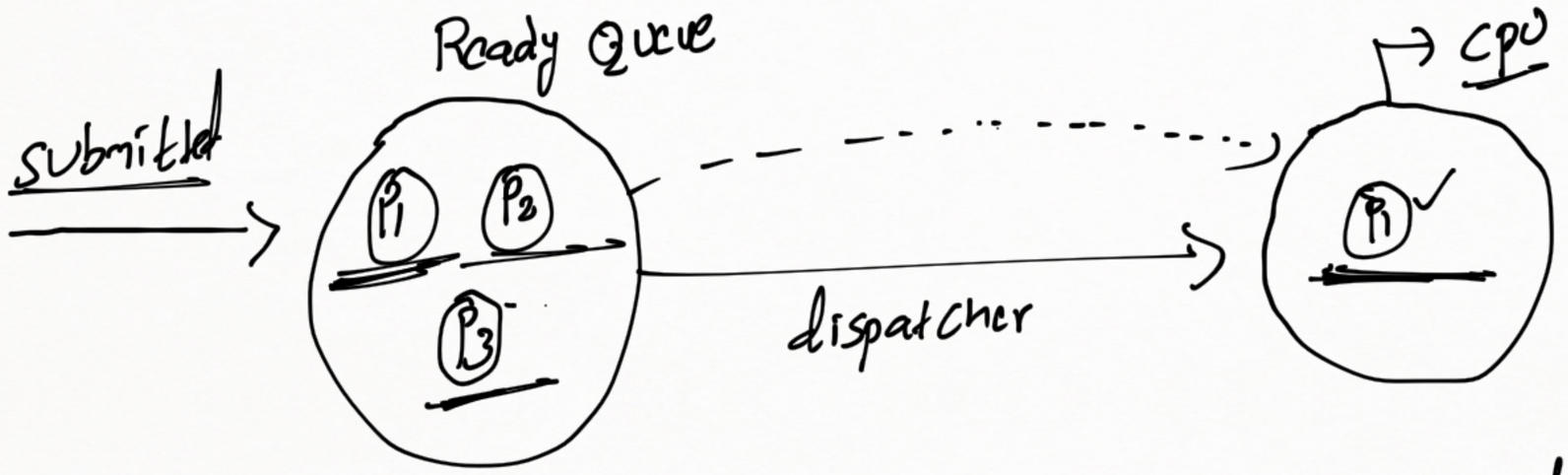
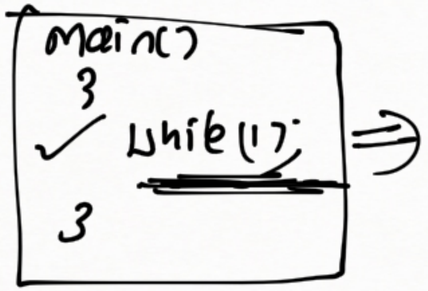
Date: 14-10-21 :

SJF



⇒ Job which is taking less amount of time to complete



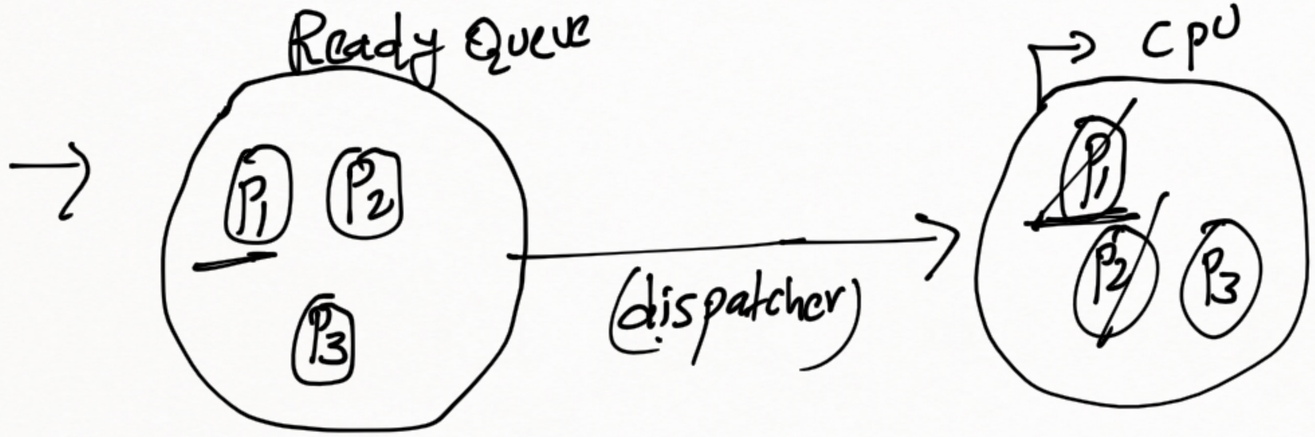
priority

P_1 - 1 ✓

P_2 - 2

P_3 - 3

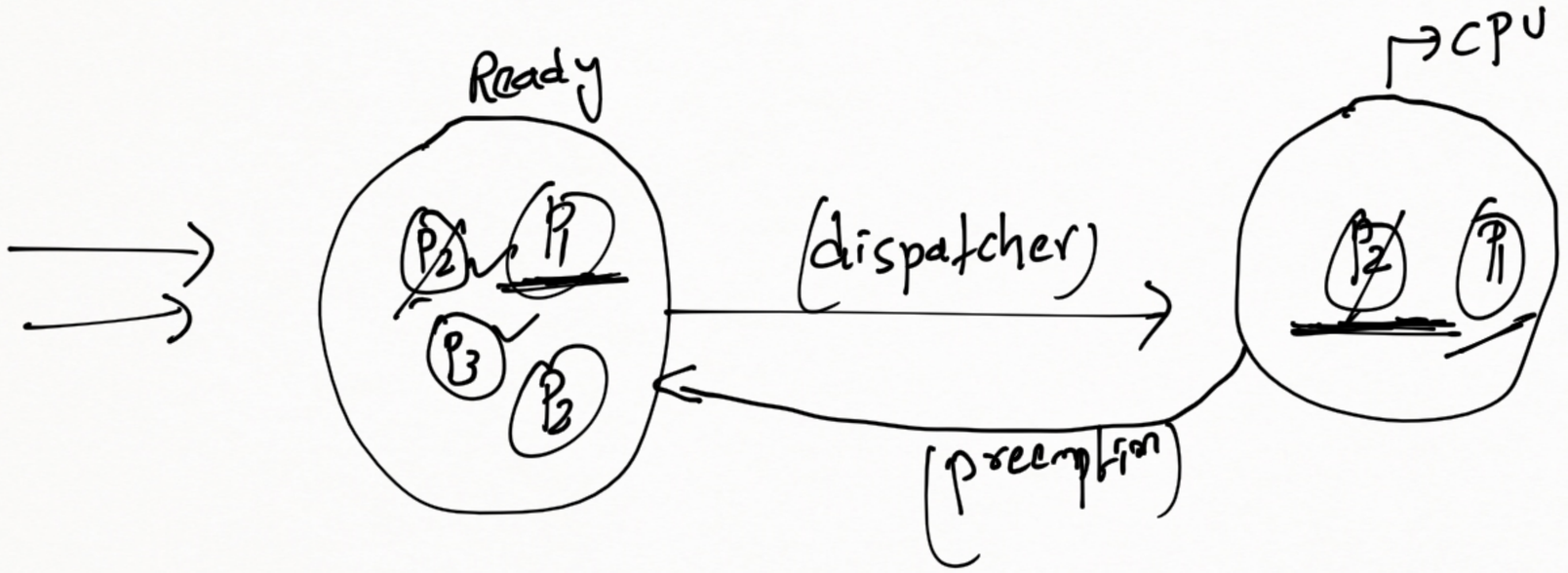
⇒ Lower number having higher priority



⇒ The highest priority process is under execution then Lowest priority process must wait

→ R.T
→ W.T
→ T.A.T

} Increase



⇒ If a lower priority process is under execution, at that time if higher priority process ready then preemption happens

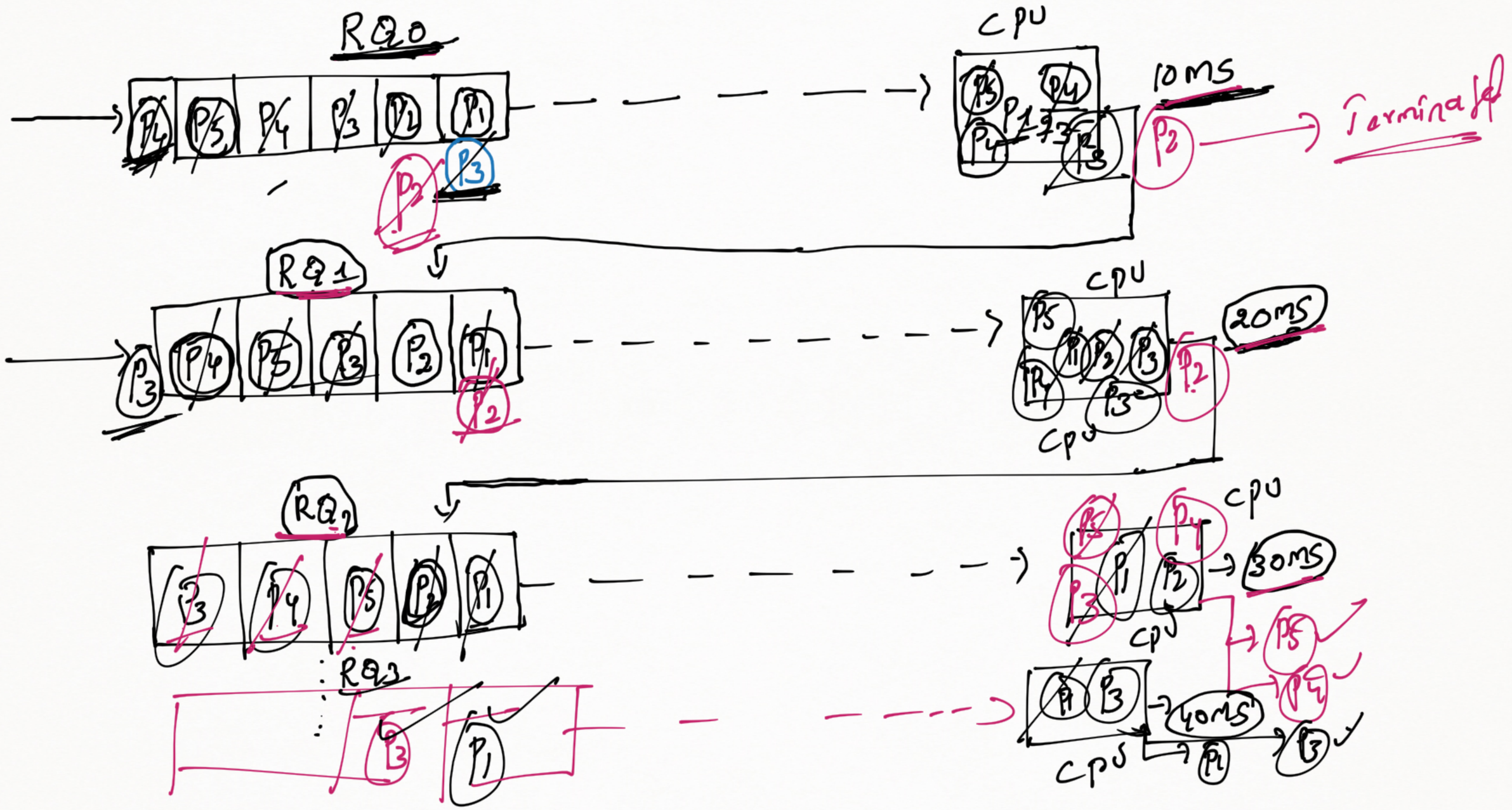
MULTI LEVEL FEEDBACK QUEUE

→ it is a combination of Round Robin plus priority scheduling policy

→ multiple Queues with different prioritys.

→ Round robin applicable in each Queue Jobs

→ If a process hogging CPU with in the time slice than that process priority increase other wise process priority decreases.


RQ0
CPU
10MS
Terminated
RQ1
CPU
20MS
RQ2
CPU
30MS
RQ3
CPU
40MS

⇒ In OS by default 80 is the priority number assigned for each process once started

⇒ Total 0 to 99 (100 priority levels)

⇒ $ ./a.out &

$ ps -l ↵

PRI	NI
80	0
85	(5)

NI ⇒ niceness value

changing niceness value possible using nice & Renice commands

⇒ nice (Run a program with modified scheduling priority)

Ex: $ nice -5 ./a.out& (+ve)

$ nice --5 ./a.out& (-ve)

⇒ nice nos range is -20 To +19 ✓

✓ -20 To +19 ⇒ 40 niceness values

PRI NI & nice -19 ./a.out& +ve

PRI	NI
85	5

PRI	NI
75	-5

PRI	NI
80	0

PRI NI
99 19
60 -20
(19)
(-20)
(19)
(-20)

⇒ renice (alter priority of Running process)

$ ./a.out &
↳ pid

PRI	NI
80	0

$ renice 5 pid
(+ve)

PRI	NI
85	5

$ renice -5 pid
(-ve)

PRI	NI
75	-5

renice niceness value (~~-20 To +19~~)

— 20 TO +19