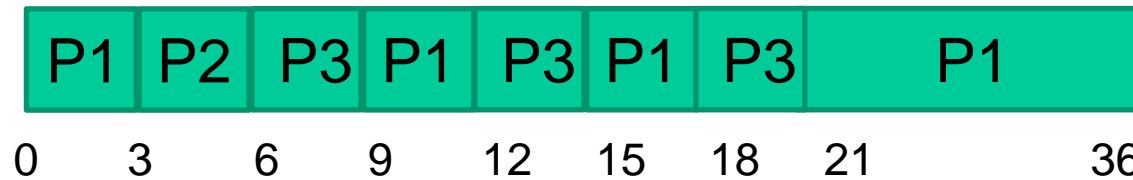❖ Longer process may have multiple context switch before completion

Round Robin Scheduling

❖ **Consider the following process arriving at T0, time quantum of 3 units**

    ❖ P1 burst time: 24

    ❖ P2 burst time: 3

    ❖ P3 burst time: 9

P1	P2	P3	P1	P3	P1	P3	P1

0   3   6   9   12   15   18   21   36

❖ Waiting Time

    ❖ P1: (6+3+3) =12,   P2: 3,   P3: (6+3+3)= 12

❖ Completion Time:

    ❖ P1: 36,   P2: 3,   P3: 21

❖ Average Waiting Time: (3+12+12)/3 =

❖ Average Completion Time: (3+21+36)/3 = 20

Round Robin Scheduling

❖ RR scheduling is better for short jobs and fair

❖ Shorter response time, good for interactive jobs

❖ Context-switching time adds up for long jobs

❖ Context switching takes additional time and overhead

❖ If the chosen quantum is

  ❖ too large, response time suffers

  ❖ infinite, performance is the same as FIFO

  ❖ too small, throughput suffers and percentage overhead grows

Priority Scheduling

❖ Each process has a priority number

❖ Highest priority process is scheduled first; if equal priorities, then FCFS

❖ Managed with a priority queue with priority value as input

❖ When a process enters the ready queue, its PCB is linked onto the priority queue at the appropriate entry

❖ CPU is allocated to the process at the head of the queue

❖ It can have 2 variants; non-preemptive and preemptive

❖ Arrival of a new process with a higher priority can preempt the currently running process.

Thank You