Real-time Scheduling

Results must be produced within a specific deadline.

Soft-real time

Meeting deadlines is desirable.

Hard real-time

Most stringent requirements.

Missing deadlines could be catastrophic

How to provide R.T Scheduling?

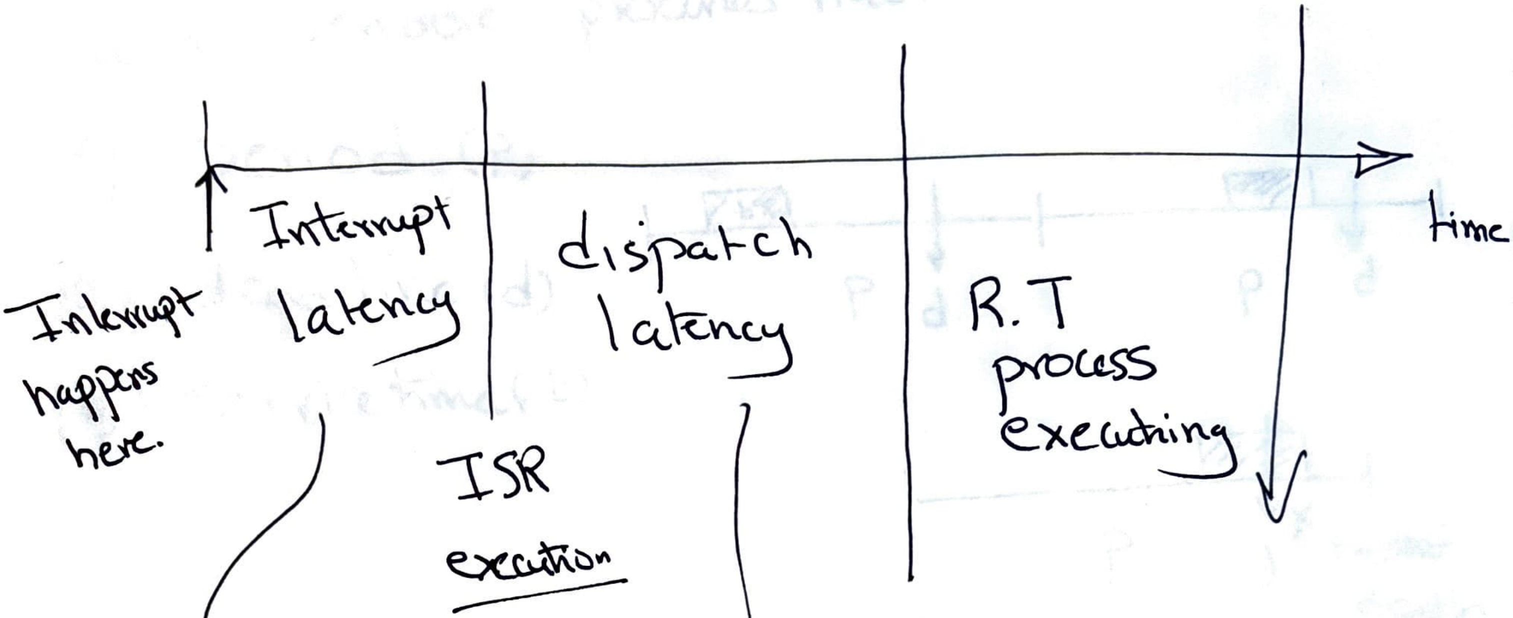
1. Preemptive - Priority-based Scheduling
2. Preemptive - Kernels

Non preemptive kernels disallow preemption when a process is running in Kernel mode.

Complex to design.

preemption points: Check for other waiting processes. (with higher priority)

3 Latency (minimize latency).



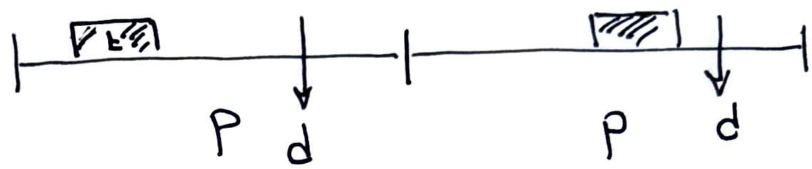
Time from an interrupt until ISR execution.

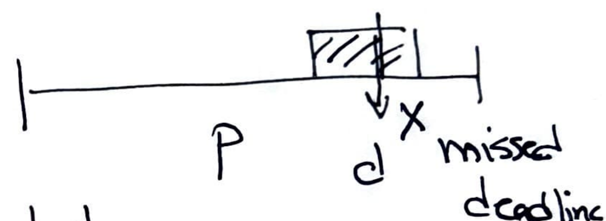
time needed to start the process.

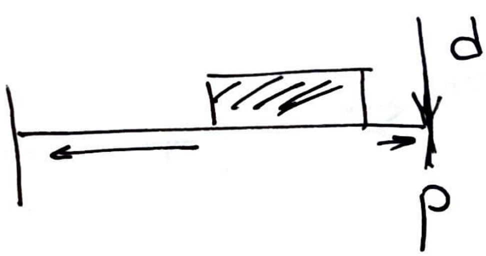
Periodic Processes

Each periodic process has:

1. period (p)
2. deadline (d)
3. service time (t)







R.T Schedulers know such characteristics for processes

admit / reject.

Before

	arrival Time	Servic Time
A	2	4

Now

A (20 , 50)

$\downarrow$ t $\quad$ $\downarrow$ p

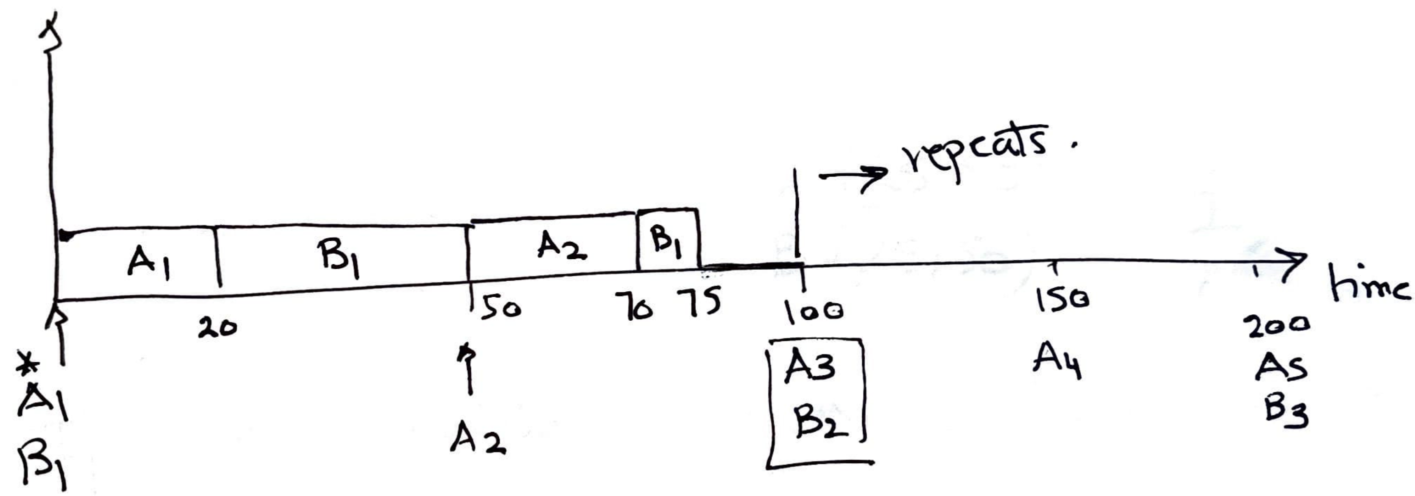
1 Rate Monotonic Scheduling

Static priority with preemption.

priority is inversely prop. to period.

A (20, 50)

B (35, 100).

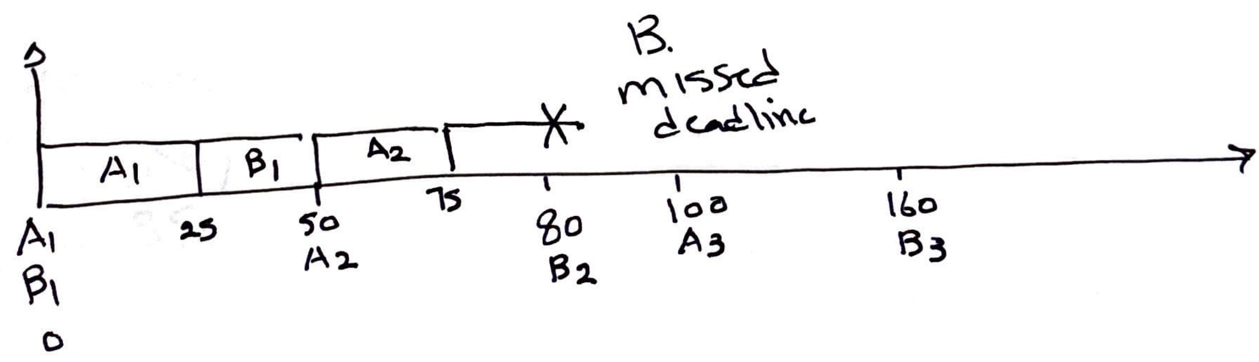


$$\frac{20}{50} + \frac{35}{100} = \frac{75}{100}$$

A (25, 50)

B (35, 80)

$$\frac{25}{50} + \frac{35}{80} = 0.94$$



CPU Utilization bound $\quad n\left(2^{1/n} - 1\right)$

n : number of process

$n=1$	1
$n=2$	0.82
$n=3$	0.779
⋮	
$n=\infty$	0.69

A(25, 50)
B(35, 80) $\Big\}$ $\boxed{0.94}$

A (25, 50)
B (25, 50) $\quad$ 1

If CPU utilization of process < bound ⇒ meet deadline

else
we do not know (need to try it).

RMS is optimal under static priorities.

A (20, 100)
B (40, 150)
C (100, 350)

$$\frac{20}{100} + \frac{40}{150} + \frac{100}{350} = 0.753$$

$0.753 < 0.779 \Rightarrow$
under RMS all process would meet.

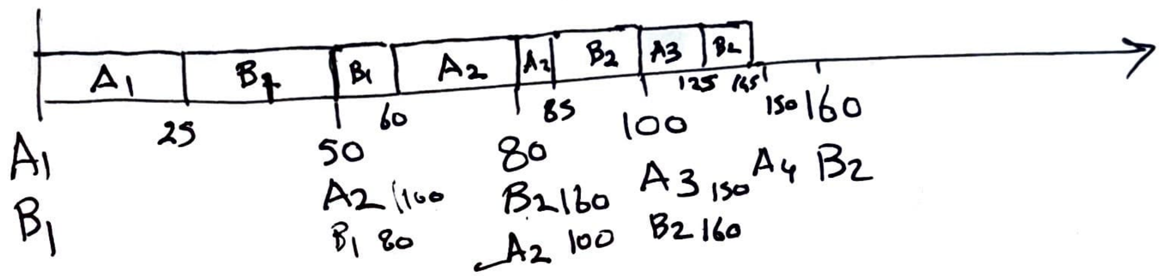
2 Earliest Deadline First

Dynamic Priority.

Earlier the deadline, higher the priority.

A(25, 50)

B(35, 80)



3 Proportional Share Scheduling

T shares to allocate.

Each process is assigned n shares.

Ex:

100 Shares

A 50

B 20

C 15

85/100

→ D asks 30? No