Sunbeam Institute of Information Technology
Pune and Karad

Module - Concepts of Operating System

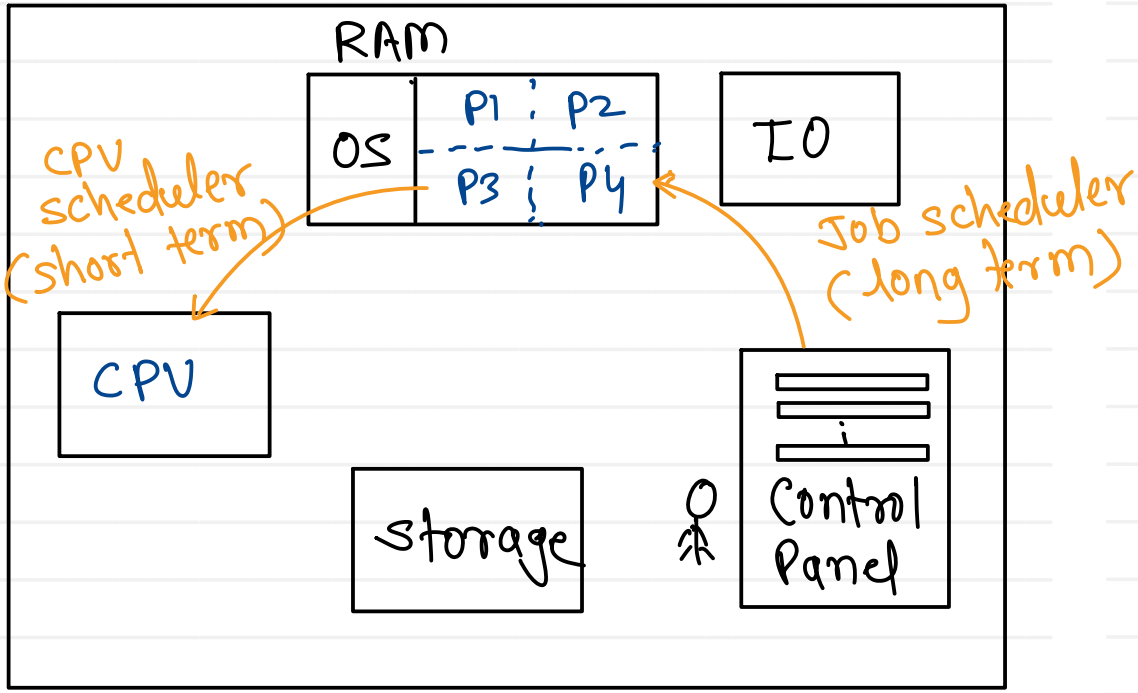
Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Types of Operating system

1) Desktop OS (GPOS)
2) Server OS
3) Embedded OS
4) Real time OS (FreeRTOS / Xenomai)
5) Handheld OS
6) Distributed OS

Types of Operating system



1. Resident monitor
2. Batch systems
3. Multiprogramming system

- multiple programs are loaded into RAM (memory)

CPU burst : time spend by process on CPU
IO burst : time spend by process for IO

CPU burst > IO burst : CPU bound process
IO burst > CPU burst : IO bound process

- mixture of CPU bound & IO bound processes was loaded inside RAM

Degree of multi programming :
No. of processes which can be loaded into RAM

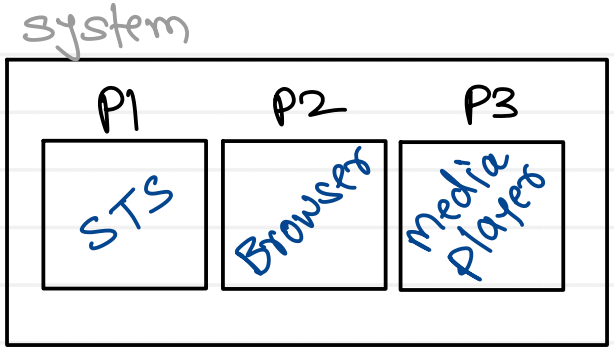
Types of Operating system

4. Time sharing system : (Multi tasking system)
- CPU time is shared in all the processes of RAM

Response time < 1 sec

- multiple tasks are getting executed in single system concurrently.

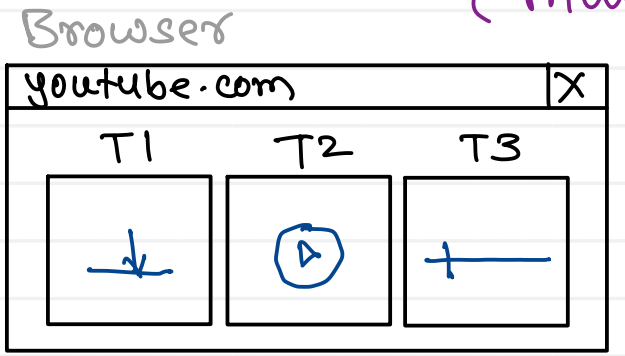
i) Process based multitasking



- system wide multitasking

ii) Thread based multitasking (multithreading)

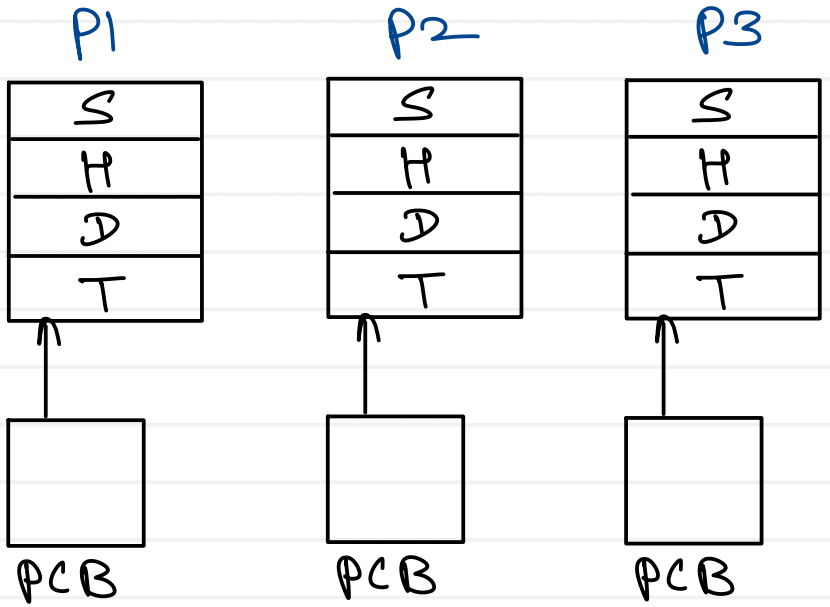
- light weight process



- multitasking within process

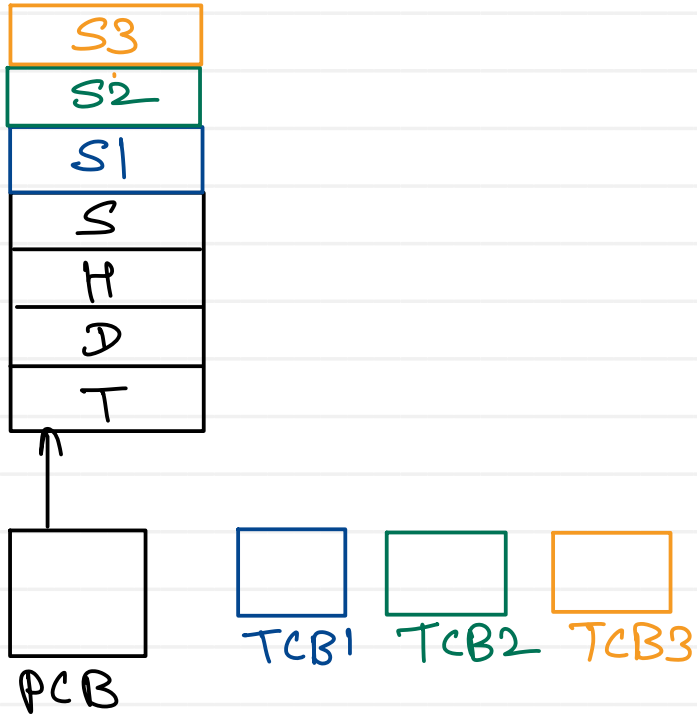
Process Vs Thread

- processes are created in system.



- for process all sections are created in user space + PCB is created in kernel space.

- threads are created always inside process



- for thread only stack section is created in user space & TCB is created in kernel space.
- remaining sections of user space are shared with parent process.

Types of Operating system

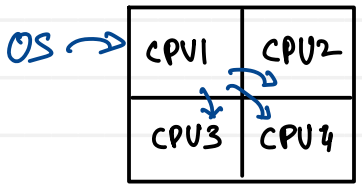
5. Multiprocessing systems :

- multiple processing units (CPU) are putted on single chip which is called as multiprocessor / multi core.

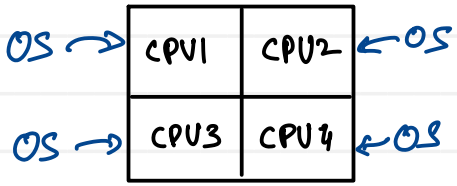
- OS can take advantage of this, by scheduling process for each core

- multiple processes can be executed parallely. (parallel systems)

i. Asymmetric multi processing

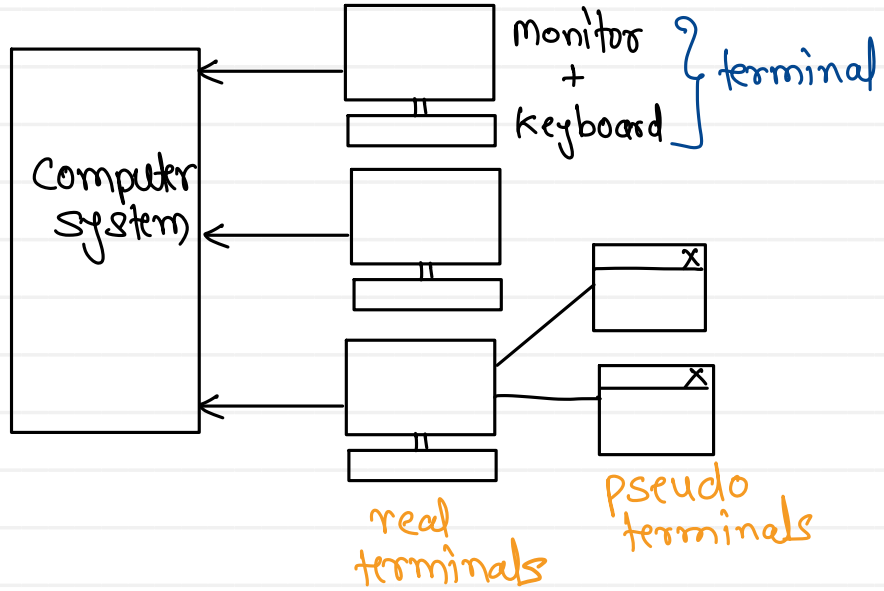


ii. symmetric multi processing



Windows Vista
linux 2.6

6. Multiuser system :




```
ps          uname       who
ps -e       uname -r    tty
top         uname -a    whoami
                        w
```

Context switching

Execution context :

- values of CPU registers

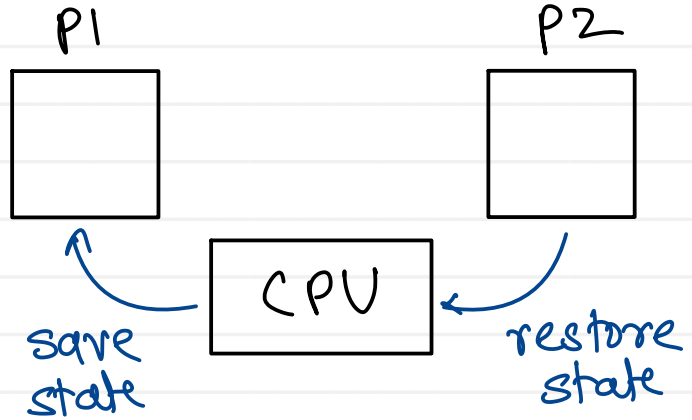
small memory elements present inside CPU

CPU dispatcher loads process on CPU. Execution context of selected from its PCB is loaded on CPU

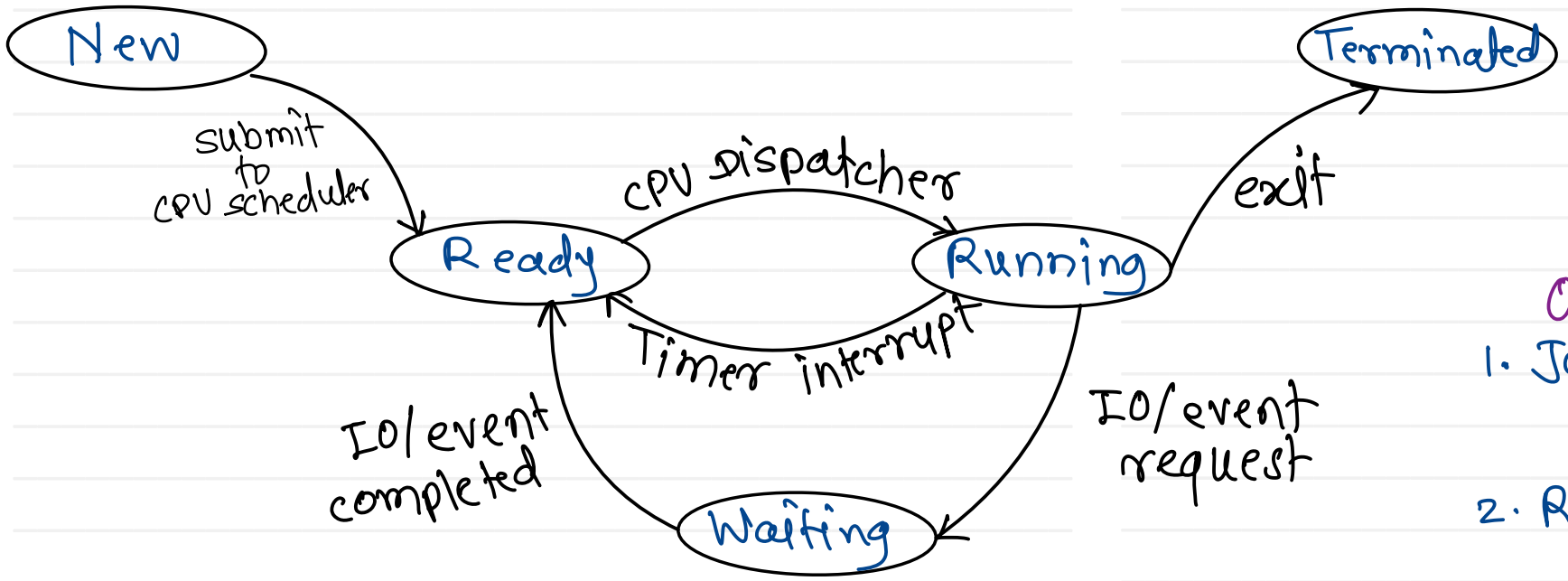
Context switching :

- changing the process of CPU

- execution context of running process is saved into its PCB & execution context of selected process is restored on CPU.



Process life cycle



OS data structures

1. Job queue / process list:
   - all processes of RAM
2. Ready queue:
   - processes which are in ready state.
3. Waiting queue(s):
   - processes which are in waiting state.

CPU scheduling

Cases in which CPU scheduler is called

1. Running → Terminated } voluntarily
2. Running → waiting
3. Running → Ready } forcefully
4. Waiting → Ready

CPU scheduling Criterias :

1. CPU Utilization (max)
2. Throughput (max)
   - amount of work done per unit
3. Waiting time (min)
   - time spend by process in ready queue

   WT = TAT − CPU burst
4. Response time (min)
   - time from arrival to first time execution

   RT = start time − AT
5. Turn Around Time (TAT) (min)
   - time spend by process inside RAM

   TAT = DT − AT

Types of scheduling :

1. Non pre emptive scheduling :
   CPU access is always given voluntarily (co operative process)
2. Pre emptive scheduling :
   CPU access is given forcefully

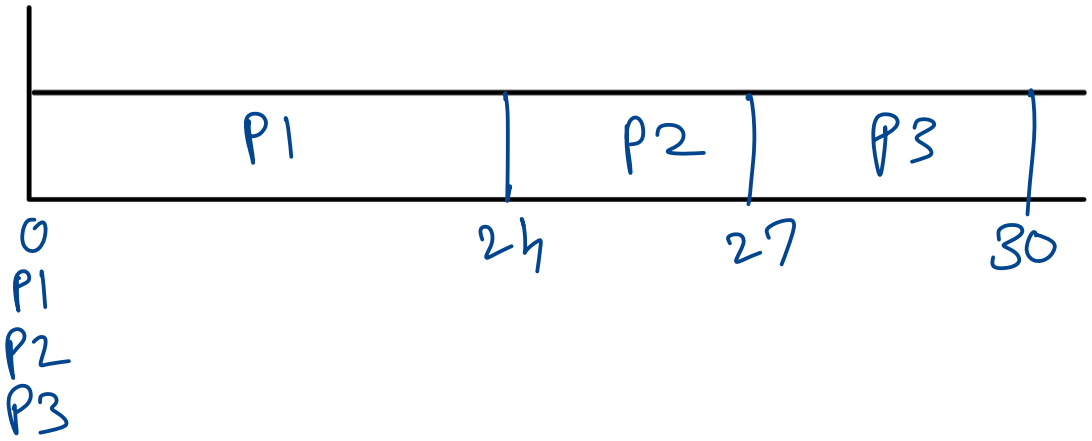
CPU scheduling algorithms :

1. FCFS
2. SJF
3. Priority
4. RR

FCFS (First Come First Serve) (Non preemptive)

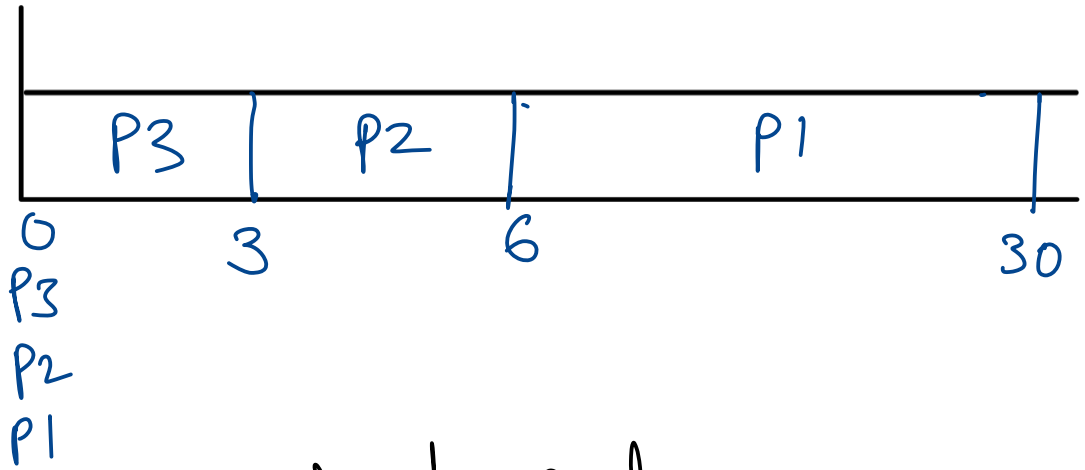
Process	Arrival	CPU Burst
P1	0	24
P2	0	3
P3	0	3

WT	RT	TAT
0	0	24
24	24	27
27	27	30



Process	Arrival	CPU Burst
P3	0	3
P2	0	3
P1	0	24

WA	RT	TAT
0	0	3
3	3	6
6	6	30



Convoy effect : due to arrival of longer processes all other processes need to wait

SJF (Shortest Job First)

Shortest Remaining Time First

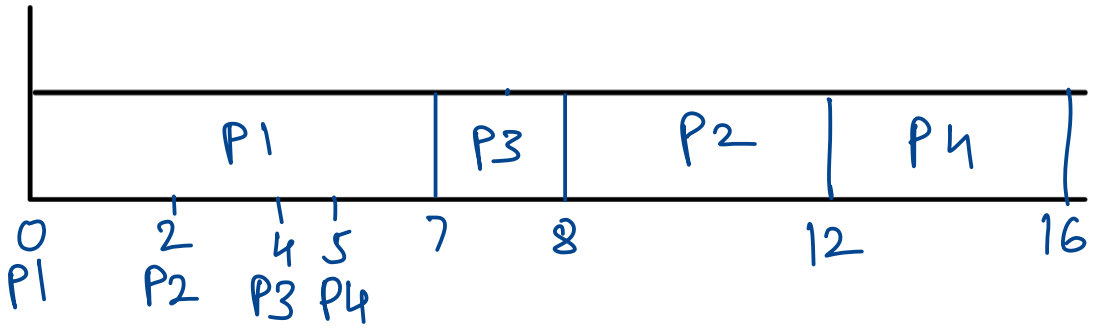
(Non Preemptive)

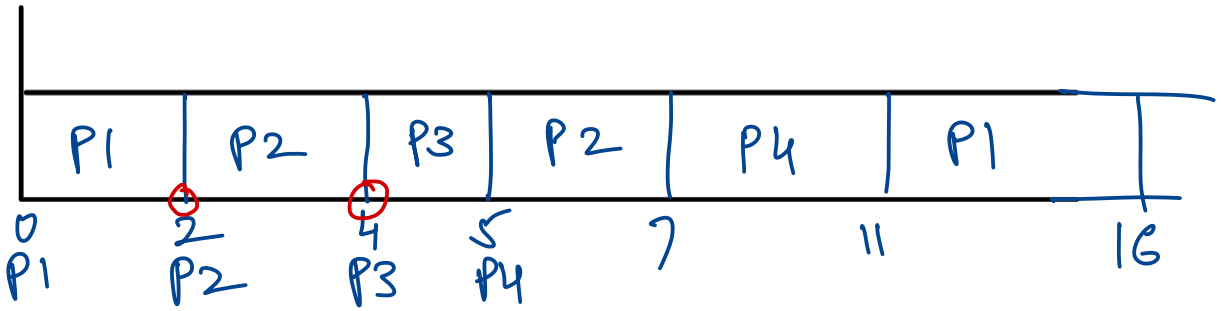
Process	Arrival	CPU Burst
P1	0	7
P2	2	4
P3	4	1
P4	5	4

WT	RT	TAT
0	0	7
6	6	10
3	3	4
7	7	11

(Preemptive)

Process	Arrival	CPU Burst	remaing time
P1	0	7	5
P2	2	4	2
P3	4	1	0
P4	5	4	





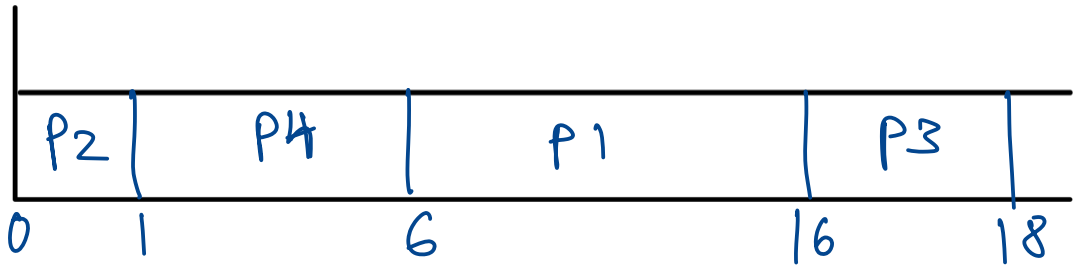
starvation :

due to longer CPU burst process do not get chance to execute on CPU. (Needs to wait for longer duration)

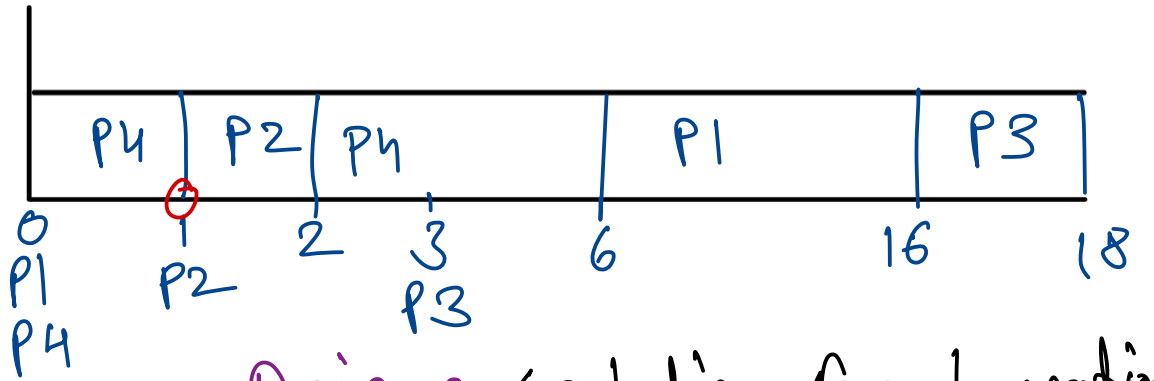
– there is no solution for starvation in SJF

Priority

Process	Arrival	CPU Burst	Priority
P1	0	10	3
P2	0	1	1 (H)
P3	0	2	4 (L)
P4	0	5	2



Process	Arrival	CPU Burst	Priority
P1	0	10	3
P2	1	1	1
P3	3	2	4
P4	0	5	2



Starvation :
- due to less priority, process do not get chance to execute on CPU (need to wait for longer duration)

Aging : (solution for starvation)
- priority of straved process is increased gradually, so that it will get scheduled.

RR (Round Robin) (pre emptive)

Process	CPU Burst
P1	53
P2	17
P3	68
P4	24

33, 13 X

X

48, 28, 8

4 X

0 + 57 + 24

20

37 + 40 + 17

57 + 40

Time Quantum: CPU time slice

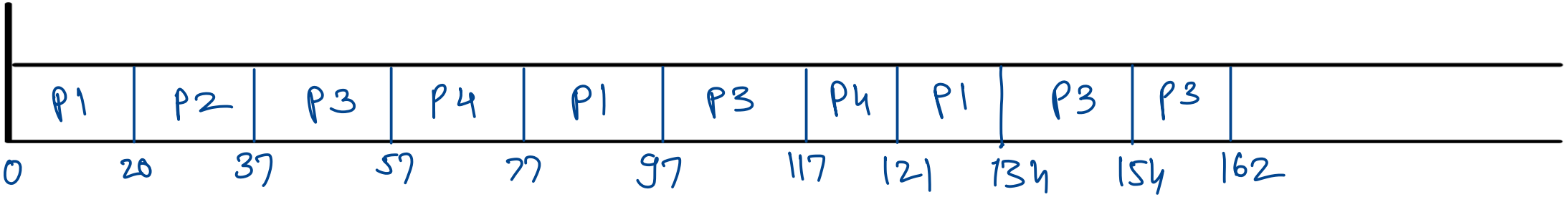
TQ = 20

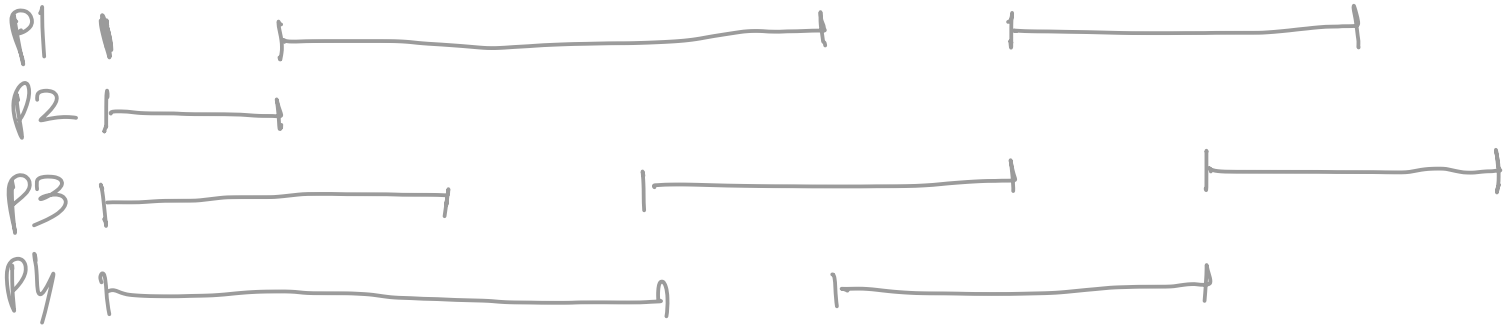
TQ = 100

↳ behave like FCFS

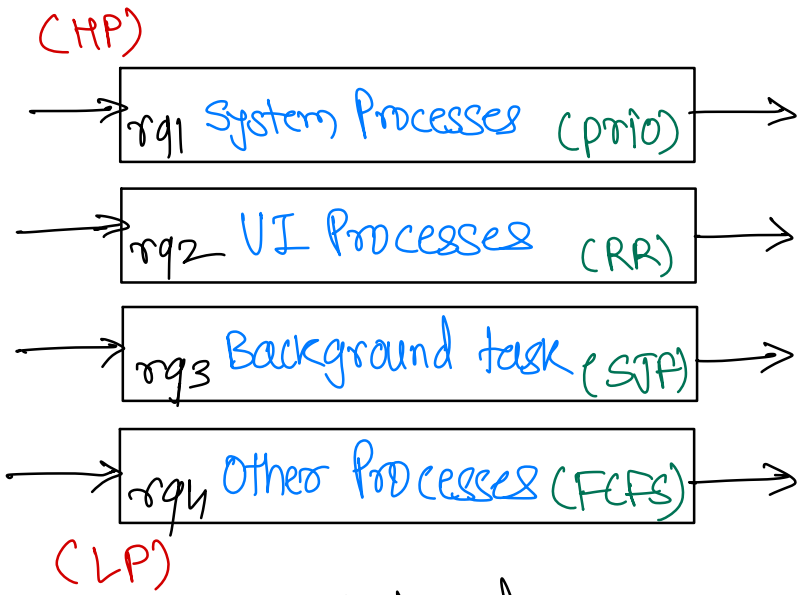
TQ = 4

↳ CPU overhead will increase

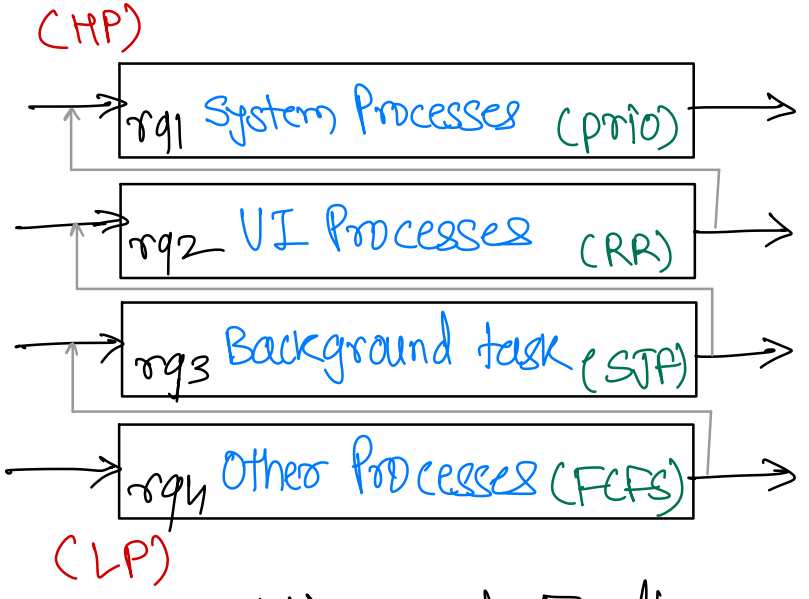




Multi Level Ready Queue



Multi level



Multi Level Feedback

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com