COMPUTER SCIENCE

Memory Management

Lecture No 01

Dr. KHALEEL KHAN SIR

Abstract View of memory

Loading vs linking

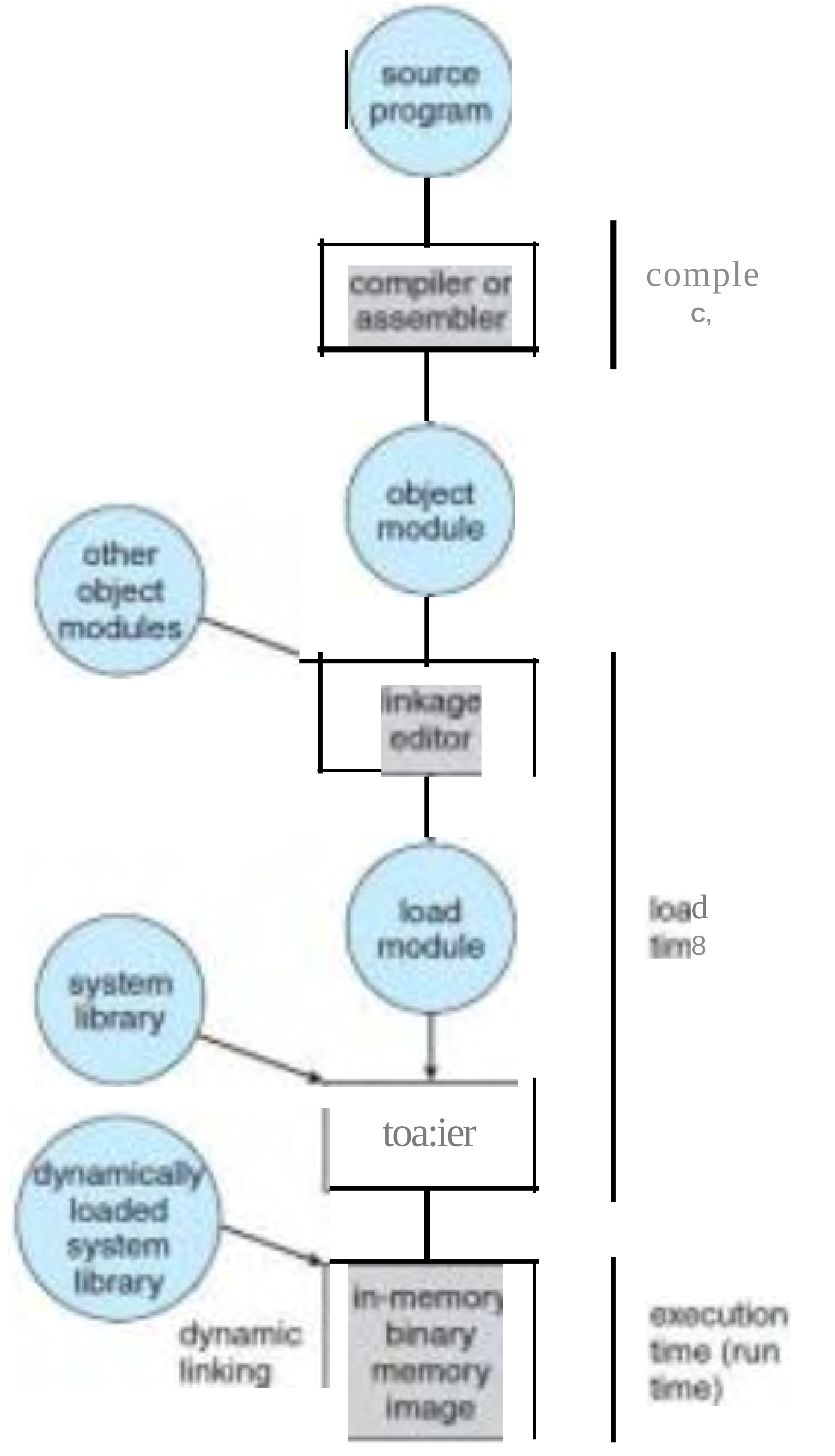
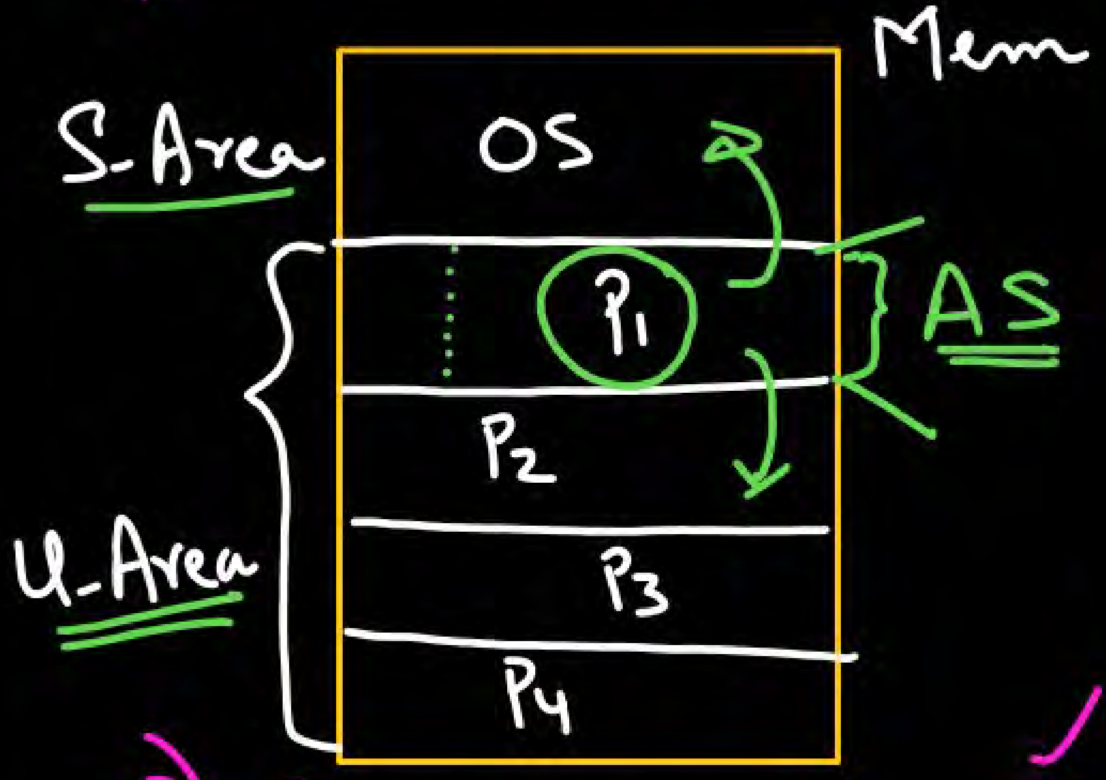


Figure 8.3 Multistep processing of a user program

Functions & Goals of Memory Manager

1) Allocation ✓
2) Protection ✓



3) Free Space Mgmnt. ✓
4) Address translation ✓
5) Deallocation –

1) Effective utilization of Memory (No wastage)
↳ Fragmentation
Internal External

2) Manage the execution of larger Programs in smaller Memory area;
Prog-Size = 100KB; Avail. Mem: 60KB
VM; overlays

Mem. Mgmnt. Techniques

Contiguous Allocation (CA)

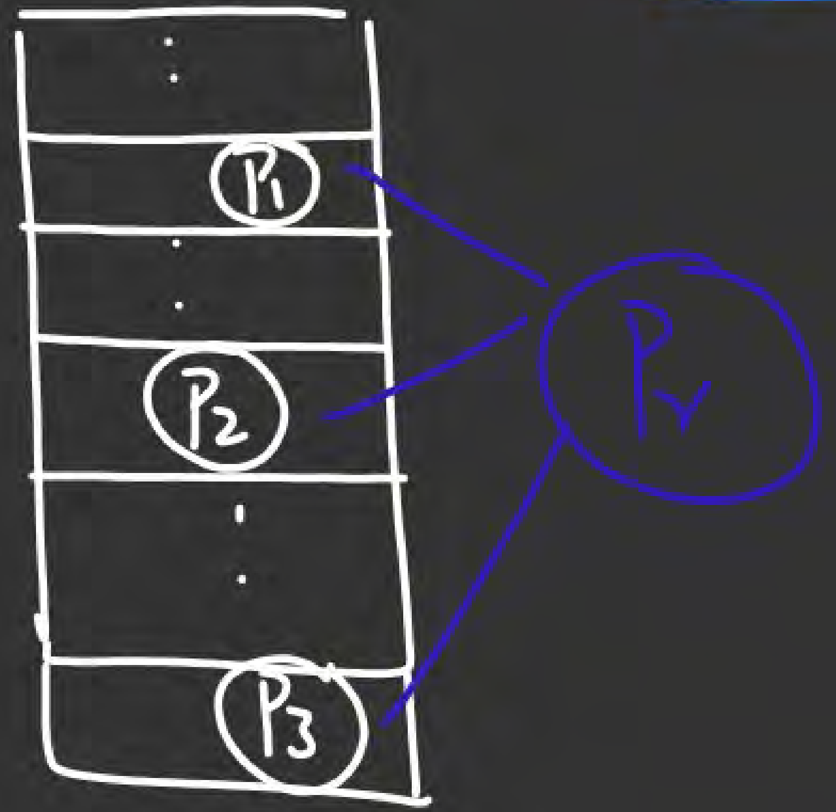
1) overlays;
2) Partitions; < fixed, variable.
3) Buddy System

<Centralized>

Non-Contiguous (NCA)

Paging;
Segmentation;
Seg-Paging;
VM

Distributed



① overlays : (replace)

2-pass Assembler

✓ Pass 1 : 70 KB

✓ Pass 2 : 80 KB

✓ Symbol Table : 30 KB

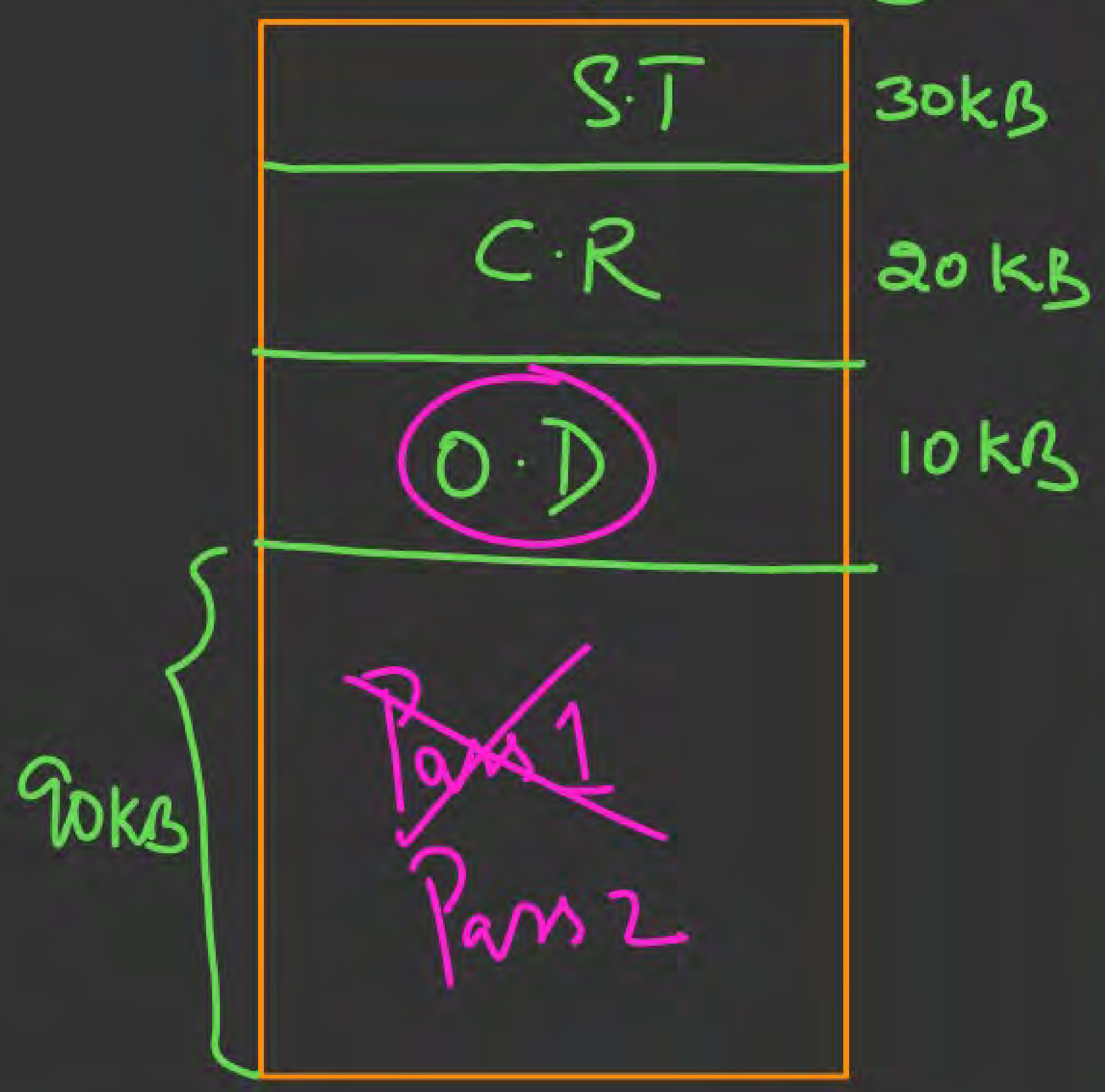
✓ Common Rts : 20 KB

✓ overlay Loader / driver : 10 KB

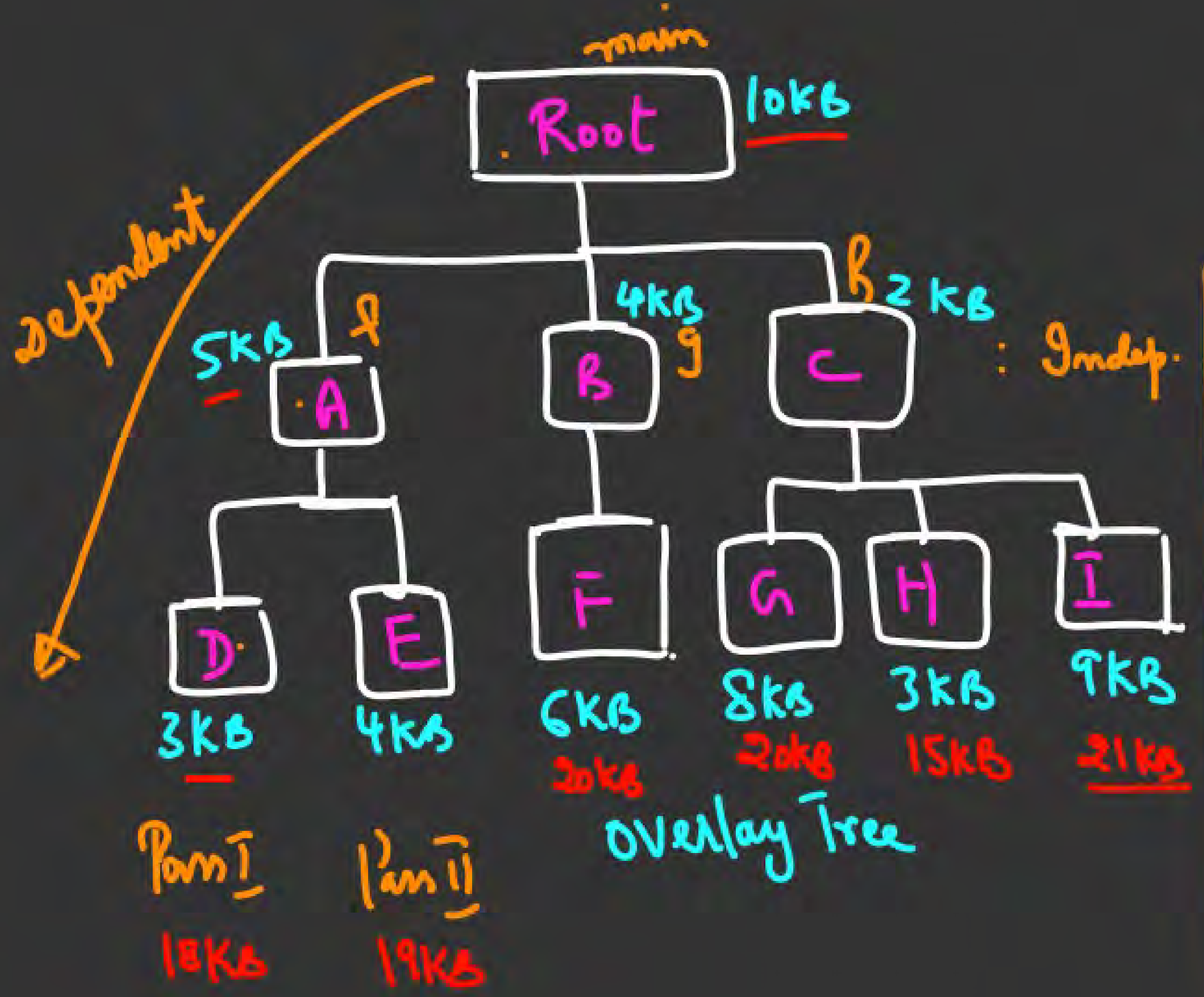
210KB

(Managed) 210KB → 150KB

M.M (150KB)



overlaying is possible only when Program can be divided into Independent Modules;



Total Program Size : 54KB

What is the minimum amount of Memory, Sufficient to execute 54KB Program using overlays?

21KB

Min. Mem Required = Max{ Path_length from Root to Leaf }

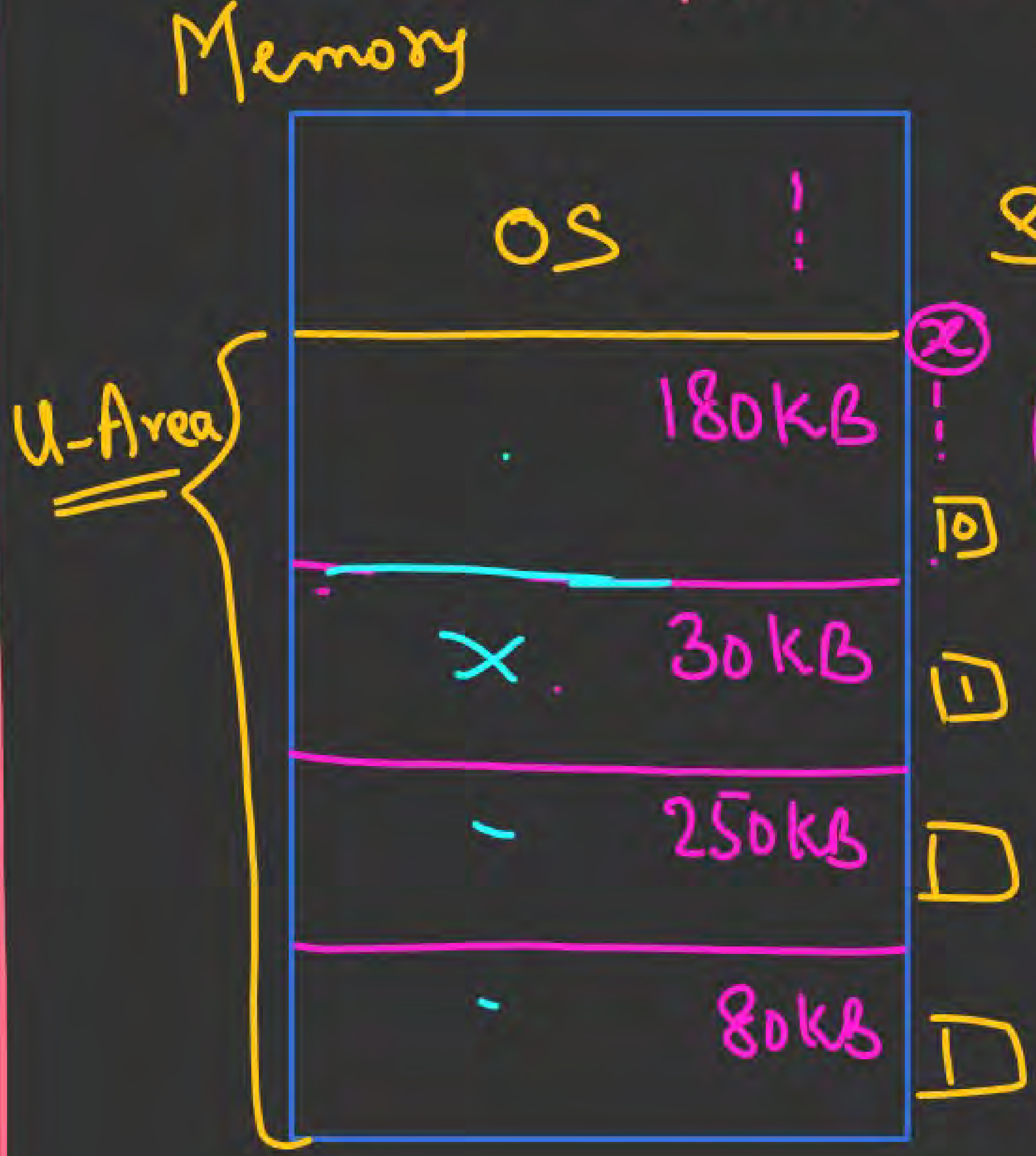
Primitive

II. PARTITIONING

- Fixed Partition (MFT) (Static approach)
- Variable Partitions (MVT) (Dynamic Approach)

(i) Fixed Partitions

MFT: Multiprogramming with Fixed No. of Task;



S_Area

1-partition = 1-program

Limit Registers

P_{new} : 50KB

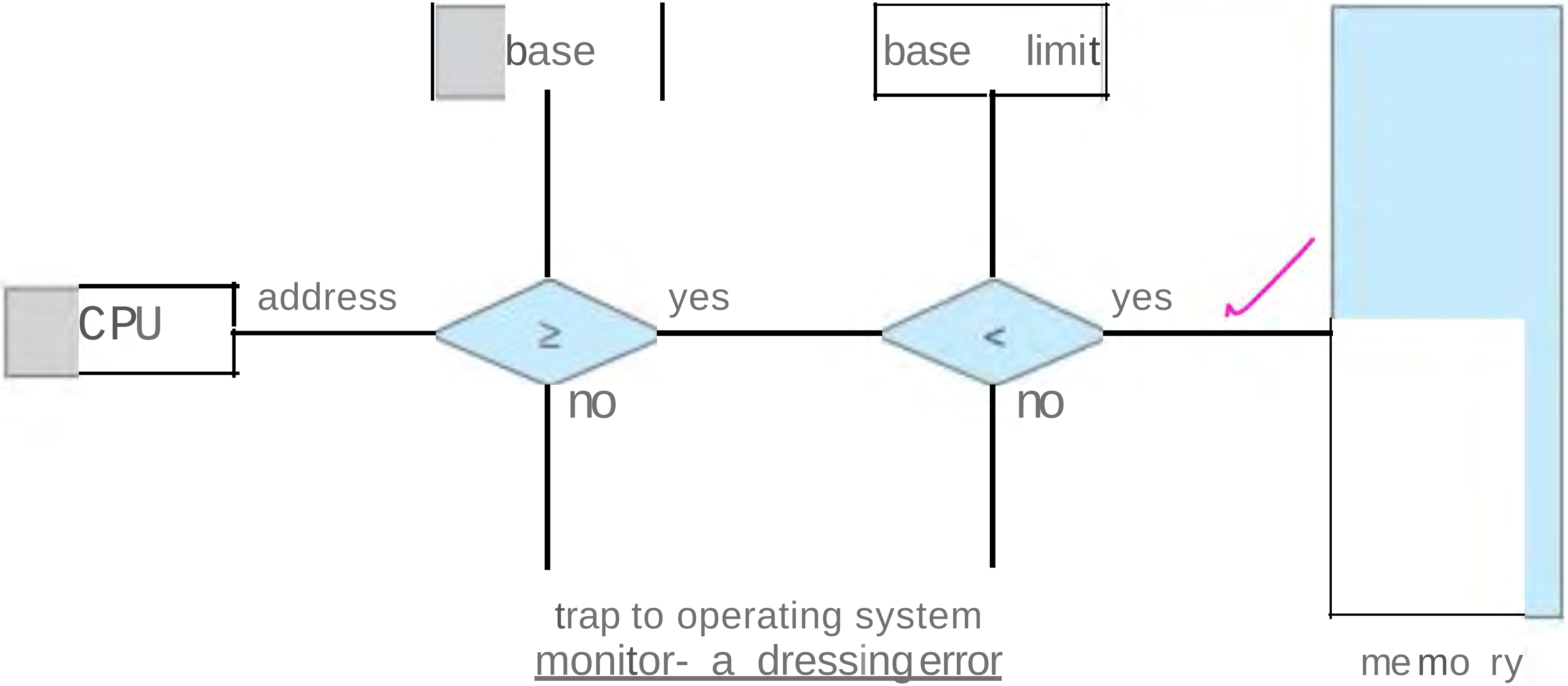


Figure 8.2 Hard\vare adcJress protectio.n ith base and limit registers.

Allocation: In which partition are we going to allocate the Incoming Process Request;

Internal fragmentation

a) First Fit (FF)

"First free big enough"

b) Best Fit: (BF)

"Smallest free big enough"

c) Worst Fit (WF)

"Largest free big enough"

d) Next Fit (NF):

works like FF, except that Search for free partition Commences from Last Allocation;

~~Pr: 20KB~~

Pr: 45KB

"Next Fit May work faster than First Fit;"

Performance:

1. Internal Fragmentation: ✓ (IF)
2. External Frag.: X
3. Degree of M.Pr: Limited
4. Max. Process Size: Limited
5. Alloc. Policy: Best Fit <less IF>

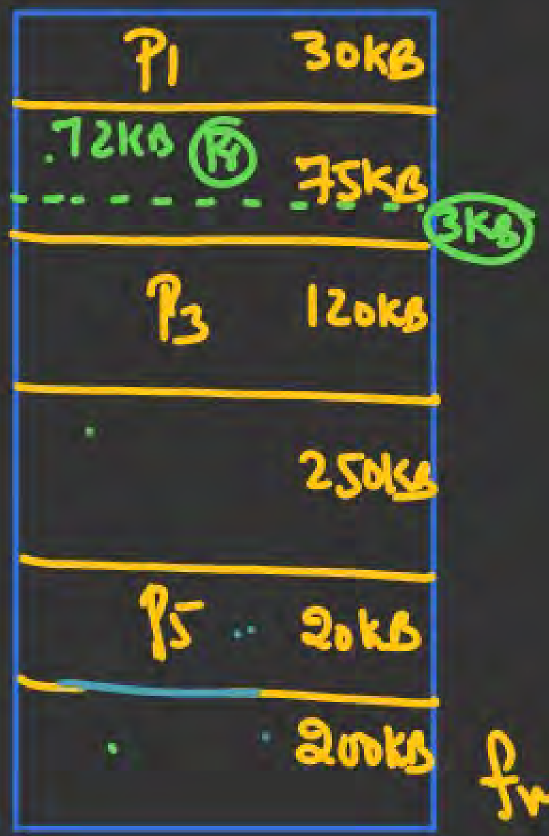
II. Variable Partitions

Dynamic

MVT: M.Pr. with variable Tasks;

} Partitions will be created on demand @ R.T

't': Req's: 30KB; 75KB; 120KB; 250KB; 20KB;



free Hole

CPU

Pnew: 72KB

FF BF NF WF

Two adjacent free holes are Merged to a single hole.

Coalescing

Total free Space: 220K

Pr: 110KB

Pnew: 48KB

Performance Issues:

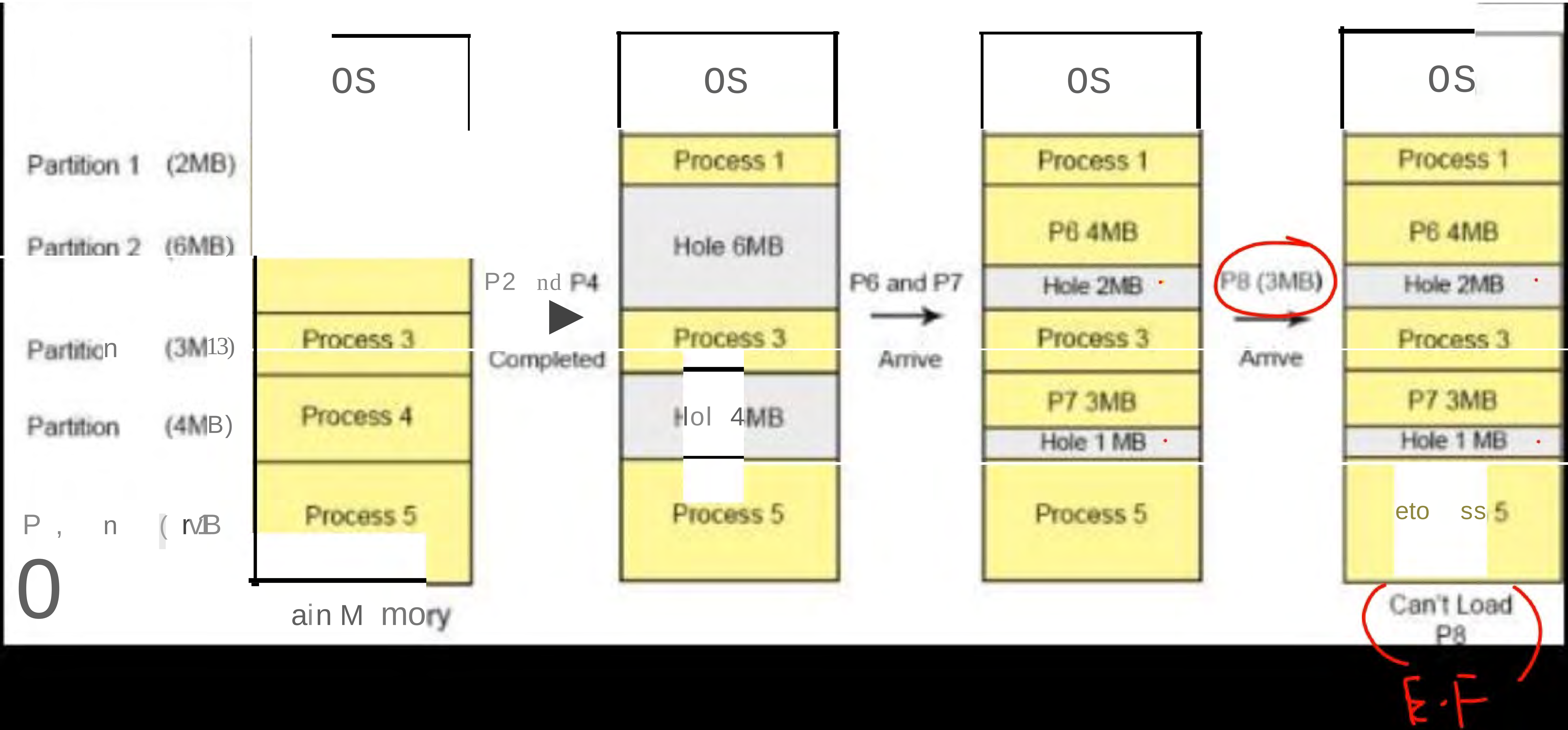
1) I.F: X

2) Ext-Frag: ✓ (May have)

3) Degree of M.Pr: Flexible

4) Max. Process Size: Flexible

5) Alloc. Policy: Worst Fit



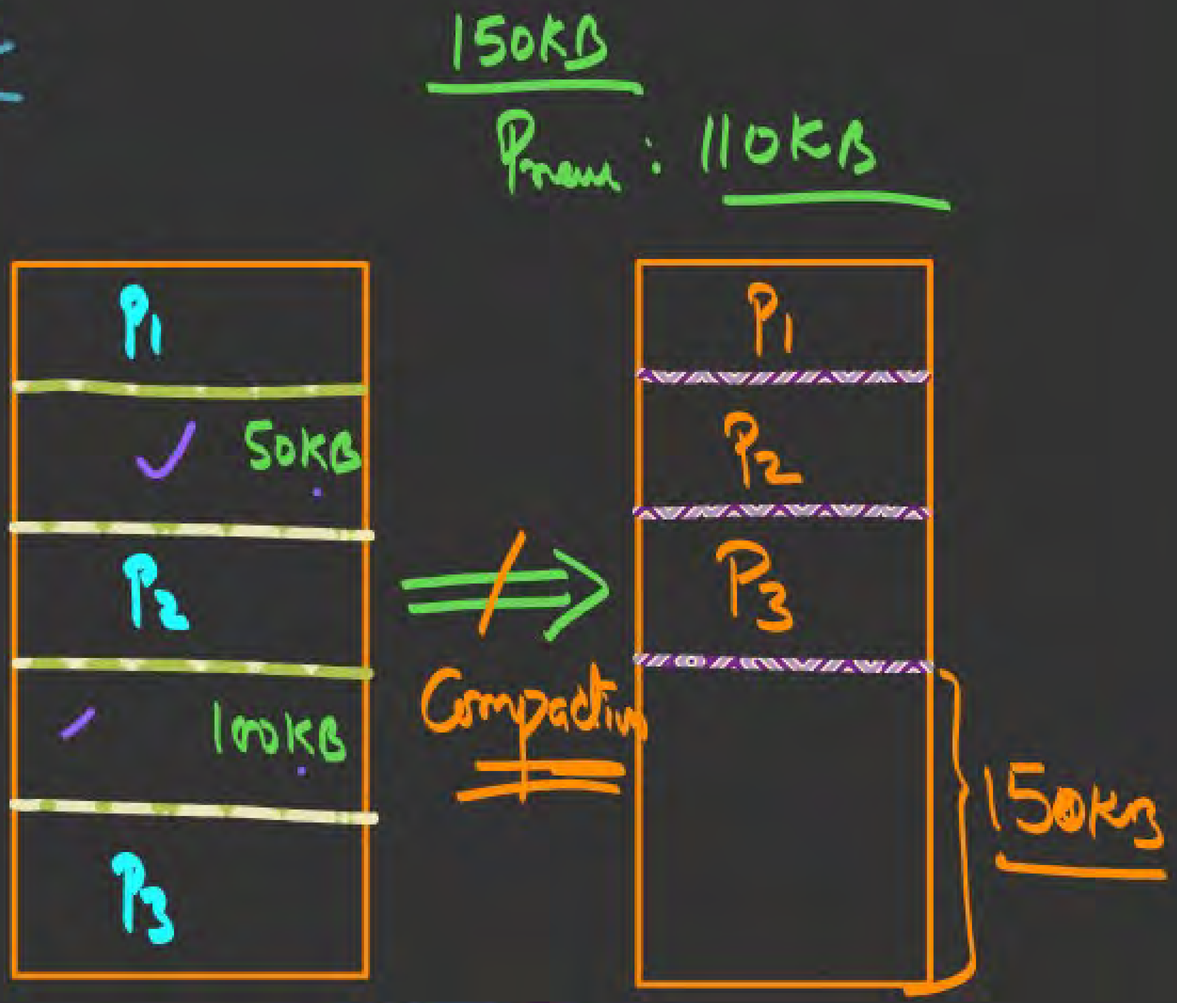
Partition 1 (2MB)
Partition 2 (6MB)
Partition (3MB)
Partition (4MB)
OS
Process 3
Process 4
Process 5
P2 nd P4
Completed
OS
Process 1
Hole 6MB
Process 3
Hole 4MB
Process 5
P6 and P7
Arrive
OS
Process 1
P6 4MB
Hole 2MB
Process 3
P7 3MB
Hole 1 MB
Process 5
P8 (3MB)
Arrive
OS
Process 1
P6 4MB
Hole 2MB
Process 3
P7 3MB
Hole 1 MB
Process 5
Can't Load P8
E.F

External Fragmentation

Compaction

Merging of Non-CG free holes, by pushing (relocating) all Processes to one end;

(Non-CG Alloc)



Pnew : 110KB

50

60

E.F

"PAGING"

Drawback of Compaction:

(i) Time Consuming operation (overhead)

ii) Compaction is possible with only R.T Addrs. Binding

Q.1 Consider a Memory System having 6 Partitions of sizes 200K; 400K; 600K; 500K; 300K; 250K. There are 4 Processes of sizes: 357K; 210K;468K; 49K. Using **Best Fit Allocation Policy,** what Partitions are not allocated/ remains Unallocated?

Q.2 Consider the following Memory Map in which blank regions are *not in use* and hatched regions are in use. Using Variable Partitions with no Compaction:

The sequence of requests for blocks of sizes 300K, 25K, 125K, 50K can be satisfied if we use:

(A) Either first fit or best fit policy (any one)

(B) First fit but not best fit policy ✓

(C) Best fit but not first fit policy

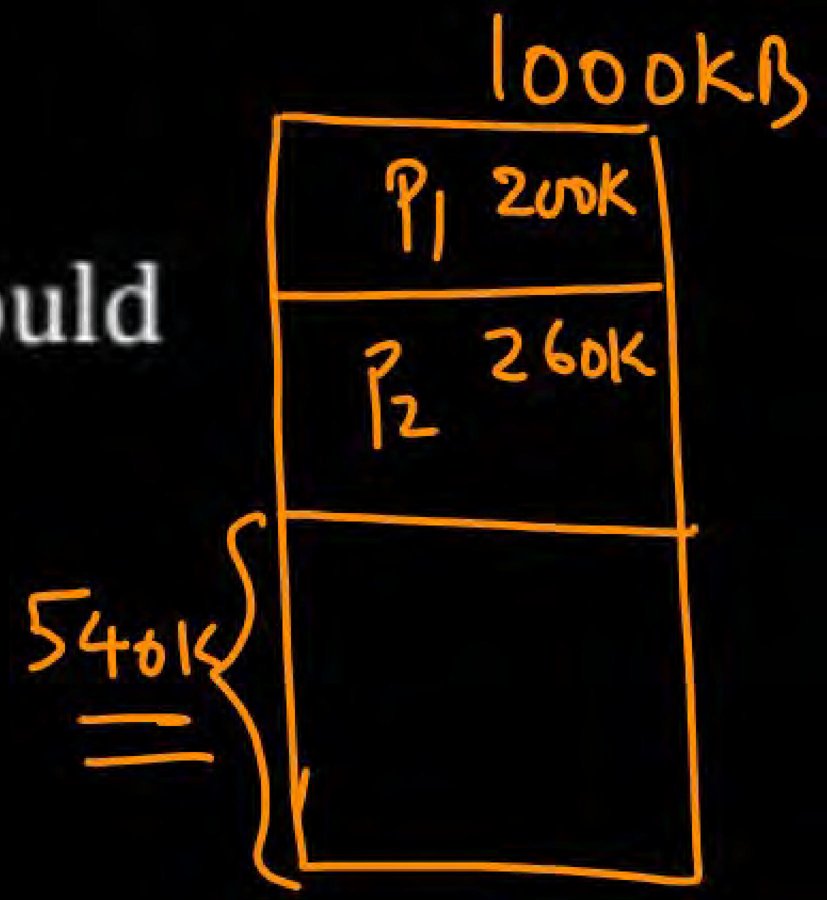
(D) None of the above.

F.F : ✓

B.F : X

Q.3

Consider a System with Memory of size 1000KBytes. It uses Variable Partitions with no Compaction. Presently there are 2 partitions of sizes 200K & 260K respectively.



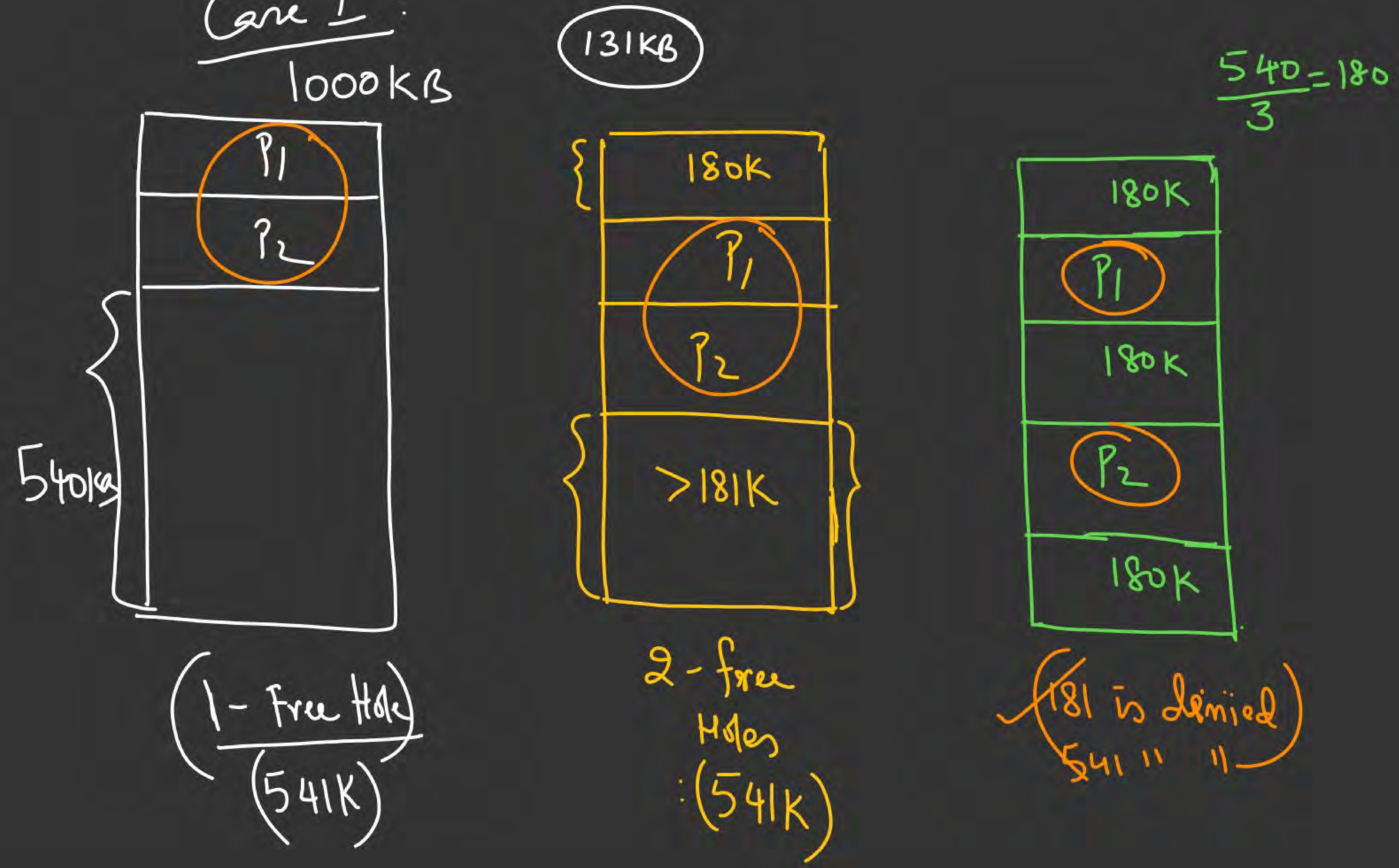
(i) What is the allocation request of the Process which would always be denied?

Ⓐ 131 K Ⓑ 151 K Ⓒ 181K Ⓓ 541K

(ii) The smallest Allocation Request which could be denied is:

Ⓐ 131 K Ⓑ 151 K Ⓒ 181K Ⓓ 541K

Case I :


1000KB
P1
P2
540KB
1 - Free Hole
(541K)
131KB
180K
P1
P2
>181K
2 - free Holes
: (541K)
540/3 = 180
180K
P1
180K
P2
180K
181 is denied
541 " "

Q.4 Consider a System having Memory of size 2^{46} Bytes, uses Fixed Partitioning. It is divided into fixed size Partitions each of size 2^{24} Bytes. The OS maintains a Process Table with one entry per Process. Each entry has, two fields: First, is a pointer pointing to Partition in which the Process is loaded and Second, Field is Process ID(PID). The Size of PID is 4Bytes.

Calculate

(a) The Size of Pointer to the nearest Byte. $\Rightarrow$ 3 Bytes

(b) Size of Process Table in Bytes if the System has 500 Processes.

No. of Part's $= \dfrac{2^{46}}{2^{24}} = 2^{22} = 4M$

Mem $= 2^{46}$

Part. size $= 2^{24}$

Partition Address (Ptr) = 22 bits = (3 Bytes)

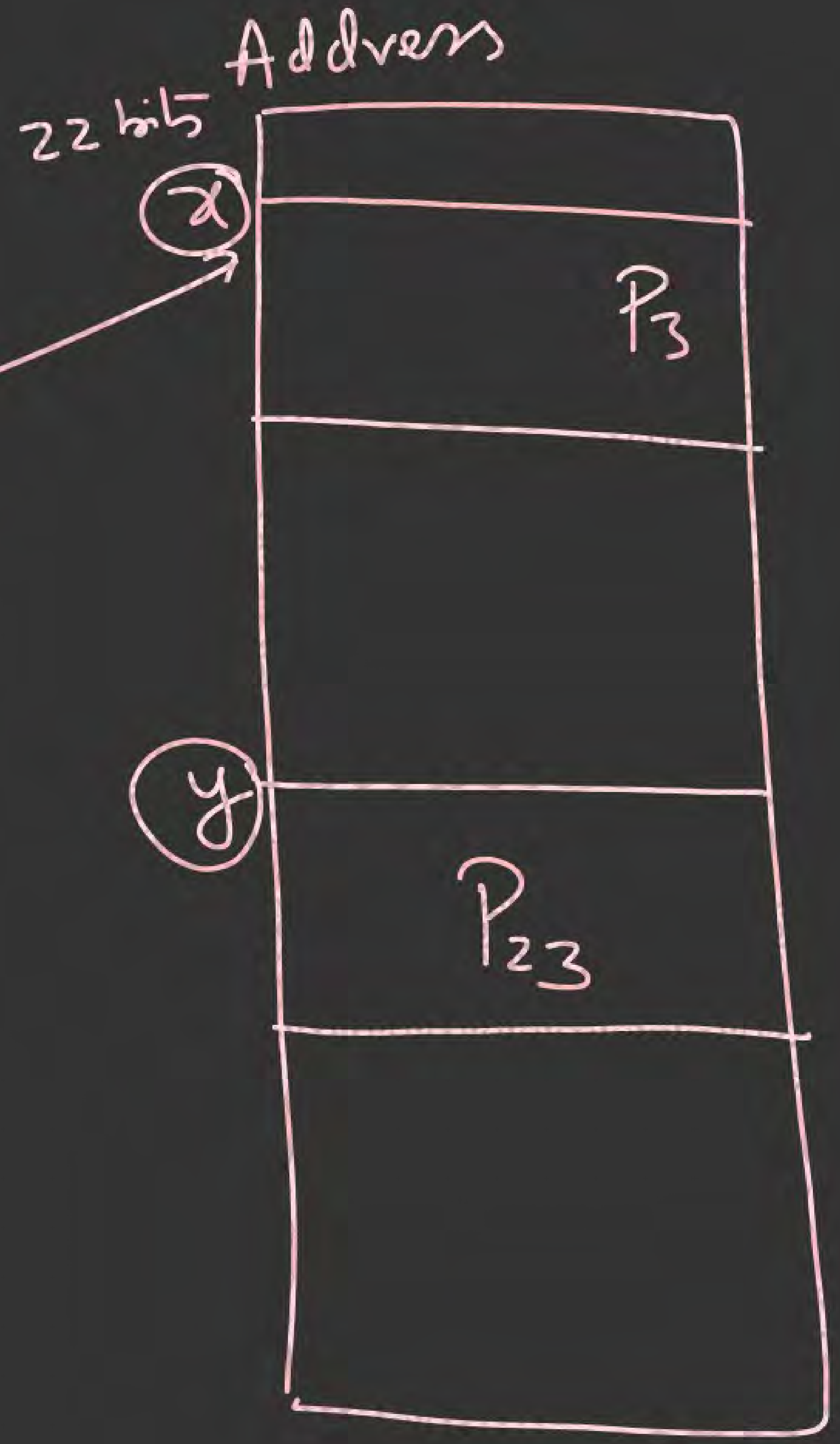
Pid (4B)	Ptr (3B)
3	x
23	y

500

Process Table

$= 500 \times 7B$

$= 3500 \text{ Bytes} = \underline{3.5\ KB}$



Q.5 Consider a System Using Variable Partition with no Compaction

11:50 am

Free l1oles	4K; BK; 201<; 2K
Program size	2K; 14K; 3K; 6K; 10K; 20K; 2K
Time for Execution	4; 10; 2; 1; 4; 1; 8

B.Ts

Using Best Fit Allocation Policy and FCFS CPU Scheduling Technique, Find the Time of Loading & Time of Completion of each program. The Burst Times are in Seconds.

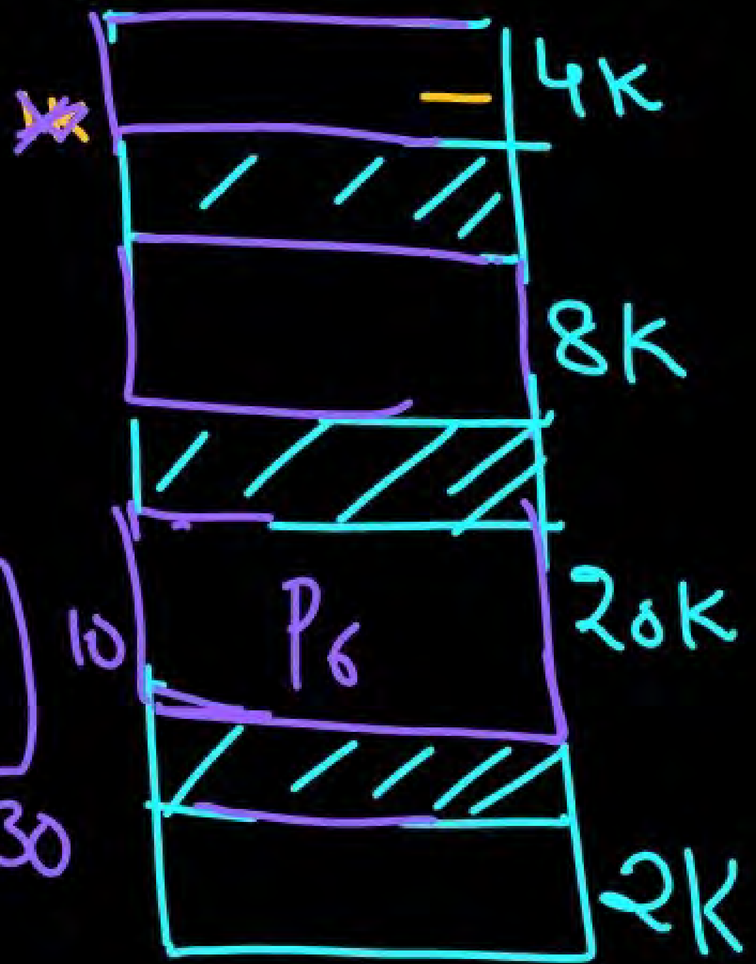
t_0: P_1, P_2, P_3, P_4, P_7

t_{14}: P_5

t_{29}: P_6

CPU

CPU	P_1	P_2	P_3	P_4	P_7	P_5	P_6
0	4	14	16	17	25	29	30



THANK YOU!