OPERATING SYSTEMS

Process Management 8

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 1

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC - Shared Memory & Message Passing, Pipes - Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

OPERATING SYSTEMS

Course Outline

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

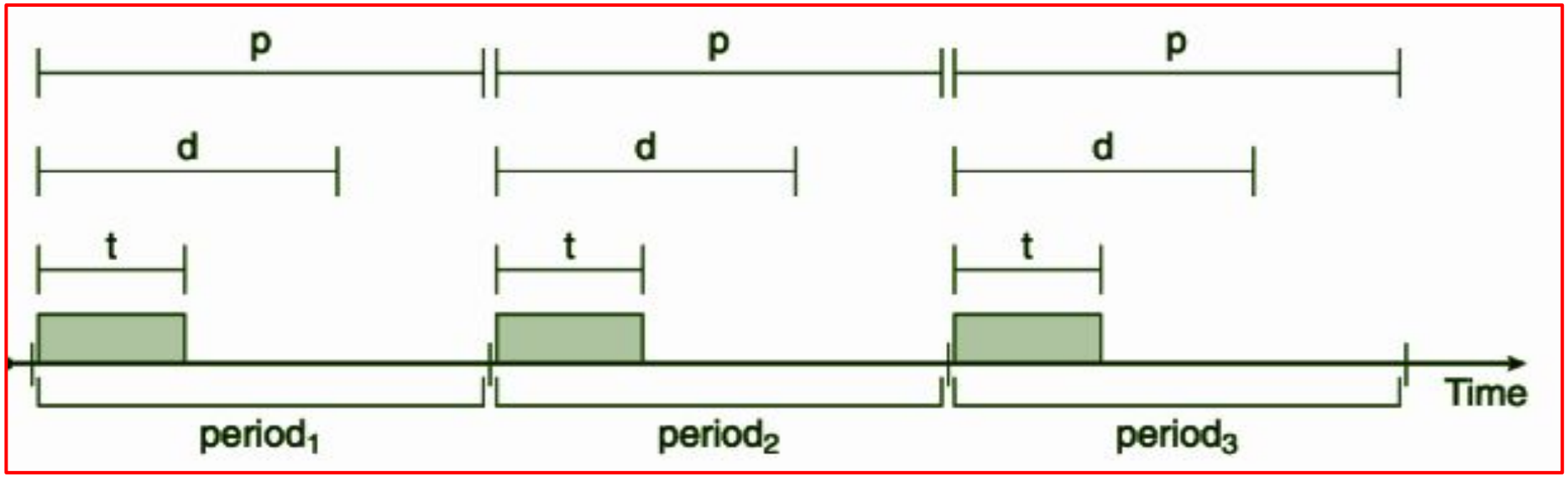
Topics Outline

- Real Time Scheduling
  - Real Time Based Priority Scheduling
  - Rate Monotonic Scheduling
  - Earliest Deadline First Scheduling
  - Proportional Share Scheduling
- Virtualization and Scheduling

Real Time Priority Based Scheduling

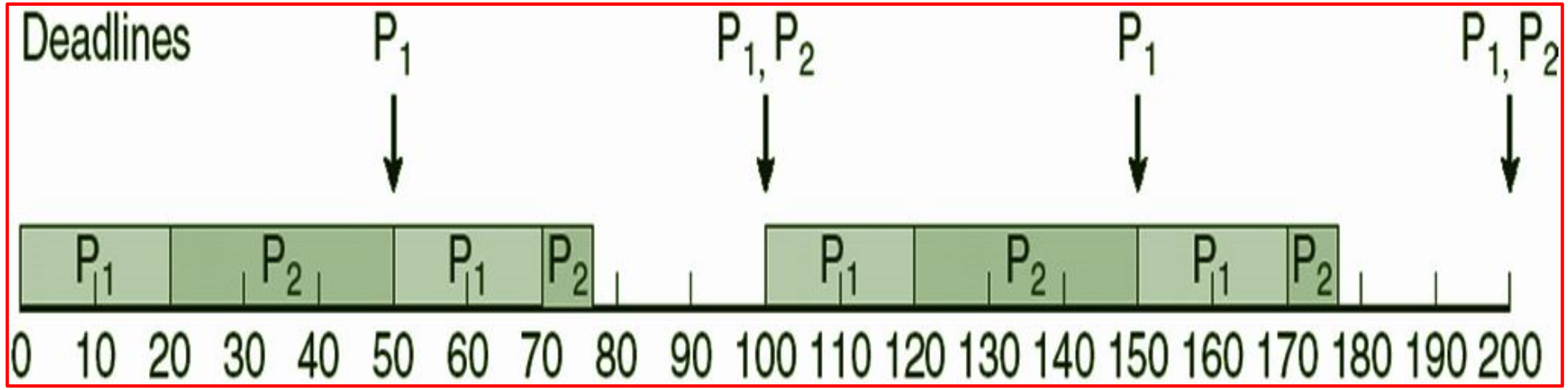
- Many real-time processes are periodic, i.e., they require CPU at constant intervals

- Has processing time t, deadline d, period p $0 \leq t \leq d \leq p$; Rate of periodic task is 1/p



Rate Monotonic Scheduling

- A priority is assigned based on the inverse of its period
  - Shorter periods = higher priority
  - Longer periods = lower priority
  - P1=>20 is assigned a higher priority than P2=>30



Rate Monotonic Scheduling

$$\sum_{k=1}^{n} \frac{C_k}{T_k} \leq U_{RM} = n(2^{1/n} - 1)$$

where n is the number of tasks in a task set.

Example of RATE MONOTONIC (RM) SCHEDULING ALGORITHM

For example, we have a task set that consists of three tasks as follows

Tasks	Release time(ri)	Execution time(Ci)	Deadline (Di)	Time period(Ti)
T1	0	0.5	3	3
T2	0	1	4	4
T3	0	2	6	6

Task set

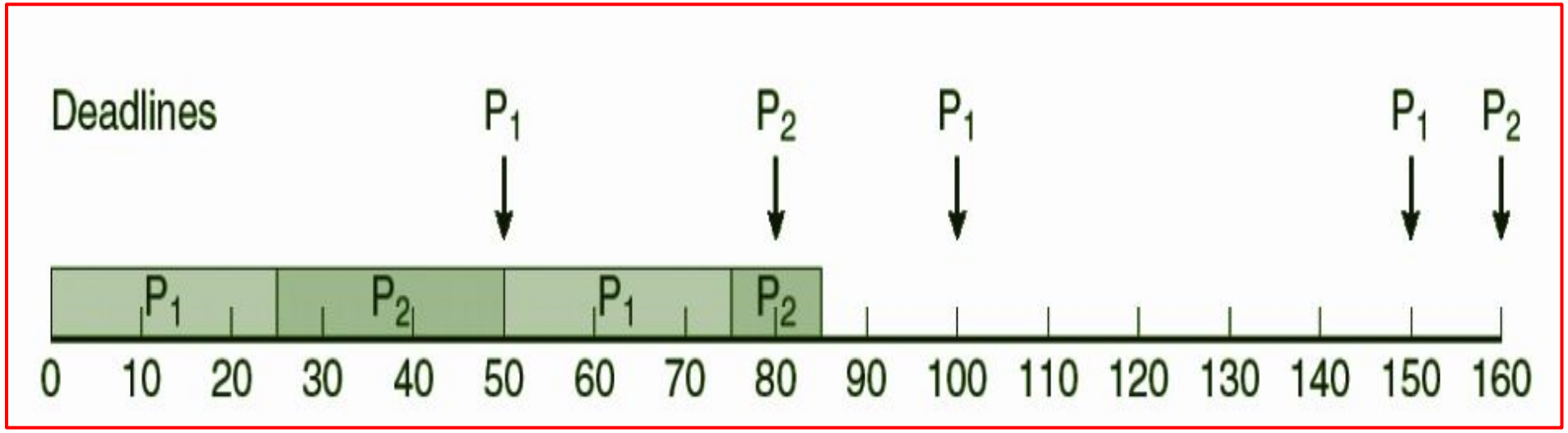
U= 0.5/3 +1/4 +2/6 = 0.167+ 0.25 + 0.333 = 0.75

As processor utilization is less than 1 or 100% so task set is schedulable and it also satisfies the above equation of rate monotonic scheduling algorithm.

RM scheduling of Task set in

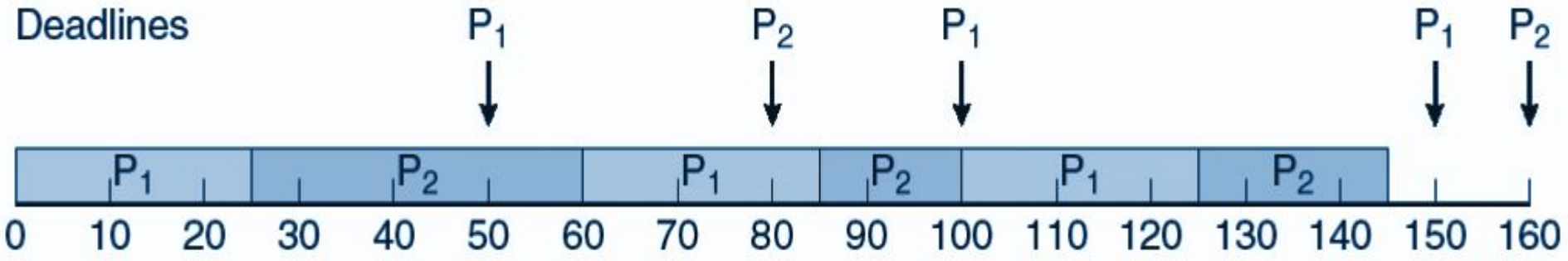
Missed deadlines with Rate Monotonic Scheduling

- A priority is assigned based on the inverse of its period
  - Shorter periods = higher priority
  - Longer periods = lower priority
  - P1=>20 is assigned a higher priority than P2=>30



Earliest deadline First (EDF) Scheduling

- Priorities are assigned according to deadlines
  - the earlier the deadline, the higher the priority;
  - the later the deadline, the lower the priority



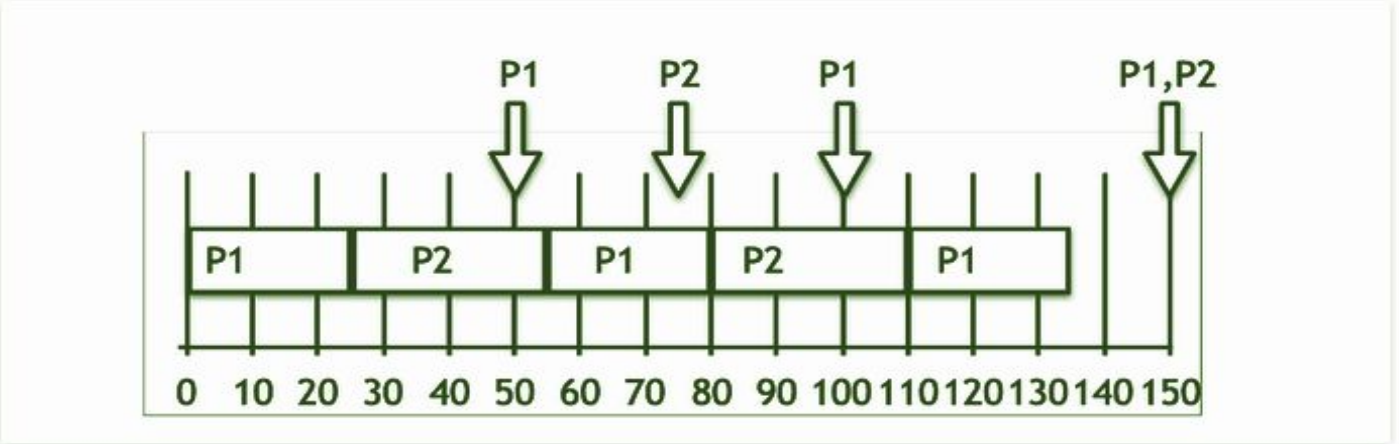
Earliest deadline First (EDF) Scheduling

Let the period of P1 be $p_1 = 50$

Let the processing time of P1 be $t_1 = 25$

Let the period of P2 be $period_2 = 75$

Let the processing time of P2 be $t_2 = 30$



Steps for solution:

1. Deadline pf P1 is earlier, so priority of P1>P2.
2. Initially P1 runs and completes its execution of 25 time.
3. After 25 times, P2 starts to execute until 50 times, when P1 is able to execute.
4. Now, comparing the deadline of (P1, P2) = (100, 75), P2 continues to execute.
5. P2 completes its processing at time 55.
6. P1 starts to execute until time 75, when P2 is able to execute.
7. Now, again comparing the deadline of (P1, P2) = (100, 150), P1 continues to execute.
8. Repeat the above steps...
9. Finally at time 150, both P1 and P2 have the same deadline, so P2 will continue to execute till its processing time after which P1 starts to execute.

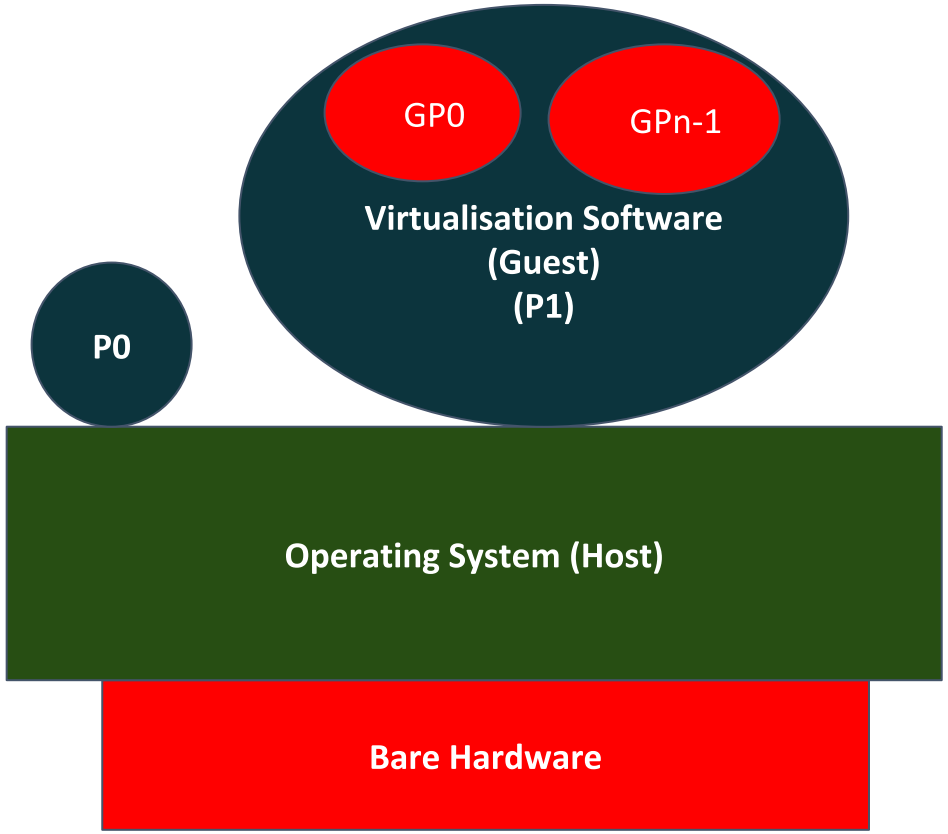
Proportional Share Scheduling

- *T* shares are allocated among all processes in the system
- An application receives *N* shares where $N < T$
- This ensures each application will receive N / T of the total processor time

Virtualization and Scheduling

- Virtualization software schedules multiple guests onto CPU(s)
- Each guest doing its own scheduling
  - Not knowing it doesn't own the CPUs
  - Can result in poor response time
  - Can effect time-of-day clocks in guests
- Can undo good scheduling algorithm efforts of guests

Virtualization and Scheduling



Topics Uncovered in this Session

- Real Time Scheduling
  - Real Time Based Priority Scheduling
  - Rate Monotonic Scheduling
  - Earliest Deadline First Scheduling
  - Proportional Share Scheduling
- Virtualization and Scheduling

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com