Operating System Concepts

06 July 2023 08:14

① Need of OS

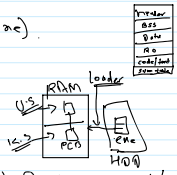
② Meaning O.S

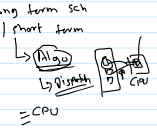
③ Compilation flow (.c → .exe)

④ Booting process

⑤ process management

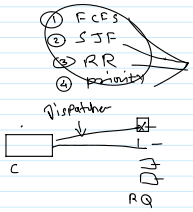
- What mean by process
- OS Evolution
- Life cycle process
  - New state → Job Queue → Job sch/long term sch
  - Ready state → Ready Queue → CPU sch/short term
  - Running state → CPU
  - Waiting state → Waiting Queue
  - Terminating state → PCB Destroy



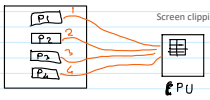


= CPU

- CPU scheduling algo
  - Cpu Utilization (max)
  - Throughput (max)
  - Waiting time (min)
  - Response time (min)
  - Turn Around time (min)
    - W.T + E.T



① FCFS CPU scheduling (It is non-preemptive)

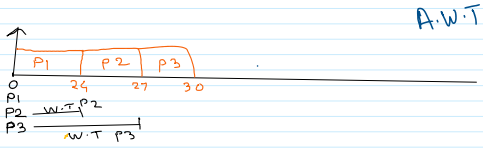


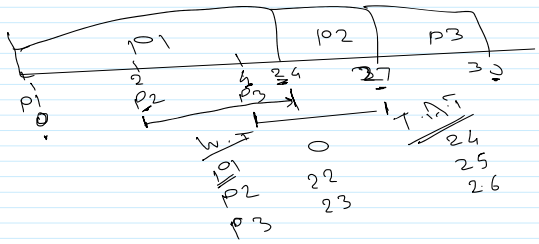
Screen clipping taken: 06-07-2023 09:21

Process	Arrival Time	CPU Burst	Wait Time	Turn Around Time
P1	0	24	0	24
P2	0	3	24	27
P3	0	3	27	30

A.T.A.T = $\frac{P1+P2+P3}{3}$ = $\frac{24+27+30}{3}$ =

A.W.T = $\frac{P1+P2+P3}{3}$ = $\frac{0+24+27}{3}$ = $\frac{51}{3}$ = 17





② SJF CPU Algo (minimum waiting time)

① Non-Preemptive (SNTF) (Arrival time diff fail) so we design SRTF

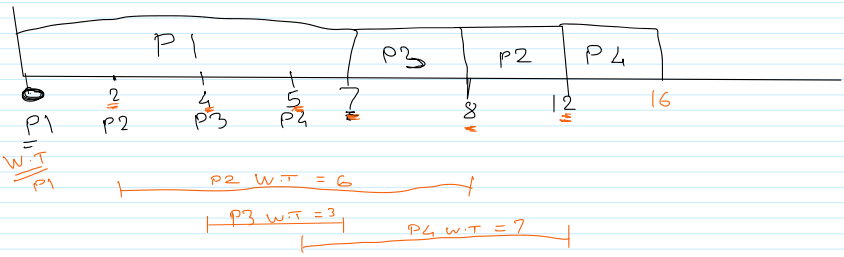
② Preemptive (SRTF)

SJF/SNTF scheduling

Process	Arrival Time	CPU Burst	Wait Time	Turn Around Time
P1	0	7 x	0	7
P2	2	4 x	6	10
P3	4	1 ✓	3	4
P4	5	4 ✓	7	11

P3	4	1	3	4
P4	5	4	7	11

Screen clipping taken: 06-07-2023 09:50



(2) SRTF (Preemptive) (get CPU Forcefully)

Screen clipping taken: 06-07-2023 10:10

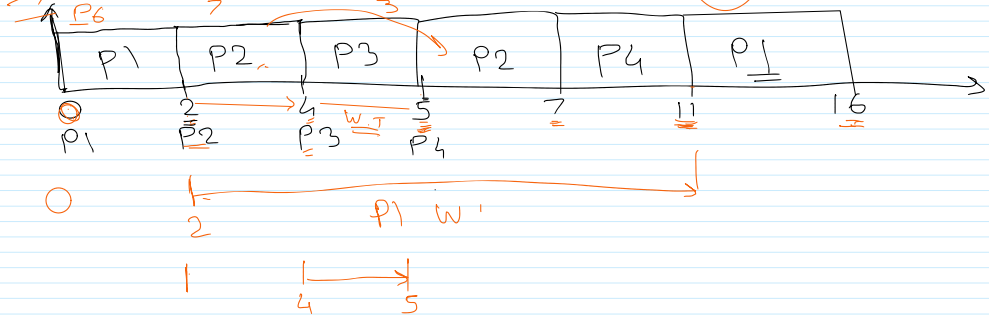
SRTF Scheduling (Preemptive) SJF

Process	Arrival Time	CPU Burst	Remaining Time	Wait Time	Turn Around Time
P1	0	7	5, 2, 0	9	16
P2	2	4	2, 0	1	5
P3	4	1	0	0	1
P4	5	4	4, 1, 0	2	6
P5	6	2			
P6	7	3			

Avg. Waiting Time = Sum of process waiting time / No of process

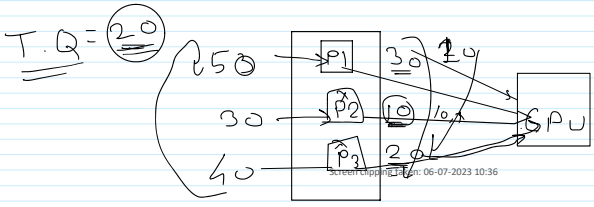
$= \frac{9+1+0+2}{4}$

$= \frac{12}{3}$



(3) Round Robin (Preemptive)

T.Q = 20



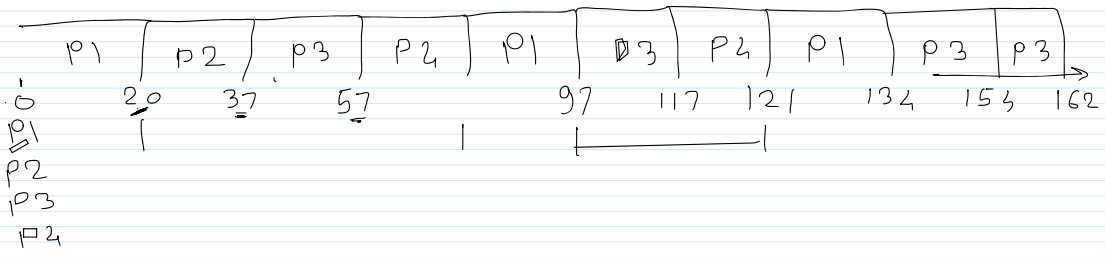
Screen clipping taken: 06-07-2023 10:36

Round Robin Scheduling

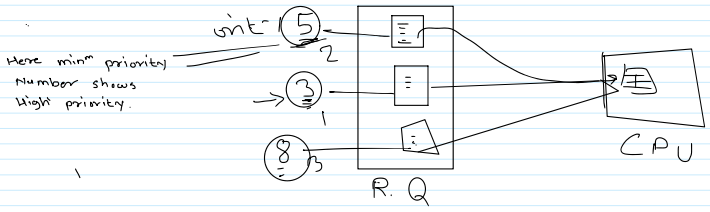
T.Q = 20

Process	CPU Burst	Remaining Time	Wait Time	Response Time
P1	53	33, 13, 0		0
P2	17	0		20
P3	68	48, 28, 8, 0		37
P4	24	4, 0		57

A.R.T = $\frac{0+20+37+57}{4}$

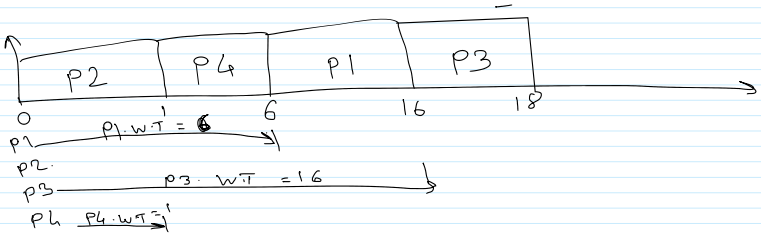


(4) Priority (Preemptive)

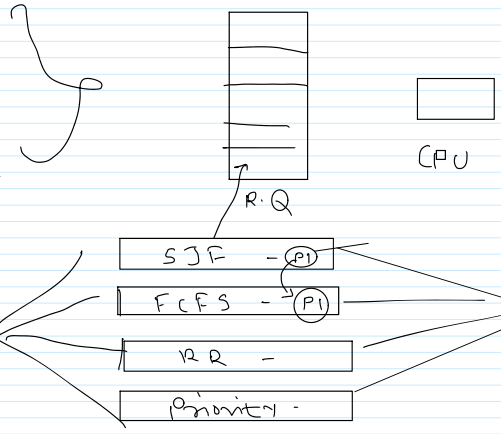


Process	Arrival Time	CPU Burst	Priority	Wait Time
P1	0	10	3	6
P2	0	1	1	0
P3	~~0~~ 3	2	4	16
P4	0	5	2	1

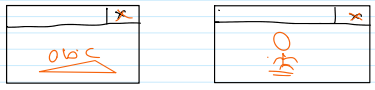
Screen clipping taken: 06-07-2023 10:54



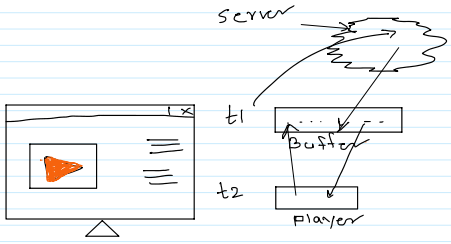
1. FCFS -
2. SJF -
3. RR -
4. Priority -



* IPC



These two processes Not depend on each other it known as Independent Process

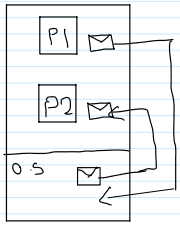


Here process P1 depends or share data with other process is known as Co-operative process

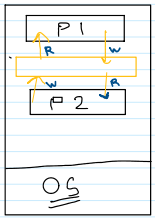
* Linux IPC

1. Shared memory
2. pipe
3. msg Queue
4. Signal
5. Sockel

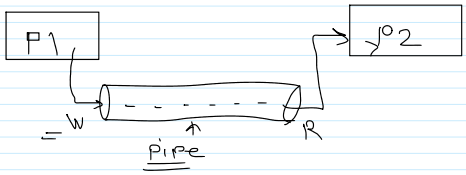
1. message Passing Model



2. Share memory Model

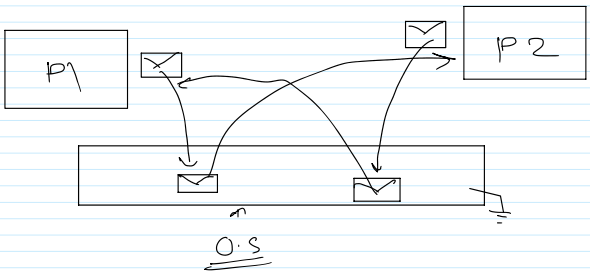


1. pipe



- Named pipe
- Unnamed pipe

2. message Queue



③ Signal

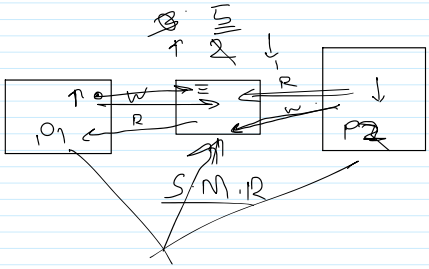
```
while (1)
{
  x+y;
}
```

Ctrl + C

① - System

② Software

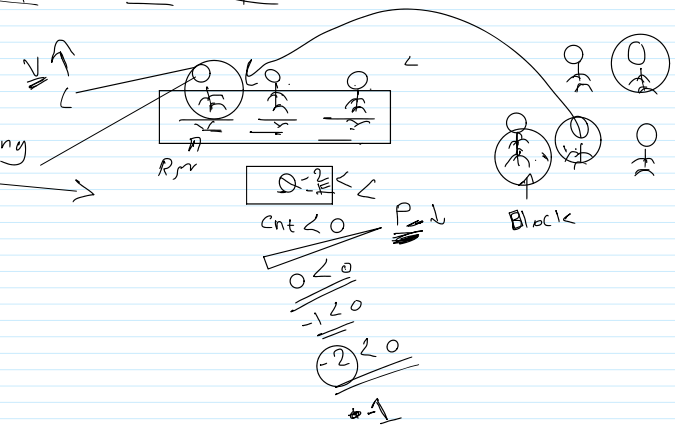
① SH



① Semaphore (Counter.) ↑ ↓

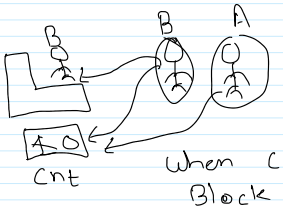
— Decr : P oprn : wait oprn

① Classic Counting



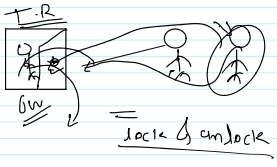
② Binary Semphore

0 or 1

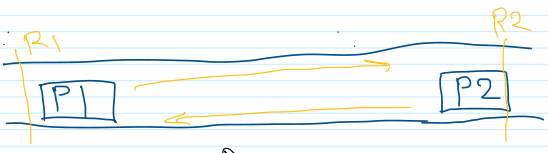


When cnt 0 then A is Block

② Mutex (lock and unlock)



1. Mutual exclusion
2. No Resource Preemp
3. Hold & wait



4. Circular wait

Here P1 Hold R1 & wait R2
& P2 Hold R2 & wait R2
So Here Hold & wait