## Module 19: Multi-core computing Operating Systems

# Lecture 37: Multi-core computing CPU Scheduling

### The Lecture Contains:

Previous Next

## Multi-core computing
## Operating Systems

### Contents

- Scheduling
    - Traditional
    - Real time
    - Multi-processor
- Synchronization and synchronization objects
- Inter-process communication
- Security

## Multi-core computing
## CPU Scheduling

### Scheduling

- To hide the effects of I/O Bursts and achieve higher CPU utilization
- To give a fair chance to all processes
- Short term scheduling
    - CPU scheduling
- Long term scheduling
    - Process admission policies
- Medium term scheduling
    - Swap management

Previous Next

## CPU Scheduling

- CPU scheduler selects a process from “Ready to run” queue
  - Scheduling decision
- Allocates CPU to this process.
  - Dispatch
    - Context switching
- Queue is maintained by PCB pointers
- Pre-emptive scheduling
  - When a running process may be removed and put back in the “ready to run” queue.

## When to Have CPU Scheduling

- Process moves from running state to waiting state
  - Due to an I/O request
  - Due to a call to wait
- Upon Process Termination
- Expiration of time quota (preemptive)
- Upon any other time when OS is called
  - Interrupts, System Calls

Previous Next

## Module 19: Multi-core computing Operating Systems

## Lecture 37: Multi-core computing CPU Scheduling

### Scheduling Criteria

- CPU Utilization (System centric)
    - Keep CPU as much busy as possible
- Throughput (System centric)
    - Number of processes completed/time
- Turnaround time (Process centric)
    - Real time taken to complete a process
- Waiting time (Process centric)
    - How much time a process is in ready queue
- Response time (Process centric)
    - Factor for an interactive process
- Deadline (Real time behavior)
    - Time guarantee to schedule a task.

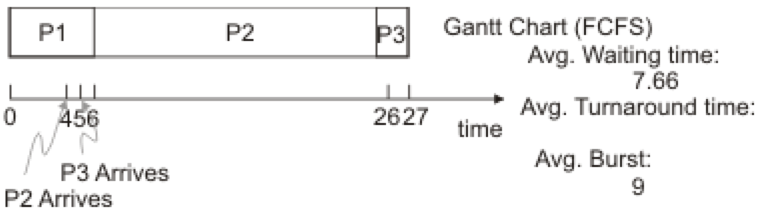
### Scheduling Algorithms

- First Come First Serve (FCFS)
    - Simplest implementation
    - No preemption
- Shortest Job First (SJF)
    - Optimal scheduling algorithm
    - Minimum Average Waiting Time
    - Difficulty: To know the time that it will take
    - Batch systems: A good choice.

Previous Next

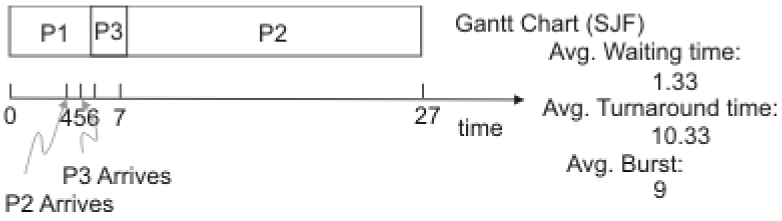
## Example

| Process | Arrival Time | CPU Burst | Wait Time | Turnaround |
|---|---|---|---|---|
| P1 | 0 | 6 | 0 | 6 |
| P2 | 4 | 20 | 2 | 22 |
| P3 | 5 | 1 | 21 | 22 |



## Example (Contd.)

| Process | Arrival Time | CPU Burst | Wait Time | Turnaround |
|---|---|---|---|---|
| P1 | 0 | 6 | 0 | 6 |
| P2 | 4 | 20 | 3 | 23 |
| P3 | 5 | 1 | 1 | 2 |



Previous Next

## SJF

- Though optimal, not practical.
    - No way to find the CPU burst.
- Scheduling techniques try to approximate SJF
    - Guess/predict the next burst ( $\tau_n$ : Predicted $n^{th}$ burst. $t_n$ : Actual $n^{th}$ burst)

$$t_{n+1} = \alpha t_n + (1-\alpha)\tau_n \qquad 0 \le \alpha \le 1$$

$$\tau_{n+1} = \alpha t_n + (1-\alpha)\,\tau_n \qquad 0 \le \alpha \le 1$$

$$\tau_{n+1} = t_n \qquad \text{when } \alpha = 1$$

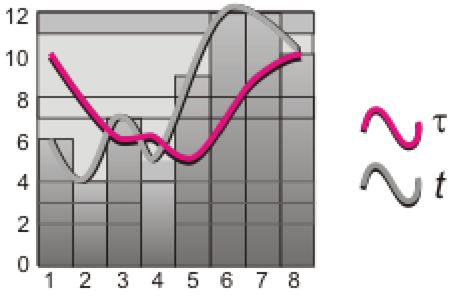
$$\tau_{n+1} = \tau_n \qquad \text{when } \alpha = 0$$

$$\tau_{n+1} = \alpha t_n + (1-\alpha).\alpha.t_{n-1} + \cdots + (1-\alpha)^j.\alpha.t_{n-j} + \cdots + (1-\alpha)^{n+1}\tau_0$$

- Burst guess is obtained by exponentialaveraging.
- $\tau_0$ is a system wide scheduling constant.
- $\alpha$ is a weight factor.

## Example of “Guessing”

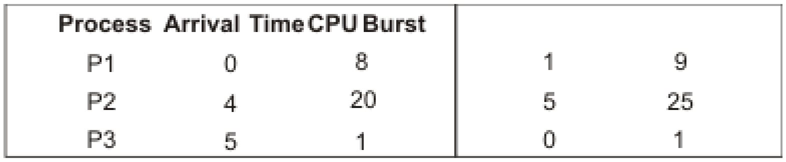
| CPU Burst | 6 | 4 | 7 | 5 | 9 | 12 | 12 | 10 |
|---|---|---|---|---|---|---|---|---|
| Guess ( $\tau_0$=10, $\alpha$=0.5 | 10 | 8 | 6 | 6 | 9 | 7 | 9 | 10 |



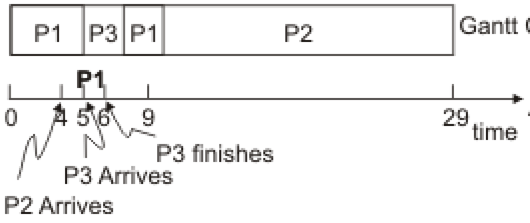
## Preemptive SJF

- When a new task arrives, the SJF is evaluated again and rescheduling can take place.
- *aka* Shortest-remaining-time-first (SRTF) scheduling

## Example (SRTF)



| Process | Arrival Time | CPU Burst | | |
|---|---|---|---|---|
| P1 | 0 | 8 | 1 | 9 |
| P2 | 4 | 20 | 5 | 25 |
| P3 | 5 | 1 | 0 | 1 |

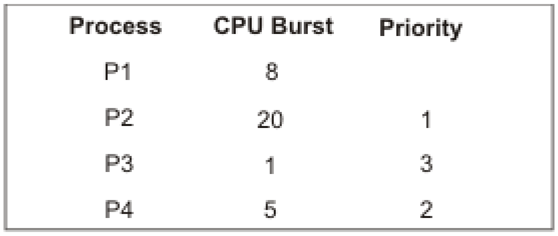


Gantt Chart (SRTF)
Avg. Waiting time:
2
Avg. Turnaround time:
11.66
Avg. Burst:
9.66

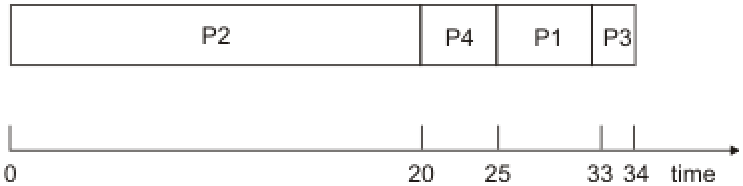
## Adding Priority (Priority Scheduling)

- Simplest form is to add priority to a process.
  - Select the highest priority job first
  - Apply SJF/SRTF/FIFO/... scheduling only within a group of processes with the same priority

## Priority scheduling + SJF



| Process | CPU Burst | Priority |
|---|---|---|
| P1 | 8 | |
| P2 | 20 | 1 |
| P3 | 1 | 3 |
| P4 | 5 | 2 |



Previous Next