Webinar ⑪ : 25th July '21 (Sunday)

DATE: / /
PAGE:

PRIORITY BASED CPU SCHEDULLING

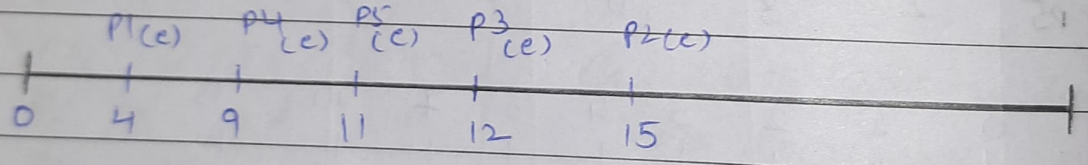
↳ priority can be based on either small first or high first.

↙ ↘

pre-emptive non pre-emptive

Process ID	AT	BT	Priority
P1 ✓	0	4	2
P2 ✓	1	3	3
P3 ✓	2	1	4
P4 ✓	3	5	5
P5 ✓	4	2	5

Non pre-emptive Approach.



At 0 P1 arrived ⇒ P1 executed

At 4 All here ⇒ P4 and P5 ✓

⇒ P4 ✓

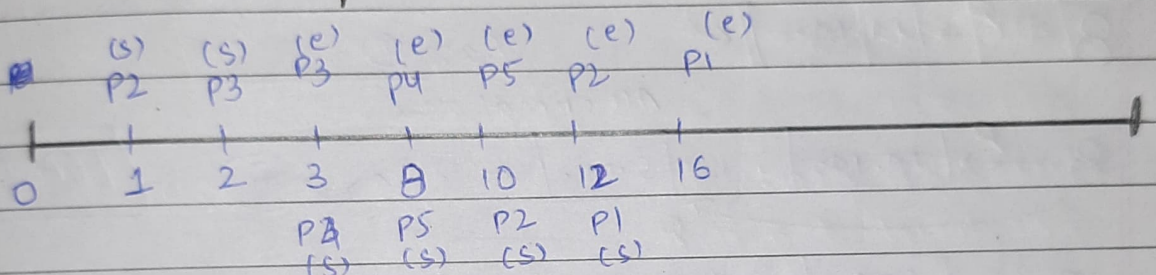
At 9 P5 ✓

(PID)	AT	BT	Pr.	ET	TAT	WT	RT
P1	0	4	2	4	4	0	0
P2	1	3	3	15	14	11	11
P3	2	1	4	12	10	9	9
P4	3	5	5	9	6	1	1
P5	4	2	5	11	7	5	5

DATE: / /
PAGE:

Process ID	AT	BT	Pr.	ET	TAT	WT	RT
P1 ✓	5	4	2	16	11	7	7
P2 ✓	1	~~3~~ 2	3	12	11	8	0
P3 ✓	2	~~1~~ 0	4	3	1	0	0
P4 ✓	3	~~5~~ ~~4~~ 0	5	8	5	0	0
P5 ✓	4	2 0	5	10	6	4	4

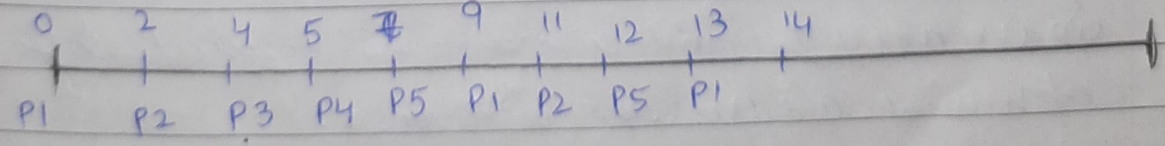
Pre-emptive Approach



ROUND ROBIN ALGORITHM

P-ID	AT	BT	ET	TAT	WT	RT
P1	0	~~5~~ ~~3~~ 1	14	14	9	0
P2	1	~~3~~ 1	12	11	8	0
P3	2	1	5	3	2	2
P4	3	2	7	4	2	2
P5	4	~~3~~ 1	13	9	6	3

Time quantum = 2



P1 P2 P3 P4 P1 P2 P3 P4 .... what we think!

What it actually is!

P1	0
P2 P3 P1	2
P1 P4 P5 P2	4
P4 P5 P2 P1	6

and so on...

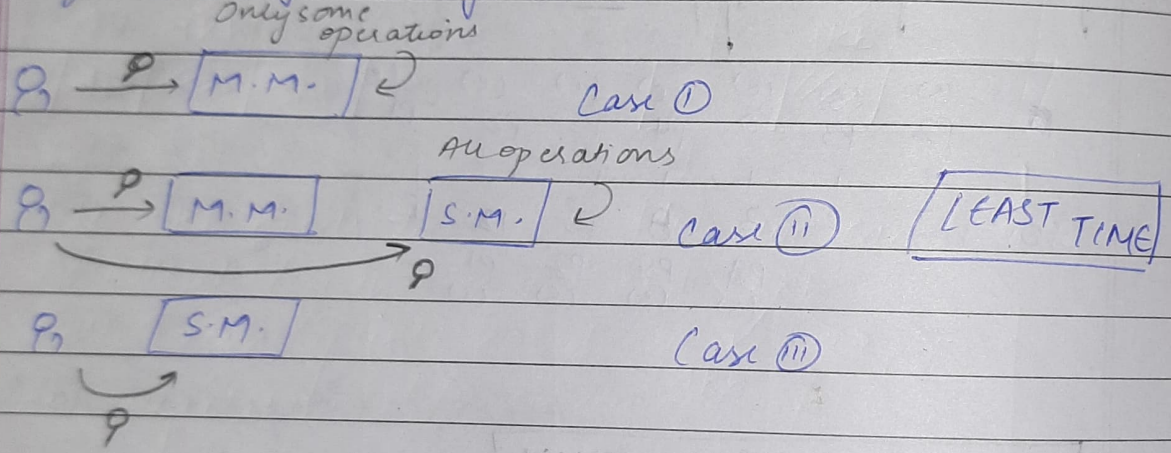
Memory Management

DATE: / /
PAGE:

In normal cases → memory needs to be large
→ access time needs to be less.

But in real time : If memory ↑ access time ↑

Access time from memory can be reduced by usage of secondary memory. How?



In real life, CPU has NO IDEA that main memory exists

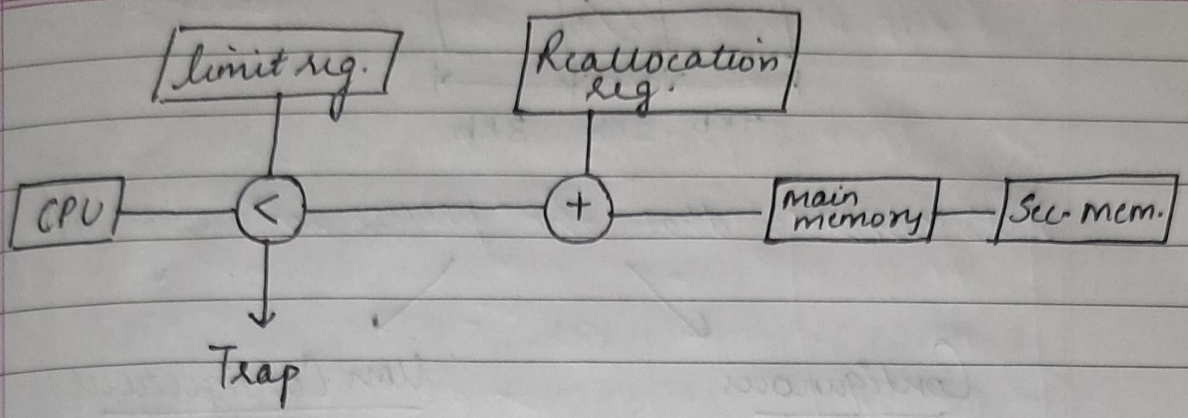
To access anything from Main memory
↓
Physical Add. is required

Logical Addr. is used by sec. memory
↑
But CPU makes use of logical address

So we need to find a way to convert logical add. to physical address or vice versa.

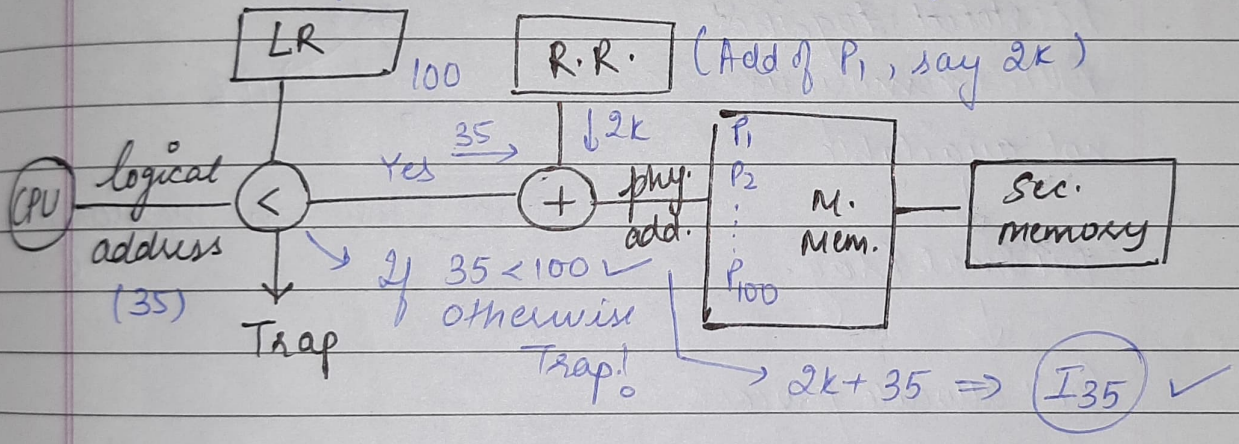
Logical Address to Physical Address

DATE: / /
PAGE:



① Reallocation Reg → store base add. of processes in main memory

② Limit Reg → keep track of no. of instructions.



Memory Allocation

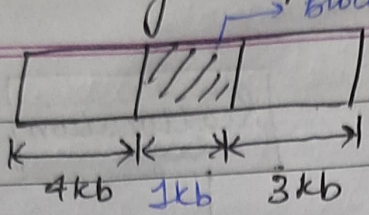
memory can be alloted using either → contiguous policy
→ non contiguous policy

Contiguous Policy → similar to array memory allocation
→ direct access available

Non Contiguous Policy → Similar to linked list
→ direct access NA.

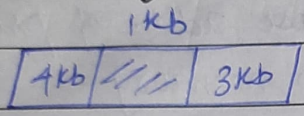
DATE: / /
PAGE:

Consider memory snapshot



Let's consider that the process incoming is of 5kb

Contiguous



Here 5kb needs to be stored together BUT memory is not available

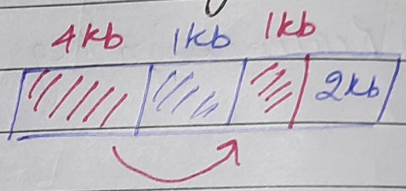
↓

External Frag.

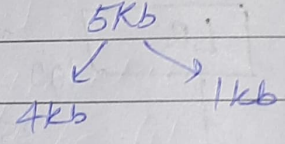
↓

Can't allocate process

Non Contiguous



Here we can break



and store in memory

Contiguous Memory Allocation

Variable side partioning (dynamic)

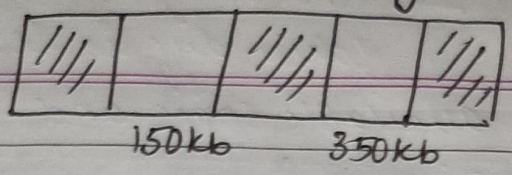
- first fit
- Best fit
- Worst fit

Fixed side partioning (static)

- first fit
- Best fit
- Worst fit

Consider a random memory screenshot

DATE: / /
PAGE:



Let's try to solve the problem for the processes mentioned below:

P1 (300Kb) P2 (25Kb) P3 (120Kb)
P4 (45Kb)

Use variable side partioning and compute result for first, best and worst fit.

① FIRST FIT → whenever a potential block is found, fit the process.

Part ① 150 Kb
Part ② 350 Kb

Process No.	Size	Part ①	Part ②
P1	300 Kb	×	✓ fit
P2	25Kb	✓ fit	
P3	120 Kb	✓ fit	
P4	45Kb	×	✓ fit

After P1 ⇒ Part ②: 50Kb
After P2 ⇒ Part ①: 125Kb
After P3 ⇒ Part ①: 5Kb
After P4 ⇒ Part ②: 5Kb

All process are fit!

(ii) BEST FIT → find the partition where diff is less

DATE: / /
PAGE:

(diff b/w available space and process size is minimum)

PART-① [150Kb] PART-② [350Kb]

Process No	Size	Part ①	Part ②
P1	300Kb	X	✓ fit!
P2	25Kb	diff 125 ✓	✓ 25 fit!
P3	120Kb	✓ fit!	X
P4	45Kb	X	X

After P1 ⇒ Part ② [50Kb]
After P2 ⇒ Part ② [25Kb]
After P3 ⇒ Part ① [30Kb]
After P4 ⇒ External fragmentation (process can't be fit).

(iii) WORST FIT

Part ① 150Kb Part ② 350Kb → opposite of best fit

Process No.	Size	Part ①	Part ②
P1	300Kb	X	✓
P2	25Kb |diff|	✓ 125 fit!	✓ 25
P3	120Kb	✓ fit!	X
P4	45Kb	X	✓ fit

After P1 ⇒ Part ② 50Kb
After P2 ⇒ Part ① 125Kb
After P3 ⇒ Part ① 5Kb
After P4 ⇒ Part ② 5Kb

All process are fit!

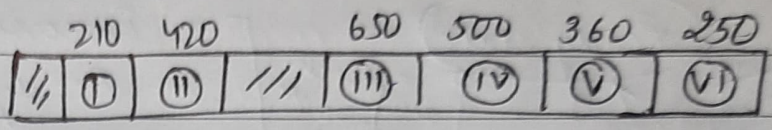
(##) In variable side partioning
↳ Best fit performs WORST!
↳ Worst fit performs BEST!

DATE: / /
PAGE:

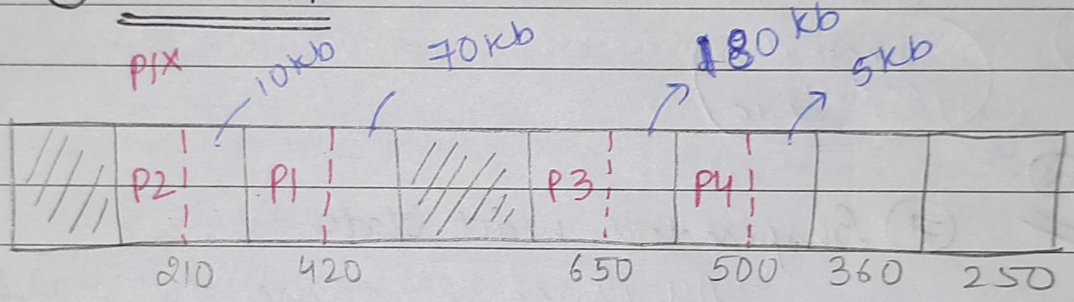
Fixed Side Partioning

↳ partioning is pre defined and only ONE process can go in one block.

P1	350Kb
P2	200Kb
P3	470Kb
P4	495Kb

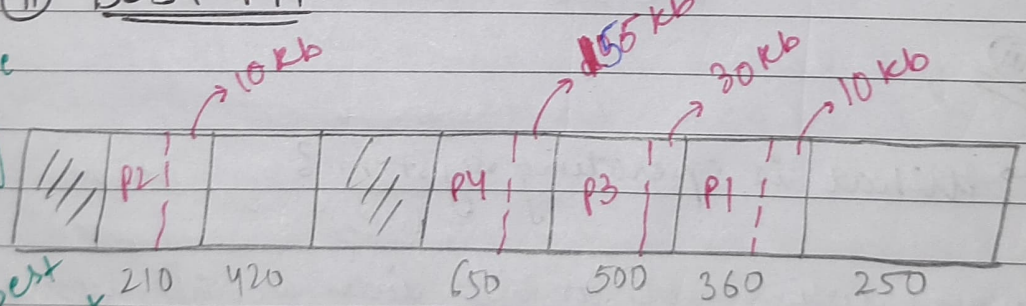


(I) FIRST FIT



Total Waste = 10 + 70 + 180 + 5 + 360 + 250
= 80 + 185 = 265Kb

(II) BEST FIT



Total Waste = 10 + 155 + 40 = 205Kb

In fixed side partioning
↓
Best fit works best
Worst fit works worst

(III) WORST FIT

Total Waste = 300 + 300 + X + X
External Frag!