Multi - Processor Scheduling :

→ We need to handle some more complexities along with earlier ones

1) Processor affinity : Imbalance of load in multi - processors. Can migrate some processes to increase efficiency.

1) clean cache of prev. processor

2) load into cache of new processor

process gets affinity to processor which, they try to avoid. ie, is keep trying to get same process so that migration is not needed.

2) Load Balancing : A component to maintain load in all processors.

Migration

Push

Push load to another CPU

Pull

Pull load to another CPU

Real Time Scheduling :

Only aim is to meet the deadlines.

Properties of process : (Besides Burst time)

1) Periodicity :

2) Deadline :

→ For every process, we'll have atleast 2 parameters.

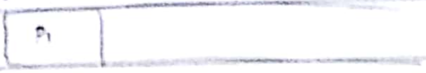
→ We don't consider arrival time here

Before 50th unit, this (1st instance) should be completed

Before arrival of next instance (100th unit) it's 1st instance should be finished.

	P_1	P_2
Periodicity	50	100
Burst time	20	35

Both P_1 & P_2 arrive at time $t=0$



0	50	100	
P_1	P_1'	P_1''	. . .
P_2	P_2	P_2'	

This way, it works in periodic manner.

→ If ~~the~~ ~~prot~~ algo isn't able to meet the deadline . . it' ~~won't~~ will be considered as a failure.

CPU utilization for $P_1 = \frac{20}{50} = 0.4 = 40\%$

$P_2 = \frac{35}{100} = 0.35 = 35\%$

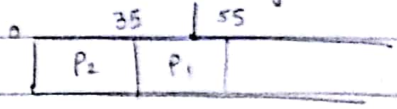
Admission Control : Scheduler decides whether to accept a process or send it to ready Queue based on a process's CPU utilization

Total : 75% utilization ⇒ We may able to meet the deadlines

1) Static Priority Algorithm

Each Process has priority

→ If in earlier eg. , $P_r(P_2) > P_r(P_1)$

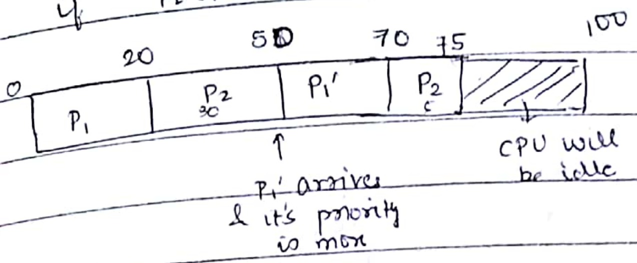


But, it's deadline was 50 .

So, this is not able to meet the deadline

Teacher's Signature

→ if $P_r(P_1) > P_r(P_2)$

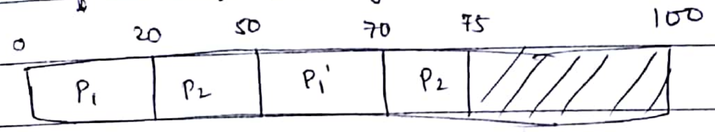


↳ May meet deadline sometimes, may miss deadline sometimes.

2.) Rate Monotonic Priority Algorithm

Rate of arrival : P_1 → 50
P_2 → 100

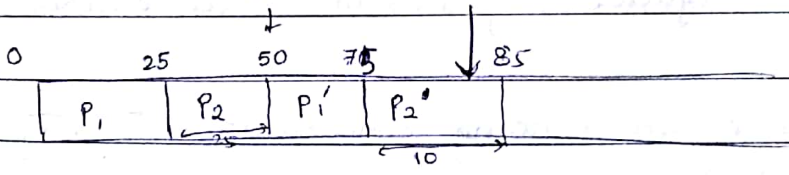
↳ Process with lower periodicity ~~should~~ will 've high priority. (they'll come & more frequently)



Eg.	P_1	P_2
Periodicity	50	80
Burst time	25	35

CPU Utilization
50%
46%
} 96% ⇒ {still, we may able to meet deadline using some algo}

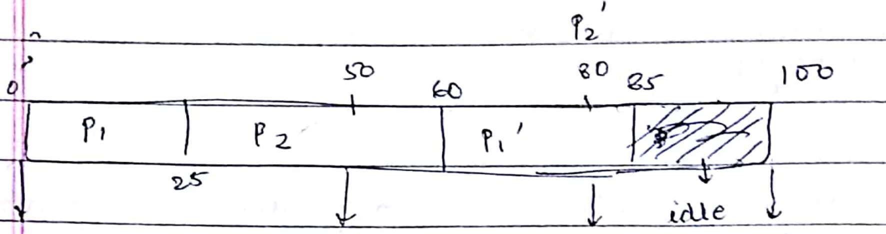
P_1' P_2' arrives ➡ Deadlines are not met.



3.) Earliest Deadline First (EDF)

Priority is assigned dynamically based on deadline

P_1	P_2
50	80
25	35

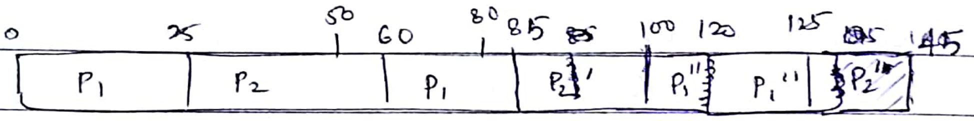


earliest deadline : 50

deadline
$P_1' \rightarrow 100$
$P_2 \rightarrow 80$
⇓
continue P_2

$P_1'' \rightarrow$ 1~~0~~0
$P_2'' \rightarrow$ ~~160~~ 160

$P_1'' \rightarrow 150$
$P_2' \rightarrow 160$



↓
~~पहले स्टार्ट करेंगे (85 tak jaayega)~~
~~phir px. dekhenge~~
(arrival par hi priority check karna h)