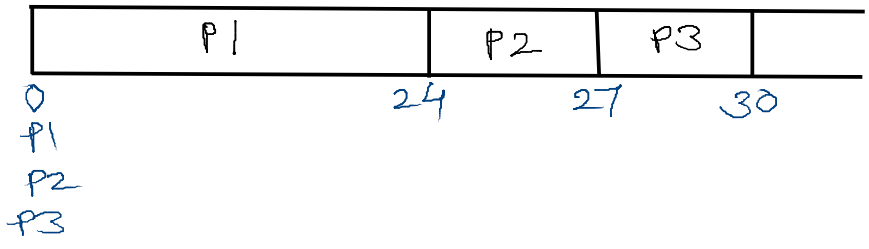
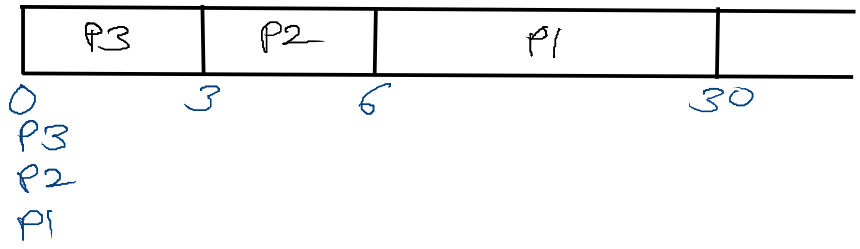
First Come First Serve (FCFS)
(Non preemptive scheduling)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	24	0	0	24
P2	0	3	24	24	27
P3	0	3	27	27	30

Process	Arrival	CPU Burst	WT	RT	TAT
P3	0	3	0	0	3
P2	0	3	3	3	6
P1	0	24	6	6	30

Gantt's chart



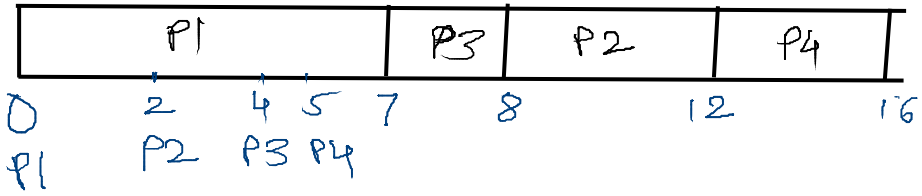


Convoy effect – due to arrival of longer process first, waiting time of all remaining processes is increased

Shortest Job First (SJF)

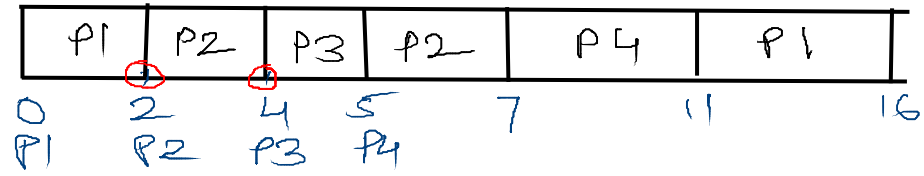
Non Preemptive

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	7	0	0	7
P2	2	4	6	6	10
P3	4	1	3	3	4
P4	5	4	7	7	11



Preemptive
Shortest Remaining time First

Process	Arrival	CPU Burst	Remain time	WT	RT	TAT
P1	0	7	~~5~~ 0	9	0	16
P2	2	4	2 0	1	0	5
P3	4	1	0	0	0	1
P4	5	4	~~4~~ 0	2	2	6
					0.5	



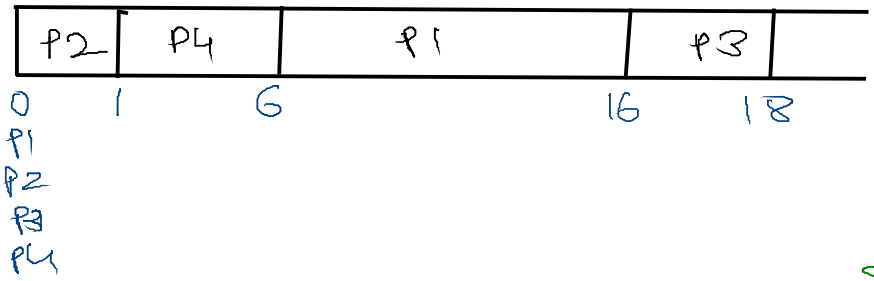
Starvation – due to bigger CPU time (burst), process will not be scheduled for longer time

Priority

Non-Preemptive

highest → lowest ↓

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	0	1	1	0	0	1
P3	0	2	4	16	16	18
P4	0	5	2	1	1	6

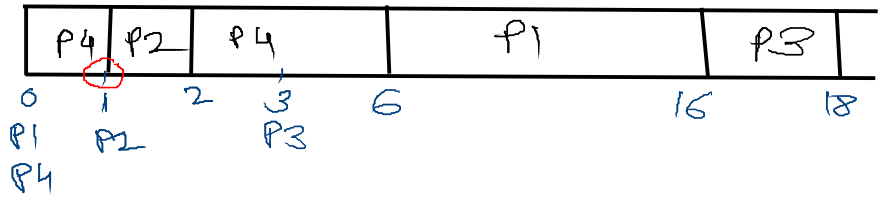


- P1(7)
- P2(9)
- P3(6)
- P4(3)
- P5(8)
- P6(7)
- P7(2)

→ P1, P4, P3, P7, P6, P5, P2

Preemptive

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	1	1	1	0	0	1
P3	3	2	4	13	13	15
P4	0	5	2	1	0	6



Starvation – due to less priority process is not getting scheduled on CPU for longer time

Aging – increase the priority of low priority process gradually.

Round Robin (RR)

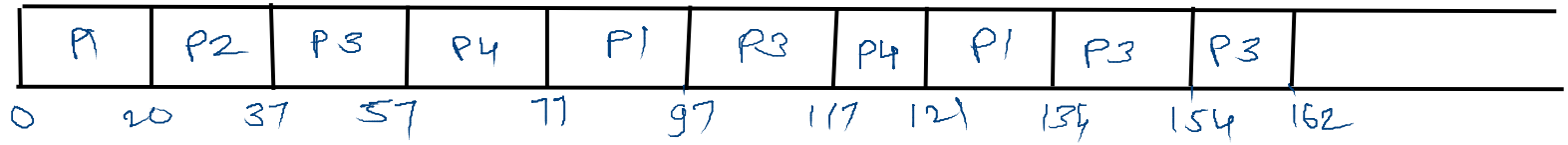
(Preemptive)

Process	CPU Burst
P1	53
P2	17
P3	68
P4	24

33, 13, 0
0
48, 28, 8
4, 0

	WT	RT	TAT
P1	0+57+24	0	134
P2	20	20	37
P3	37+40+17	37	162
P4	57+40	57	121

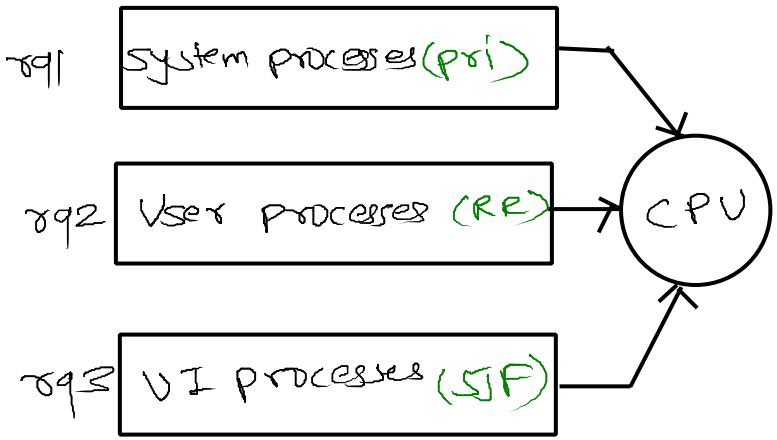
Time quantum = 20



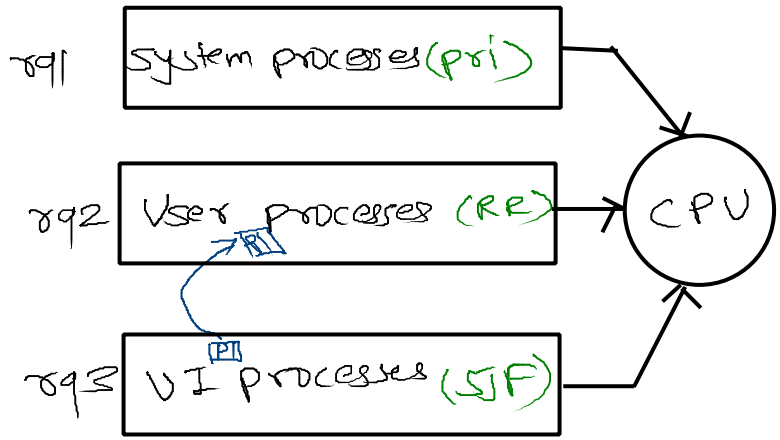
P1
P2
P3
P4

TQ = 100
↳ FCFS
TQ = 4
↳ CPU overhead ↑

Multilevel Ready queue



Multilevel feedback ready queue



Linux - CFS (Completely Fair Scheduler)

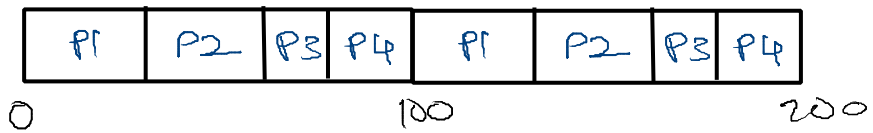
Fair share / Proportional share

- CPU time is divided into time slices (epoch)
- Every process has some priority
- Priority of process is decided by its nice value
- nice value – −20 to +19
  (Highest) (Lowest)

Process	Nice Value
P1	10
P2	10
P3	10
P4	10

Epoch time = 100

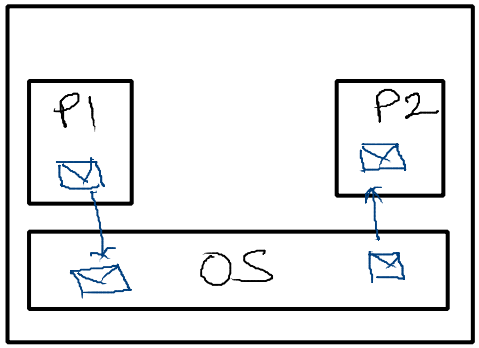
Process	Nice Value
P1	5
P2	5
P3	10
P4	10

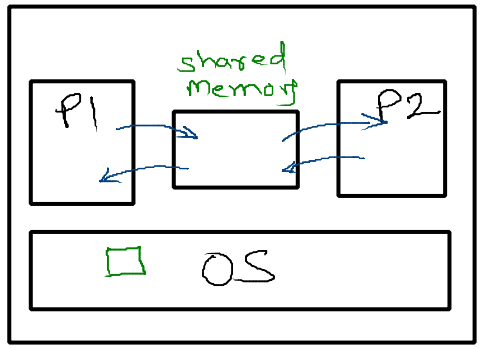


IPC

- two model are available

- Message Passing
- Shared Memory

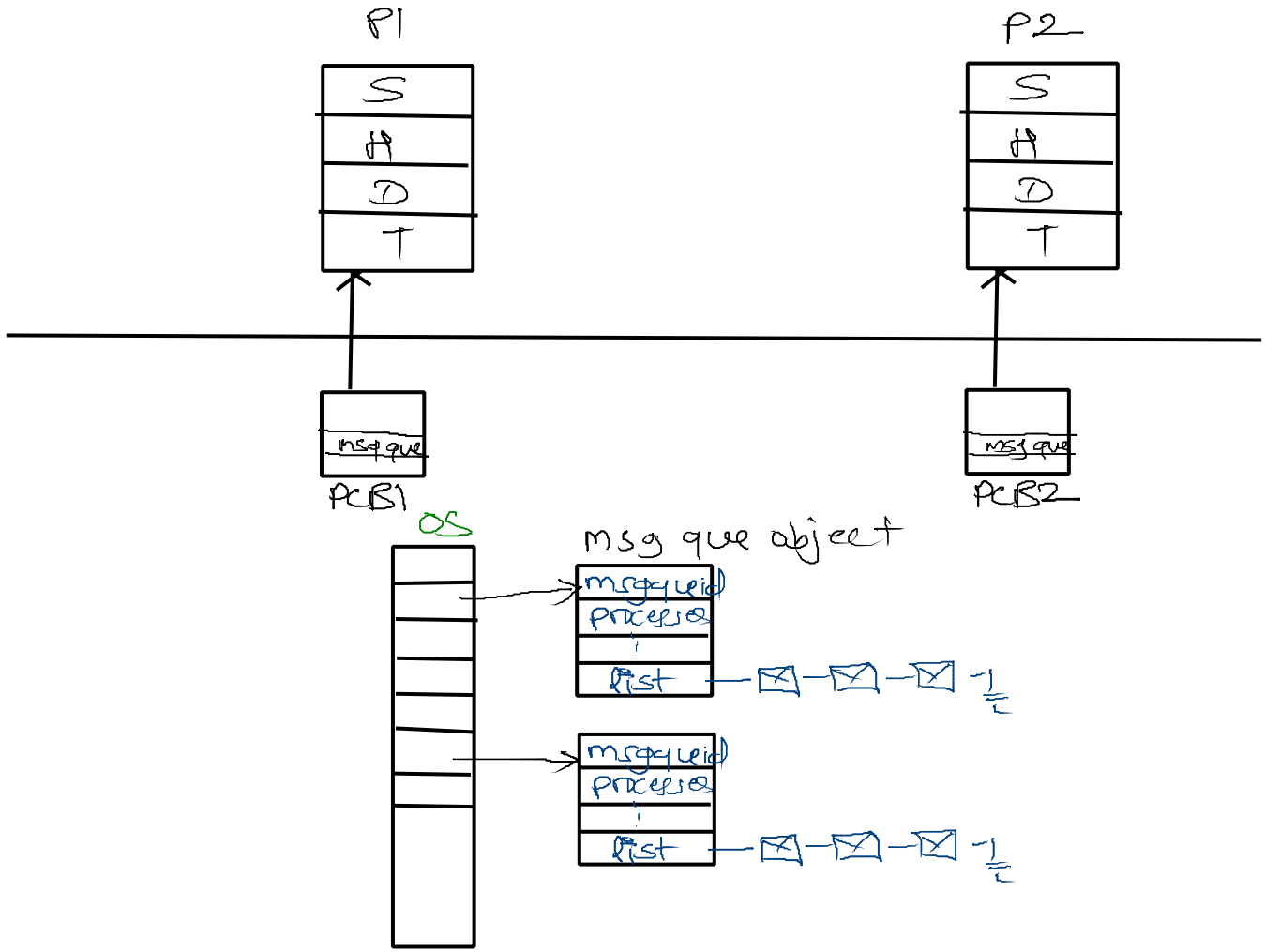




1. Message queue
2. signals
3. pipe
4. socket

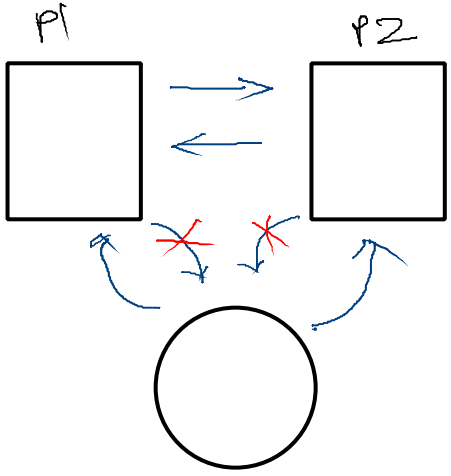
5. Shared memory

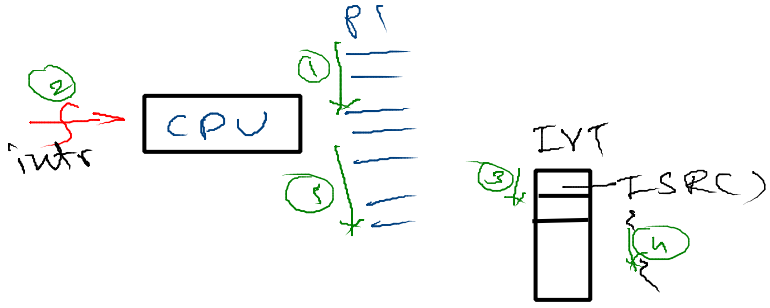
Message Queue



Signals

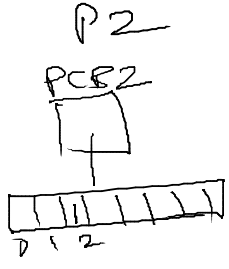
- there are 64 signals
- kill -l => list of signals

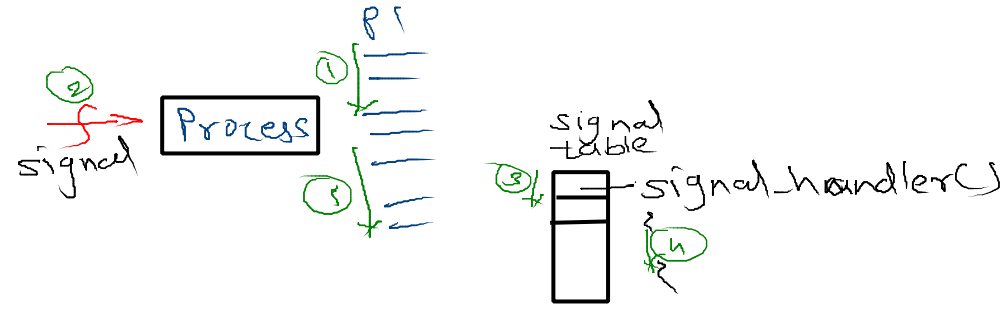




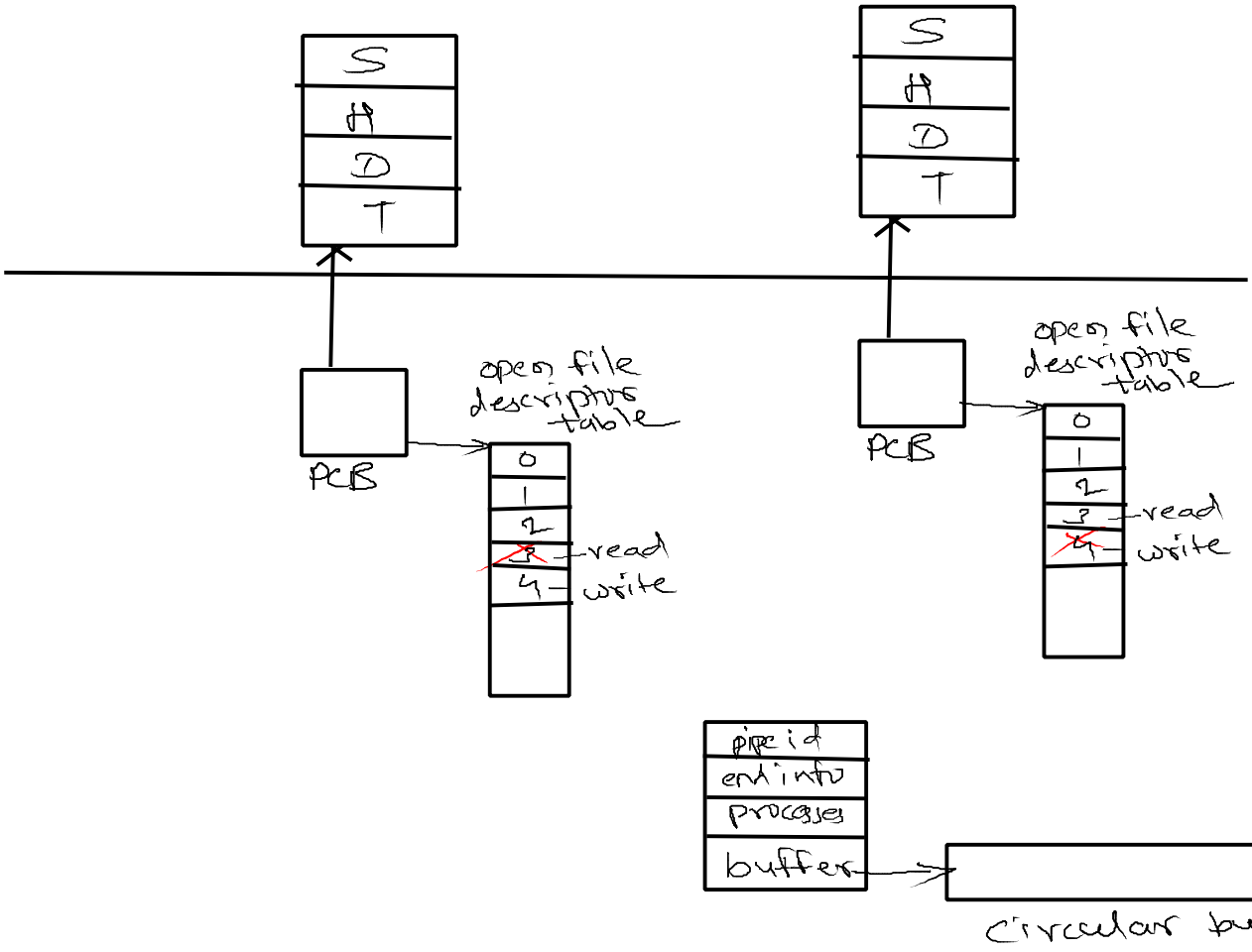
P1

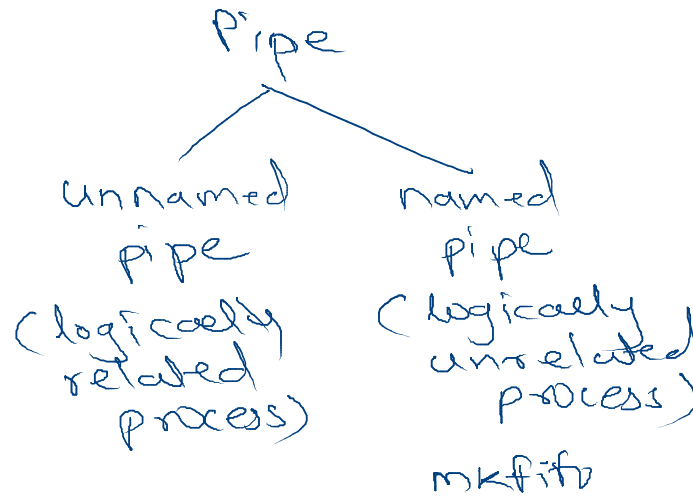
kill(P2, 2)





Pipe


S
H
D
T
S
H
D
T
PCB
open file descriptors table
0
1
2
3 — read
4 — write
PCB
open file descriptors table
0
1
2
3 — read
4 — write
pipe id
end info
process
buffer
circular buffer


Pipe
unnamed pipe
(logically related process)
named pipe
(logically unrelated process)
mkfifo