CS & IT ENGINEERING

Operating Systems

Memory Management

Lecture No. 2

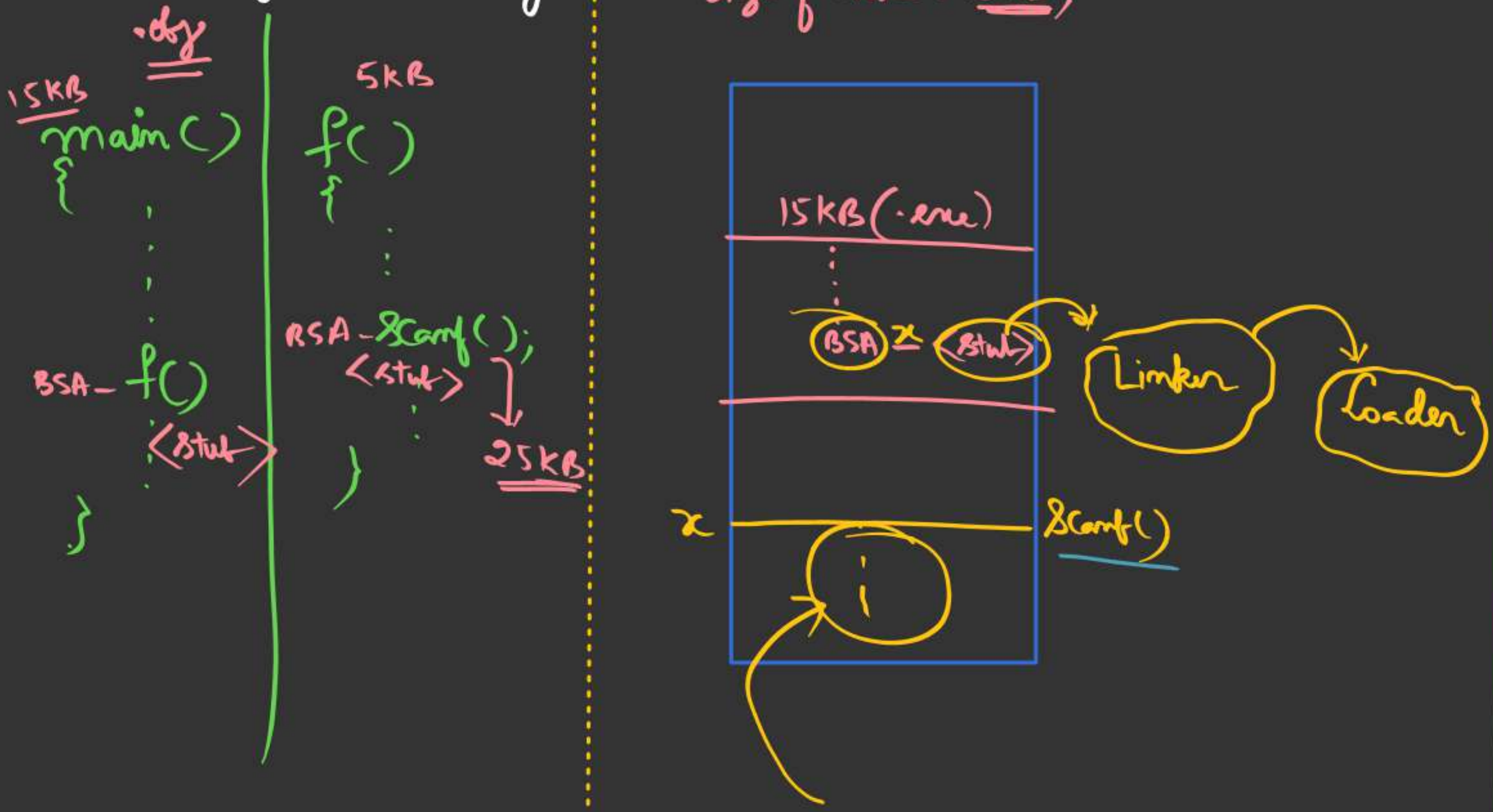
By- Dr. Khaleel Khan Sir

TOPICS TO BE COVERED

- Linking
- Address Binding
- Contiguous Allocation

Dynamic Linking :

Size of .exe : 15KB;



Benefits :

(i) Space Efficiency

(ii) Reusability

(iii) Flexibility :

Drawbacks :

→ Time inefficiency

→ Security Threat

Address Binding : Association of attributes to entities;

(Binding Time)

→ Compile Time

→ Load Time

→ Run Time

```
int x;

x = 5;
```

entity : Variable

→ Name

→ Type

→ Size (2B)

} C.T

→ Address

→ Value

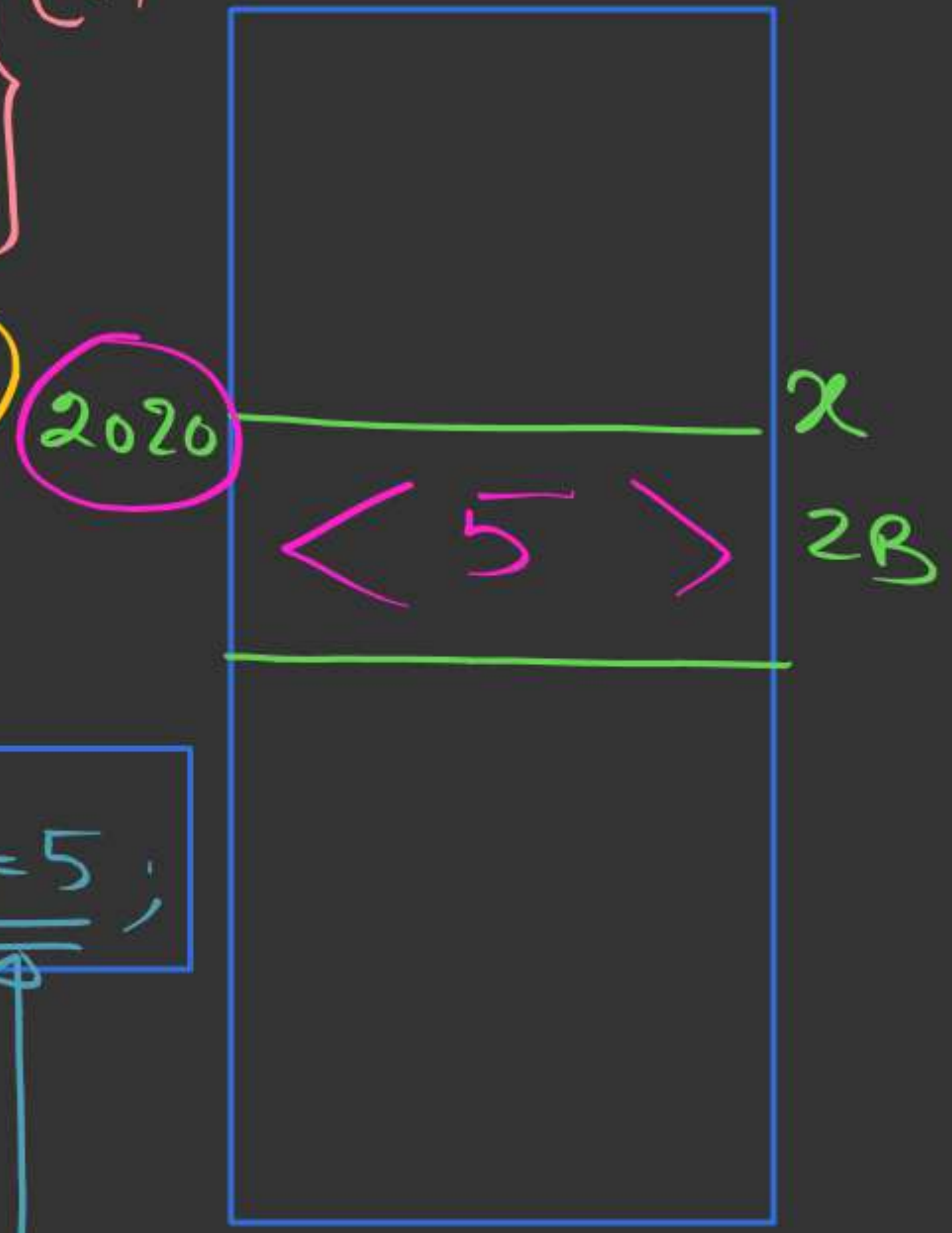
R.T

(Instn)

Store x, #5

```
int x=5;
```

Load Time



From O.S perspective :

Address Binding : Assoc. of Instrs/Data units of the Program to memory Locations (Addresses)

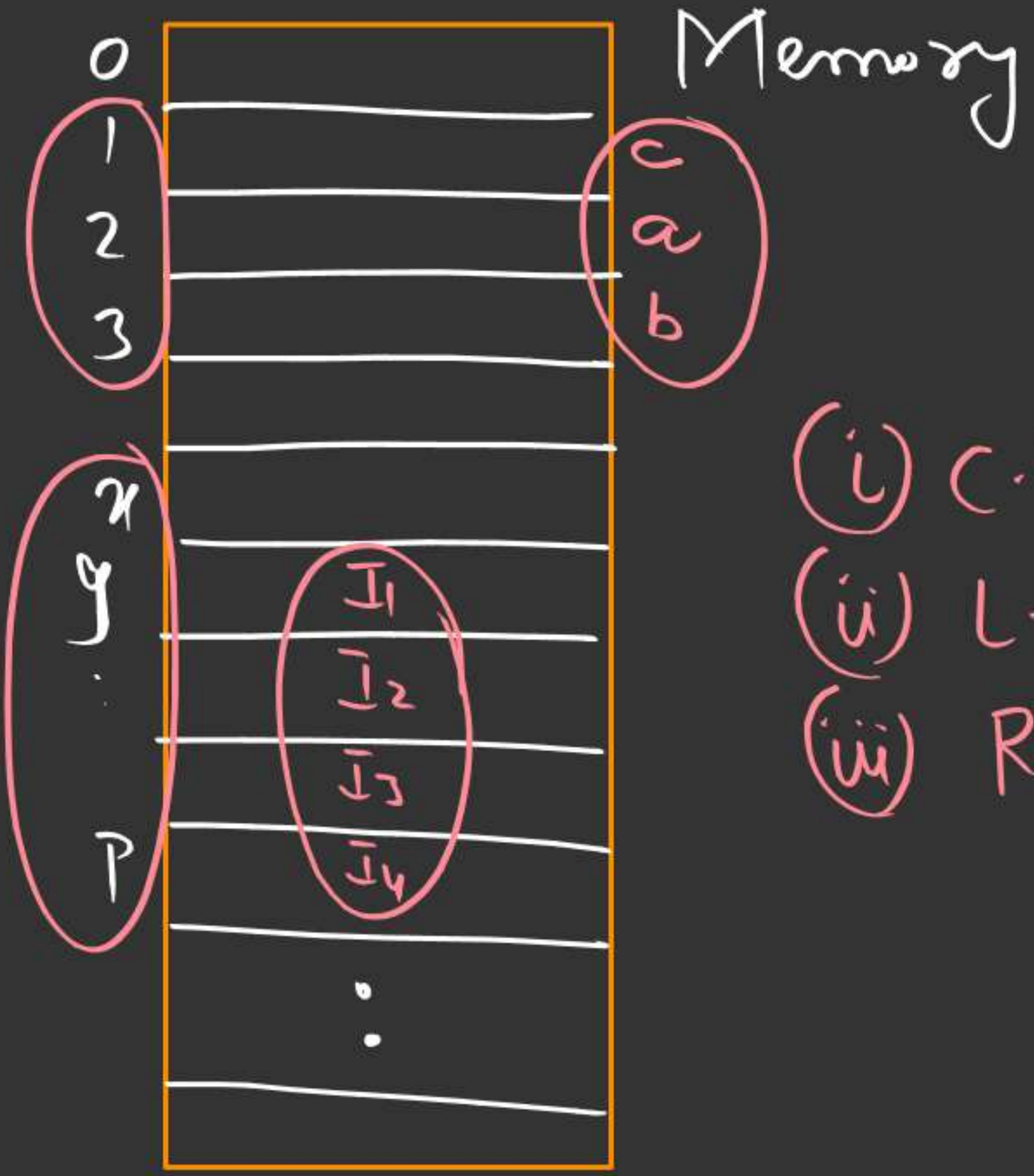
Data

```
int a, b, c;

c = a+b;
```

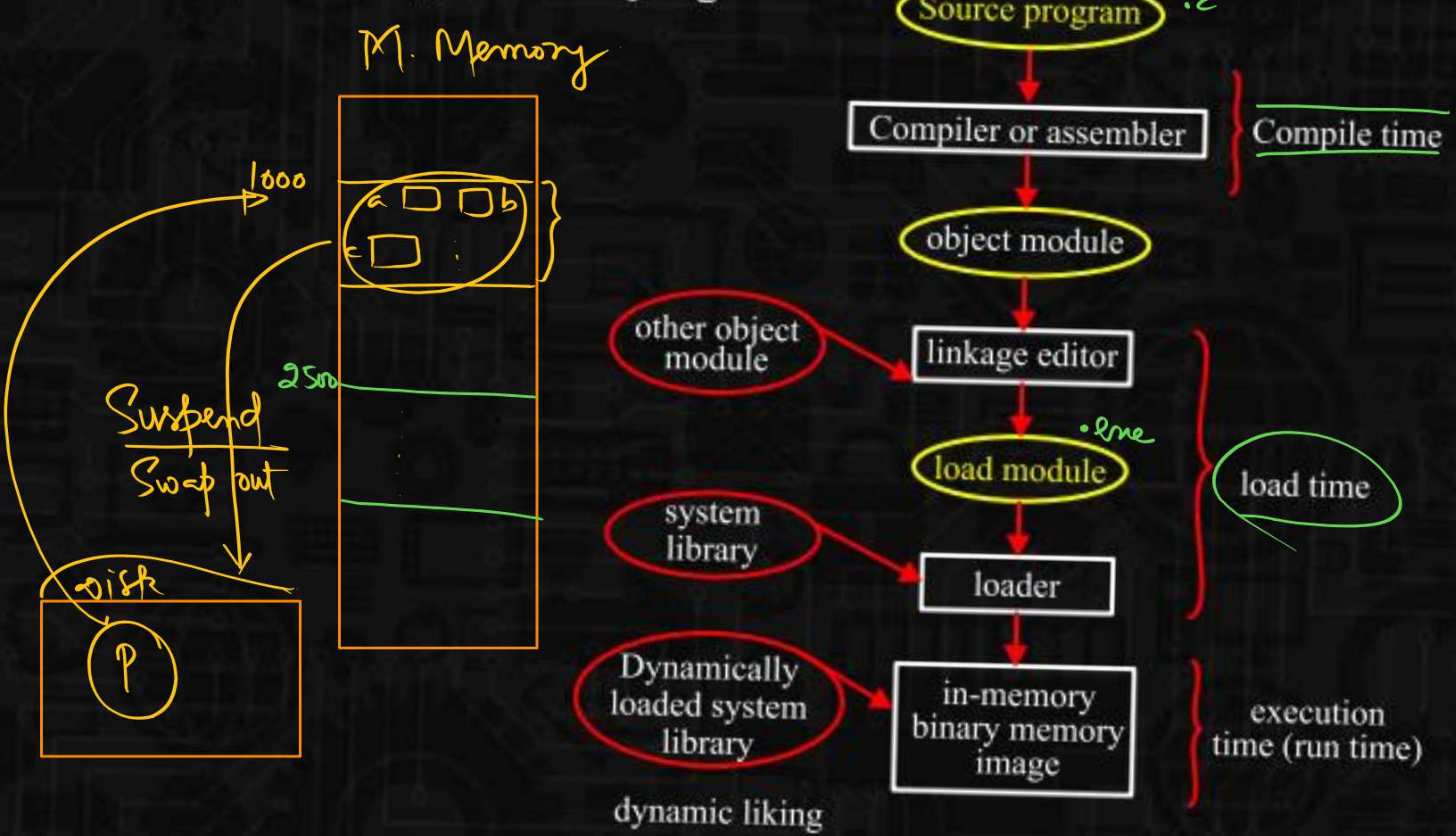
Instr's

```
I1 : Load R1, a
I2 : Load R2, b
I3 : Add R1, R2
I4 : Store c, R1
```



(i) C.T } STATIC
(ii) L.T } STATIC
(iii) R.T } dynamic

Multistep processing of a user program.



Source program

Compiler or assembler

Compile time

object module

other object module

linkage editor

load module

load time

system library

loader

Dynamically loaded system library

in-memory binary memory image

execution time (run time)

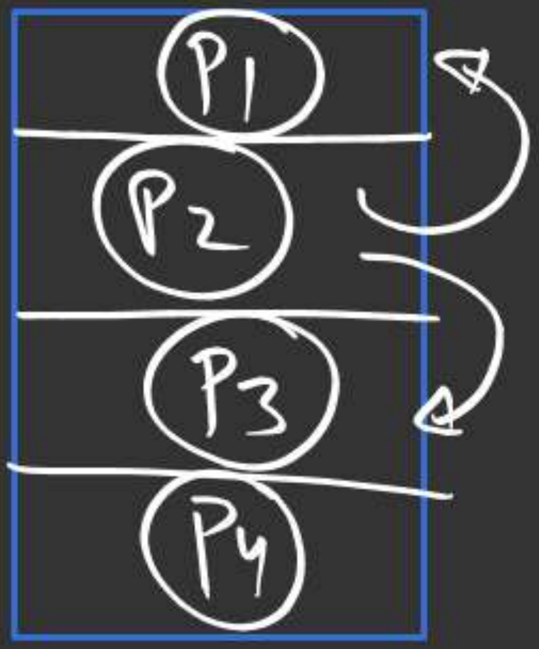
dynamic liking

Memory Mgmnt. Techniques

Functions & Goals

Functions

1) Allocation
2) protection

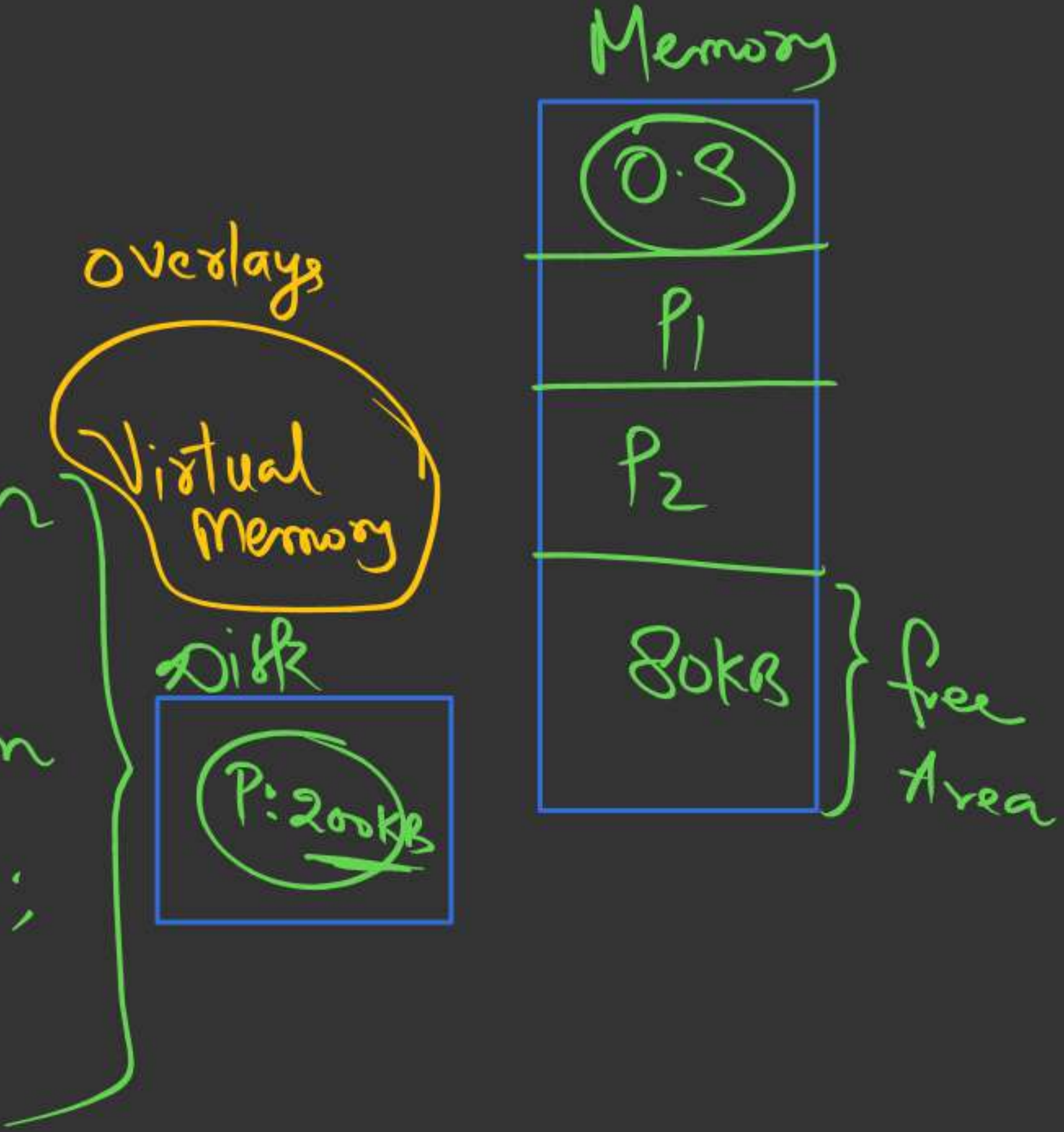


3) Free Space Mgmnt.
4) Deallocation

Goals

→ Efficient utilization of Memory;
< Fragmentation >

→ Ability of Mem Manager to manage execution of a larger program in small memory areas;

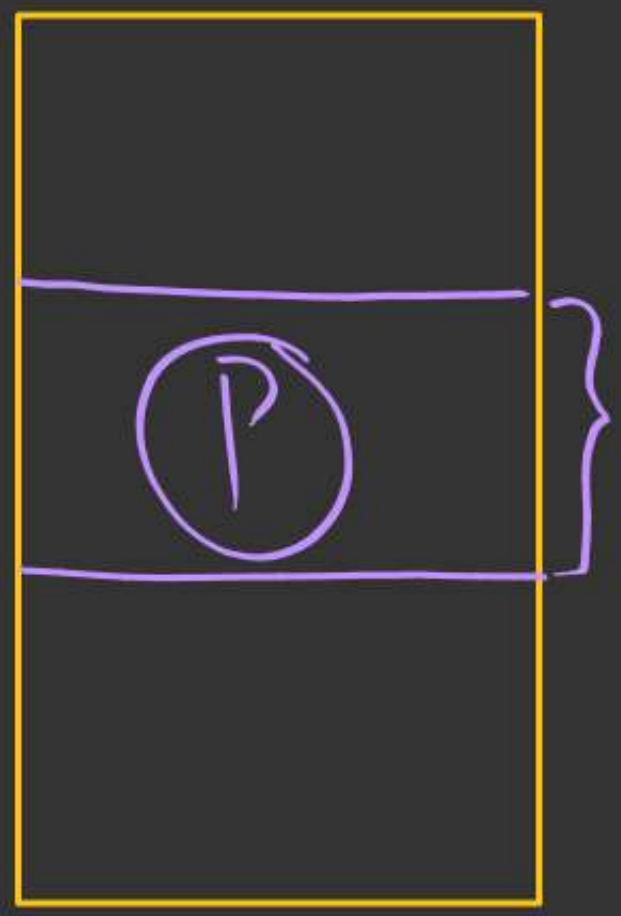


Mem. Mgmnt Techniques

Contiguous (CG)

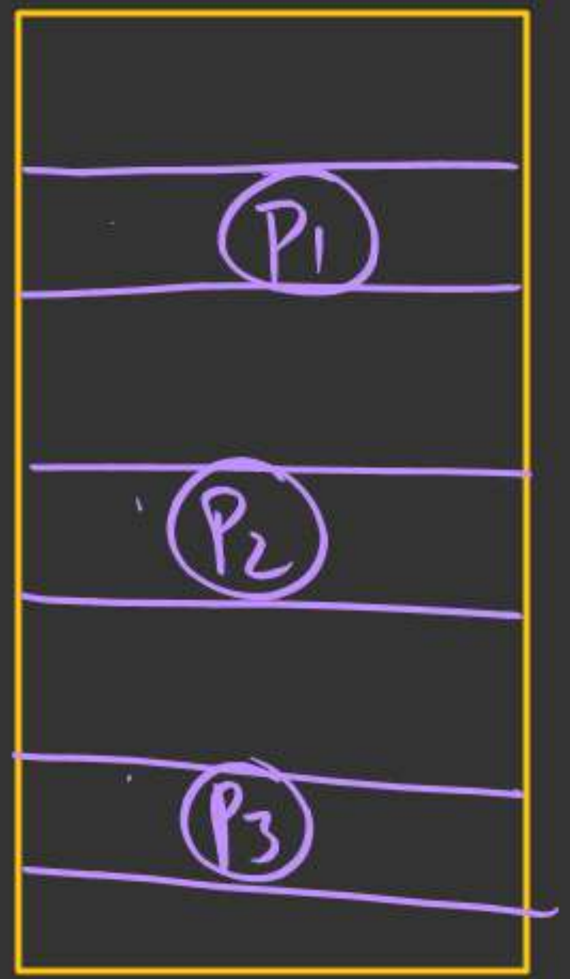
(Centralized)

- (i) Overlays
- (ii) Partitions



Non-Contiguous (NCG)

<Distributed>



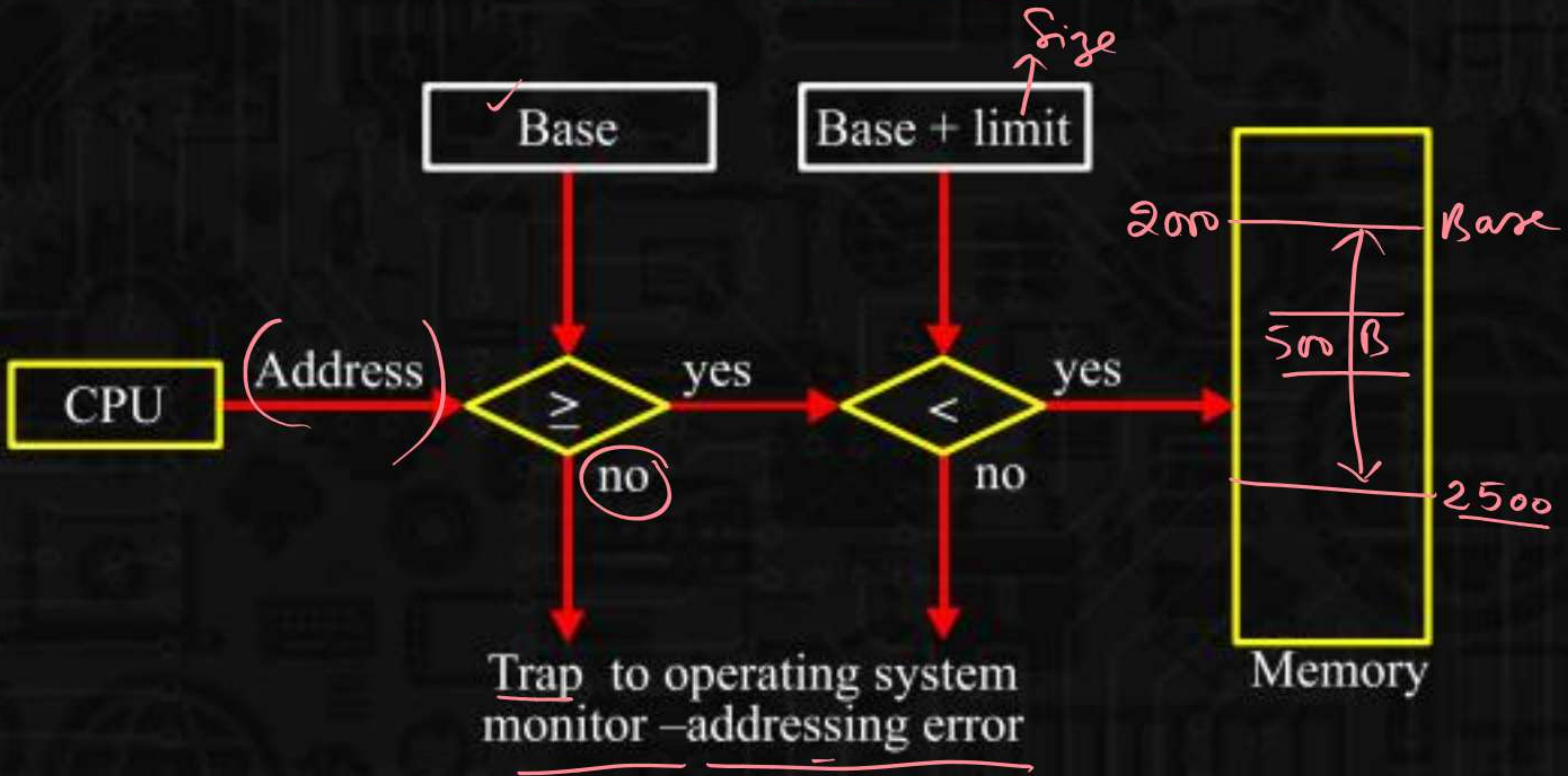
(i) Paging

(ii) Segmentation

(iii) Segmented

(iv) V.M

Protection



Size

Base

Base + limit

CPU

Address

≥

yes

no

<

yes

no

Trap to operating system monitor –addressing error

2000

Base

500 B

2500

Memory

1) OVERLAYS: → (replace)

2 Pass Assembler:

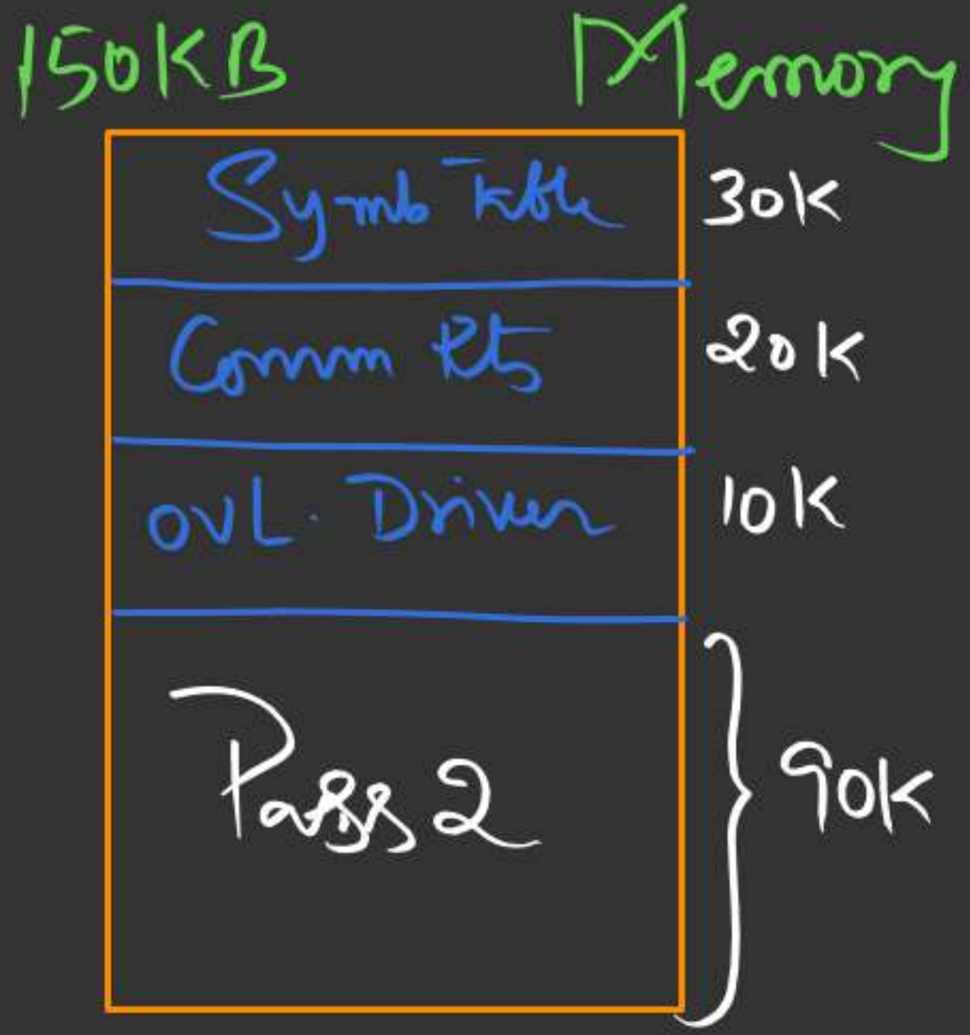
Pass 1 : 70 KB
Pass 2 : 80 KB } Mut. Exclusive

Symbol Table : 30 KB

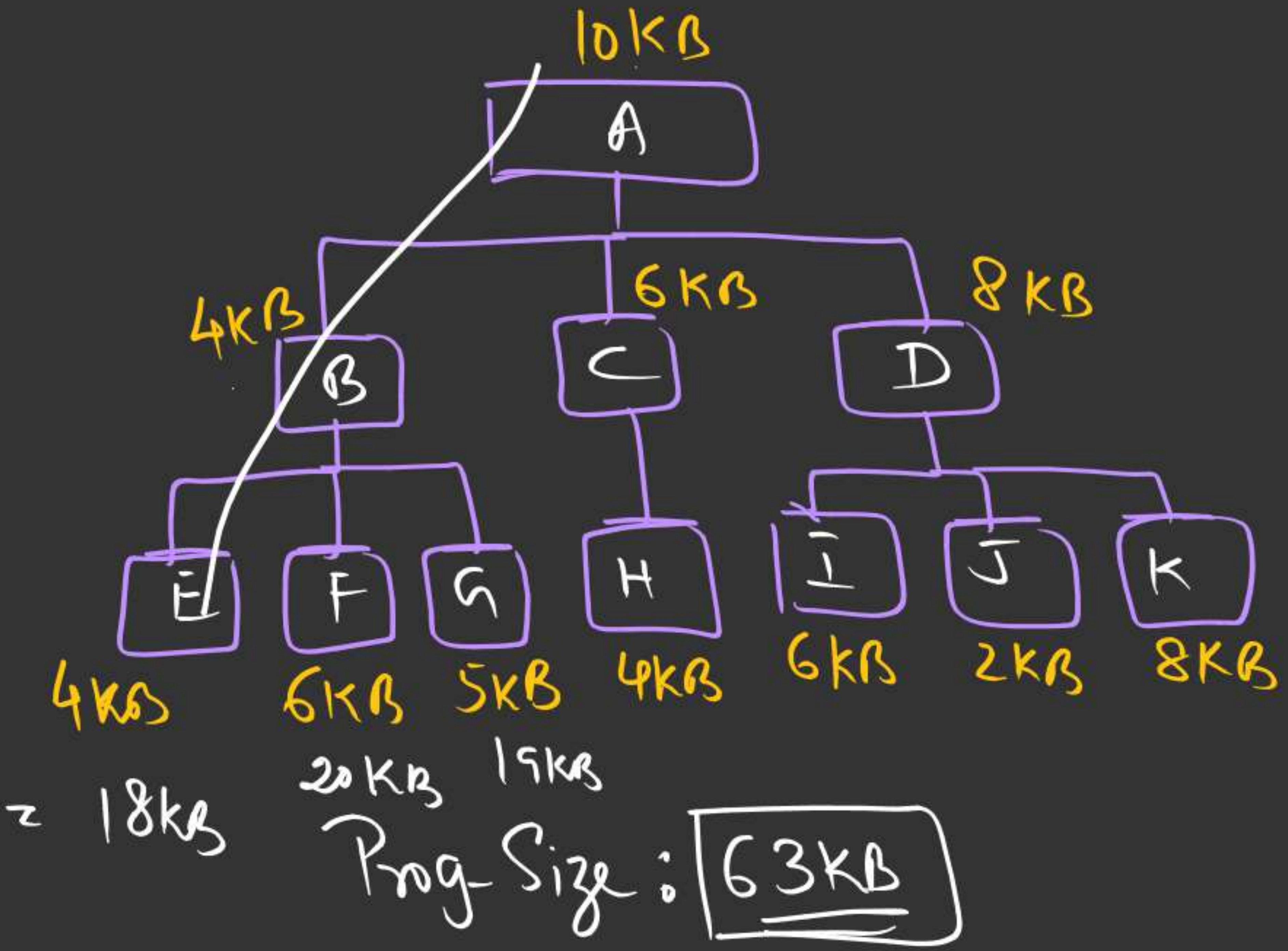
Common Rts : 20 KB

overlay Loader : 10 KB

(210 KB)



Overlay Tree of a Program



Prog Size : $\boxed{63KB}$

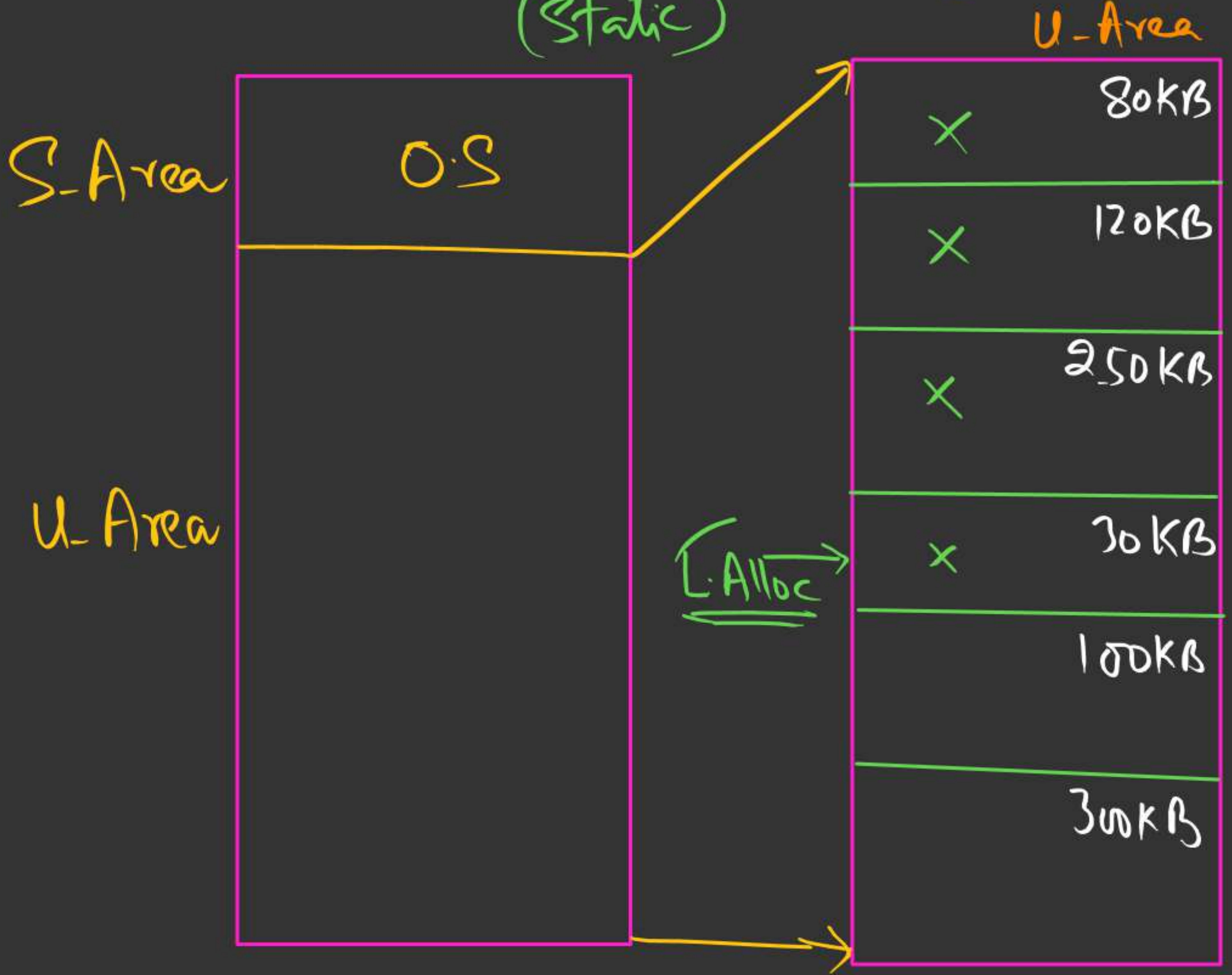
Min. Mem Reqd to execute This Program (overlays) = Max { Path Lengths from root to Leaf }

= 26KB

II: PARTITIONS (CG)

a) Fixed Partitions (M.F.T = M.Pr with Fixed Tasks)
(Static)

1 Partition = 1 Process



Allocation:

Prow: (50KB)

Partition Alloc. Policies:

1) First Fit: First free big enough
2) Best Fit: Smallest Free big enough
3) Worst Fit: Largest Free
4) Next Fit: Search for free big enough Partition commences from Last Alloc. (Faster)

Performance :

1) Internal Fragmentation: ✓

2) External Fragmentation: ✗

3) degree of M.Pr : Limited

4) Max. Process Size : Limited

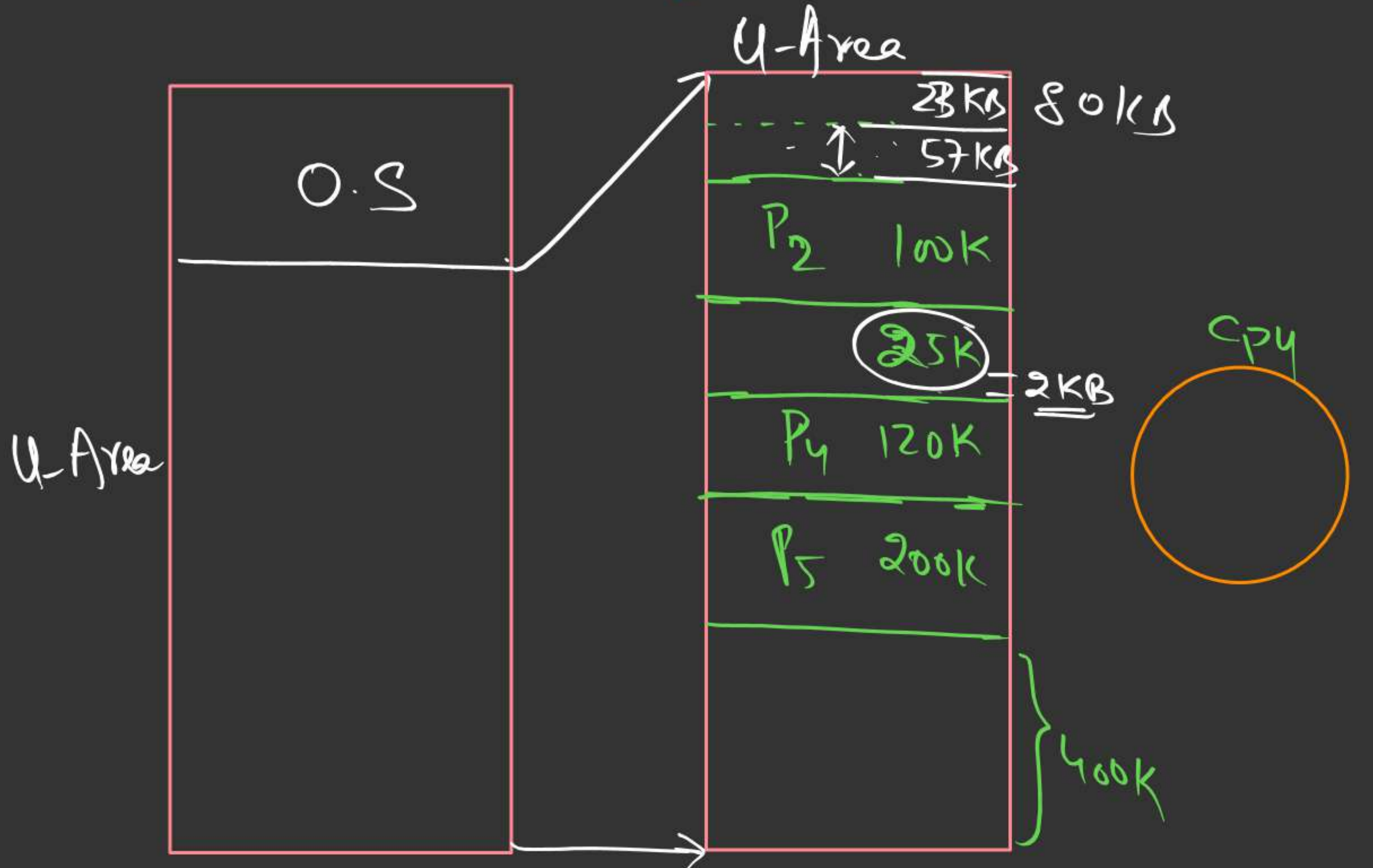
5) Alloc. Policy : Best Fit

2) Variable Partitions (Dynamic Partitioning): (M.V.T)

Job-Q / IP-Q (Disk

t

P1	P2	P3	P4	P5 ...
80K	100K	25K	120K	200K



t' : P_{new} : 23KB

FF

BF : Small free holes

WF :

NF

Performance :

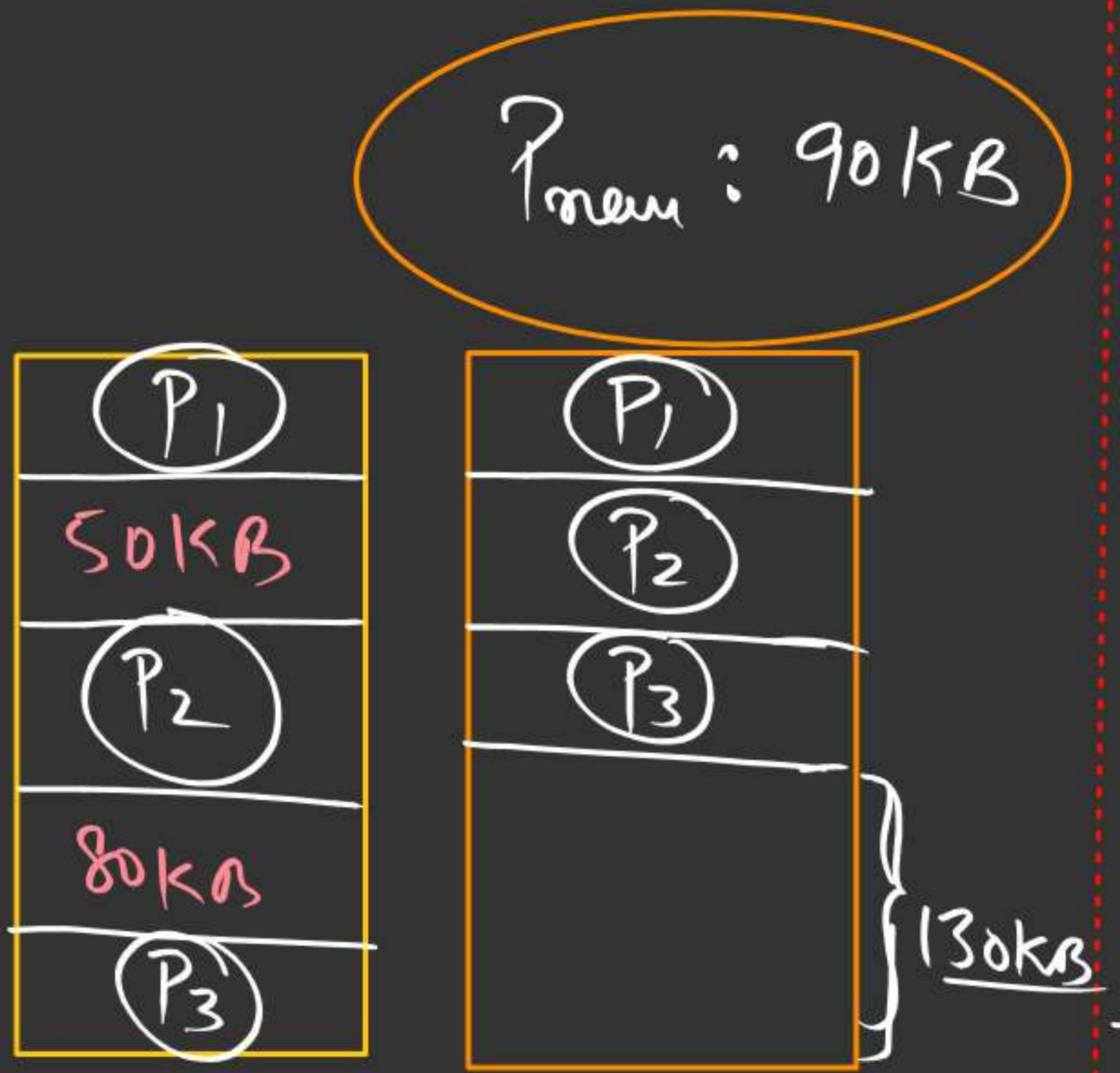
1) Int. Frag : ×

2) Ext. Frag : ✓

3) Degree of M.Pr : Flexible

4) Max. Process Size : Flexible

5) Alloc. Policy : (worst fit)



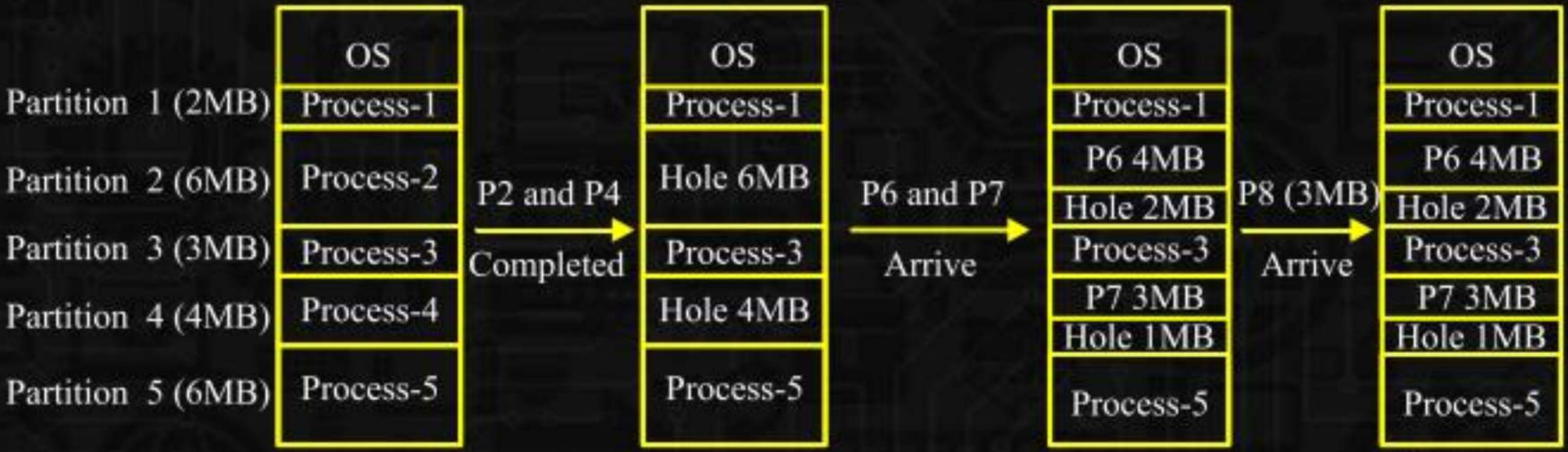
Free Space : 13 3

Ext. Fragmentation

Compaction
→ Time Consuming
→ R.T
Addrs. Binding

N.C.G Alloc

Paging



Partition 1 (2MB)

Partition 2 (6MB)

Partition 3 (3MB)

Partition 4 (4MB)

Partition 5 (6MB)

OS

Process-1

Process-2

Process-3

Process-4

Process-5

P2 and 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 1MB

Process-5

P8 (3MB)

Arrive

OS

Process-1

P6 4MB

Hole 2MB

Process-3

P7 3MB

Hole 1MB

Process-5

Main Memory

Can't Load
P8

Q.

The capacity of a memory unit is defined by the number of words multiplied by the number of bits/word. How many separate address and data lines are needed for a memory of 4K x 16?

(a) 10 address, 16 data lines

(b) 11 address, 8 data lines

[✓ (c) 12 address, 16 data lines]

(d) 12 address, 12 data lines

Q.

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?

357 → 400

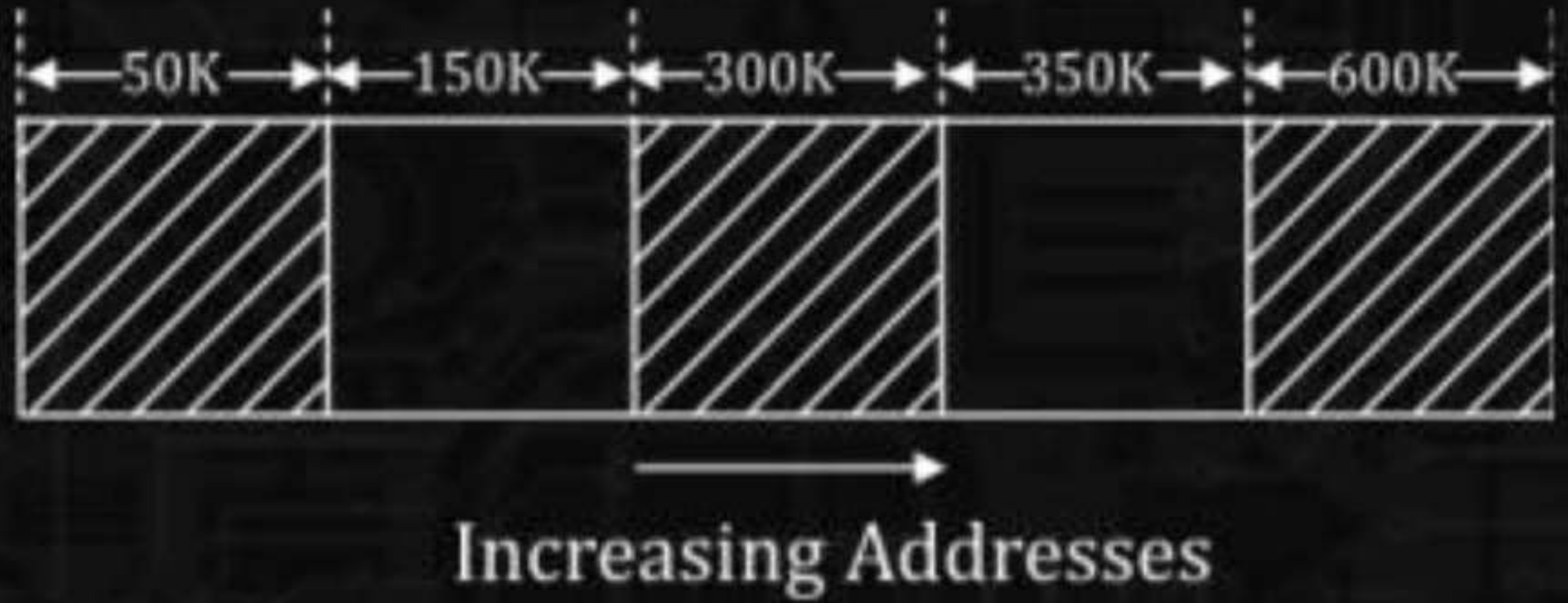
210 → 250

468 → 500

49 → 200

Q.

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.

Q.

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?

A. 131 K

B. 151 K

C. 181 K

D. 541 K

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

A. 131 K

B. 151 K

C. 181 K

D. 541 K

Q.

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.

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

Q.

Consider a System Using Variable Partition with no Compaction

Free holes	4K; 8K; 20K; 2K
Program size	2K; 14K; 3K; 6K; 10K; 20K; 2K
Time for Execution	4; 10; 2; 1; 4; 1; 8

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.

Q. Consider allocation of memory to a new process. Assume that none of the existing holes in the memory will exactly fit the process's memory requirement. Hence, a new hole of smaller size will be created if allocation is made in any of the existing holes. Which one of the following statements is TRUE?

A. The hole created by next fit is never larger than the hole created by best fit

B. The hole created by worst fit is always larger than the hole created by first fit

C. The hole created by first fit is always larger than the hole created by next fit

D. The hole created by best fit is never larger than the hole created by first fit

THANK YOU GW SOLDIERS !