CS & IT ENGINEERING

Operating Systems

Memory Management

Lecture No. 3

By- Dr. Khaleel Khan Sir

- Address Binding
- Contiguous Allocation
- Overlays

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

$4K \times 16$

$N \times m$

1W = 2B

$n = \log_2 4K = 12$ bits

$m = 16$ bits $= 2B =$ Data lines by

$N_B = 4K * 2B = 8KB$

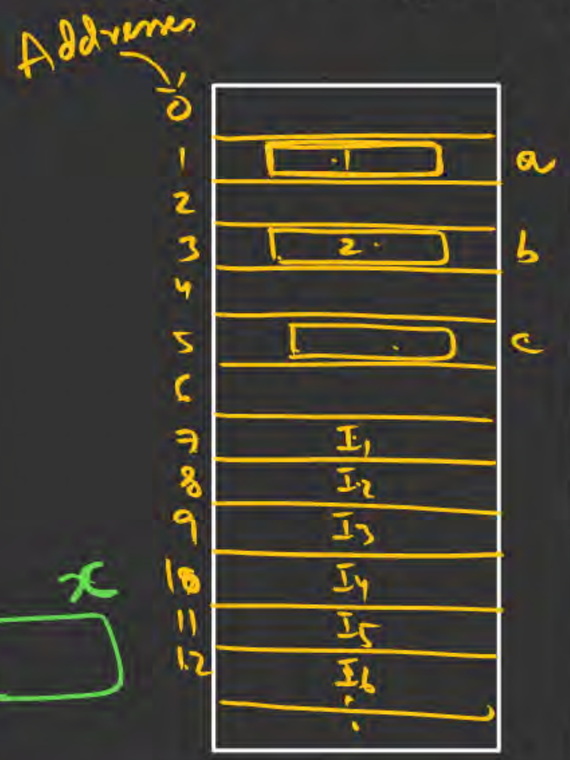
Address Binding : (Association of Prog. Instns & Data units to Memory Locations (Addresses))

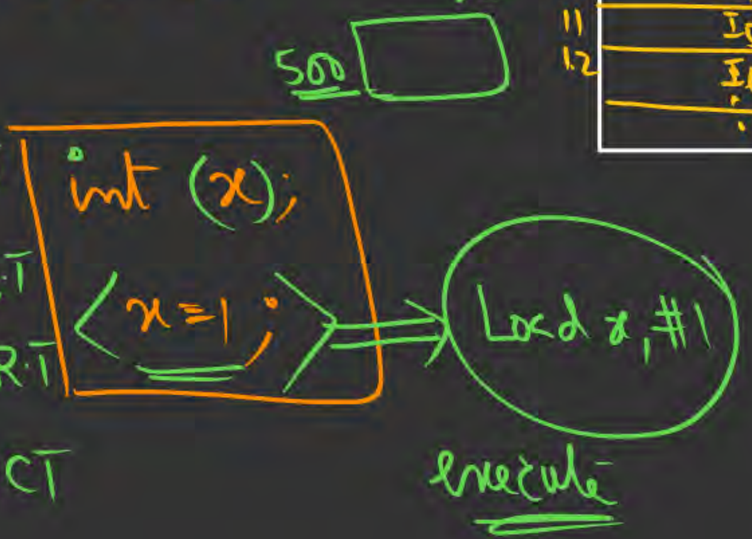
Compile Time, CT
Load Time, LT
Run Time RT

Binding Time: Time @ which the Binding takes place;

```
int a,b,c;
```

a=1; ⟹ I_1: Store a,#1
b=2; ⟹ I_2: Store b,#2
c=a+b; ⟹ I_3: Load r_1, a
I_4: Load r_2, b
I_5: Add r_1, r_2
I_6: Store c, r_1





St Name: CT
St: Type: CT
St: Address: LT
Dynamic: Value: RT
St: Size: CT

Types of Binding:

- Static → (Cannot change)
- Dynamic → (Can change)

Address Binding

Flexibility

Dynamic

Relocation

Static

Compile Time

Compiler

Addrs.
I_1 : 4000
I_2 : 4004
I_3 : 4008

→ Compiler will NOT generate/assoc. addresses

→ Loader will assoc

Offsets

I_1 : 0
I_2 : 4
I_3 : 8

6000 Base Register

OS

(Static)

Load Time

Loader

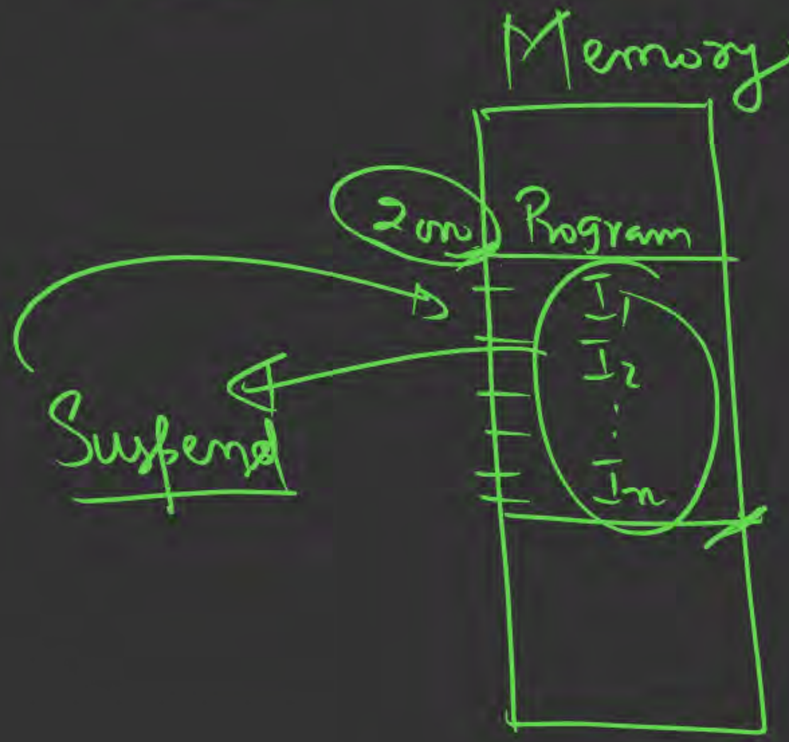
Addrs
I_1 : 6000
I_2 : 6004
I_3 : 6008

Dynamic

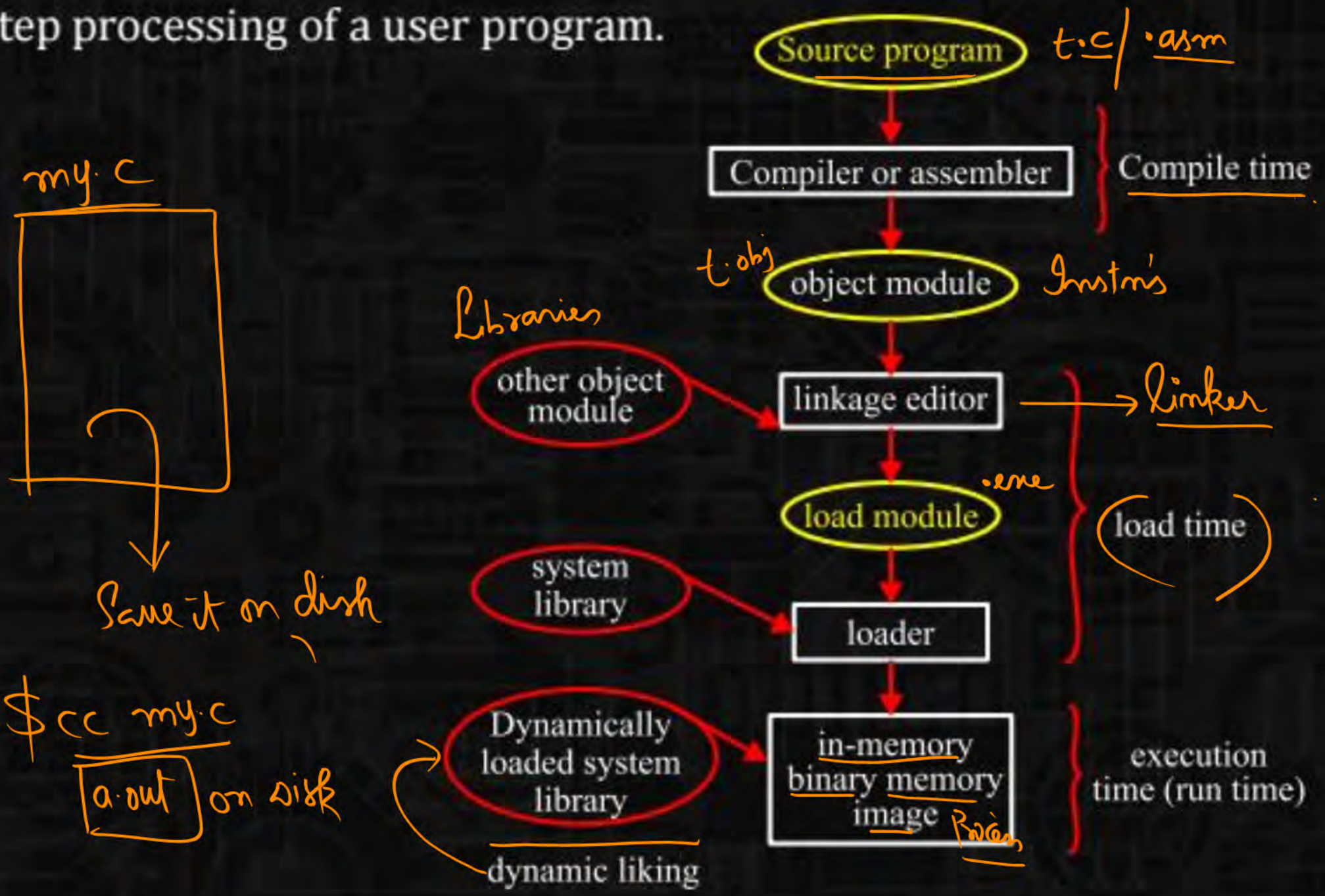
Run Time Address Binding

Loader

Memory



Multistep processing of a user program.



Mem Mgmnt Techniques

(CG) Contiguous <Centralized>

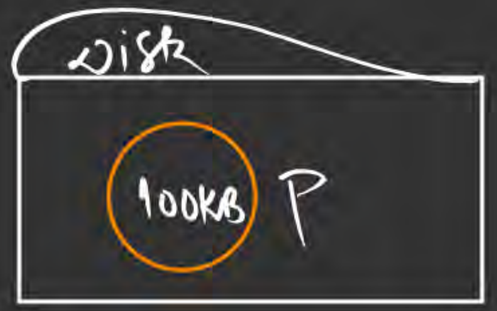
(NCG) Non-Contiguous <distributed>

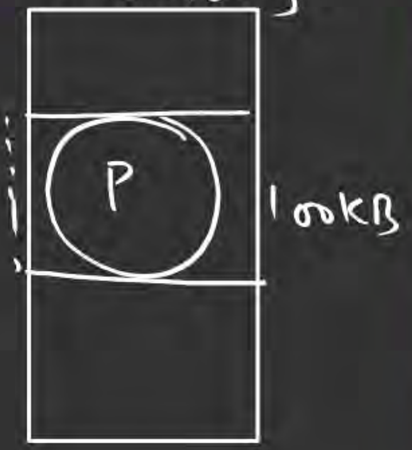
old

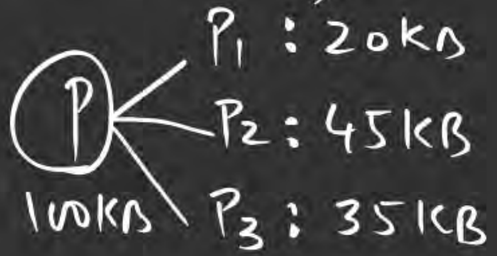
→ overlays

→ Partitions <Fixed var

→ Buddy Sys





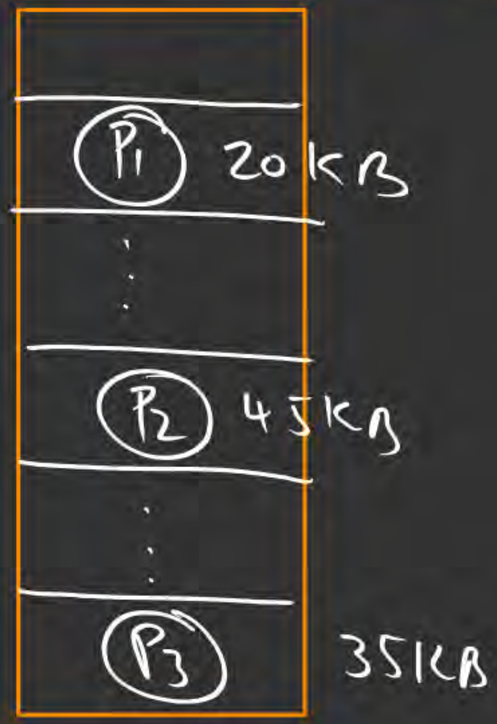


→ Paging

→ Segmentation

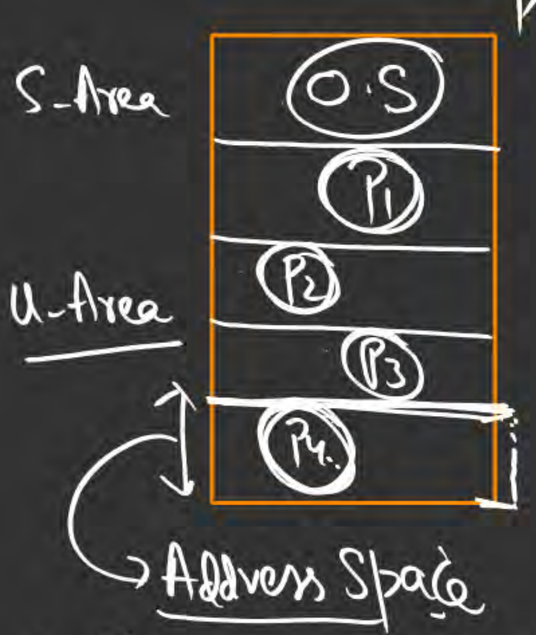
→ Seg-Paging

— Demand ll (V.M)



Functions & Goals of Mem. Manager

1. Allocation ✓
2. Protection ✓

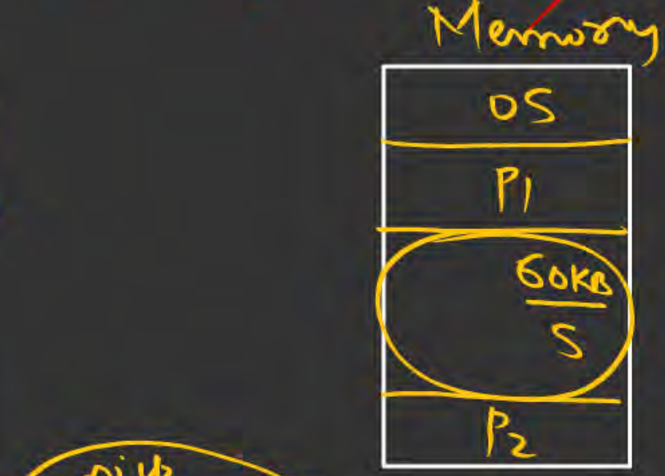


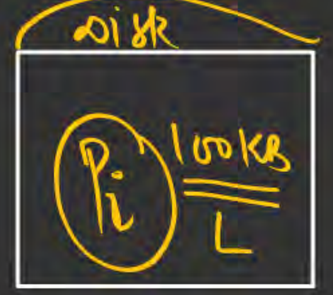
3. Free Space Management
4. Address Translation
5. Deallocation

→ utilize Mem efficiently (Min. wastage)
<Fragmentation>

→ (Ability of MM, to manage execution of Larger Program in Small Memory Areas)
<V.M + Overlays>

(Inc. degree of MPr) ⟹ Throughput / Efficiency





① overlays:

2 – Pass Assembler:

Pass 1 : 70KB

Pass 2 : 80KB

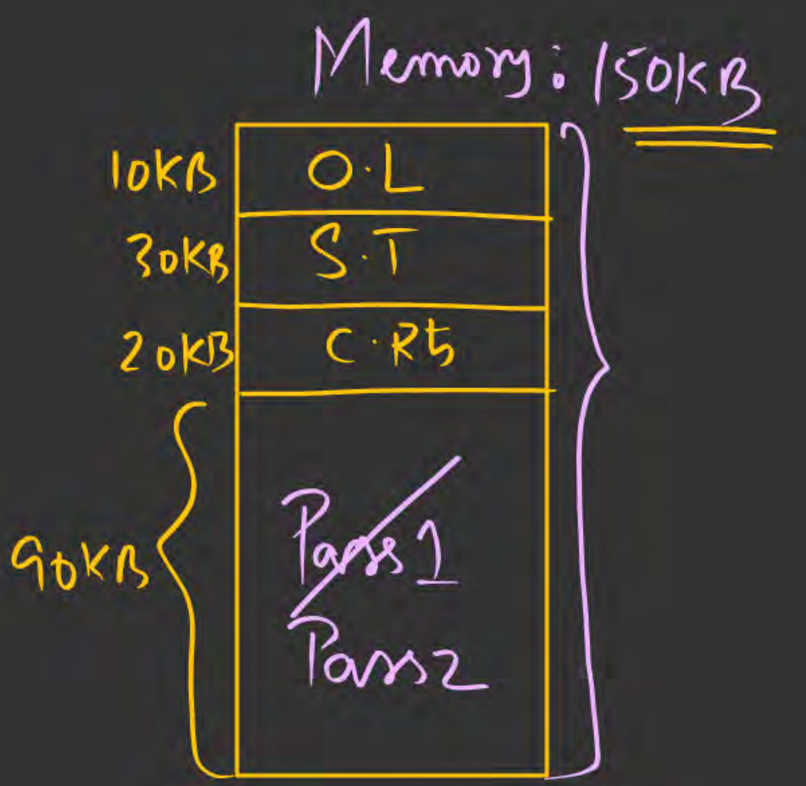
✓ Symbol Table : 30KB

✓ Common Rts : 20KB

✓ OV. Loader : 10KB

: 210KB

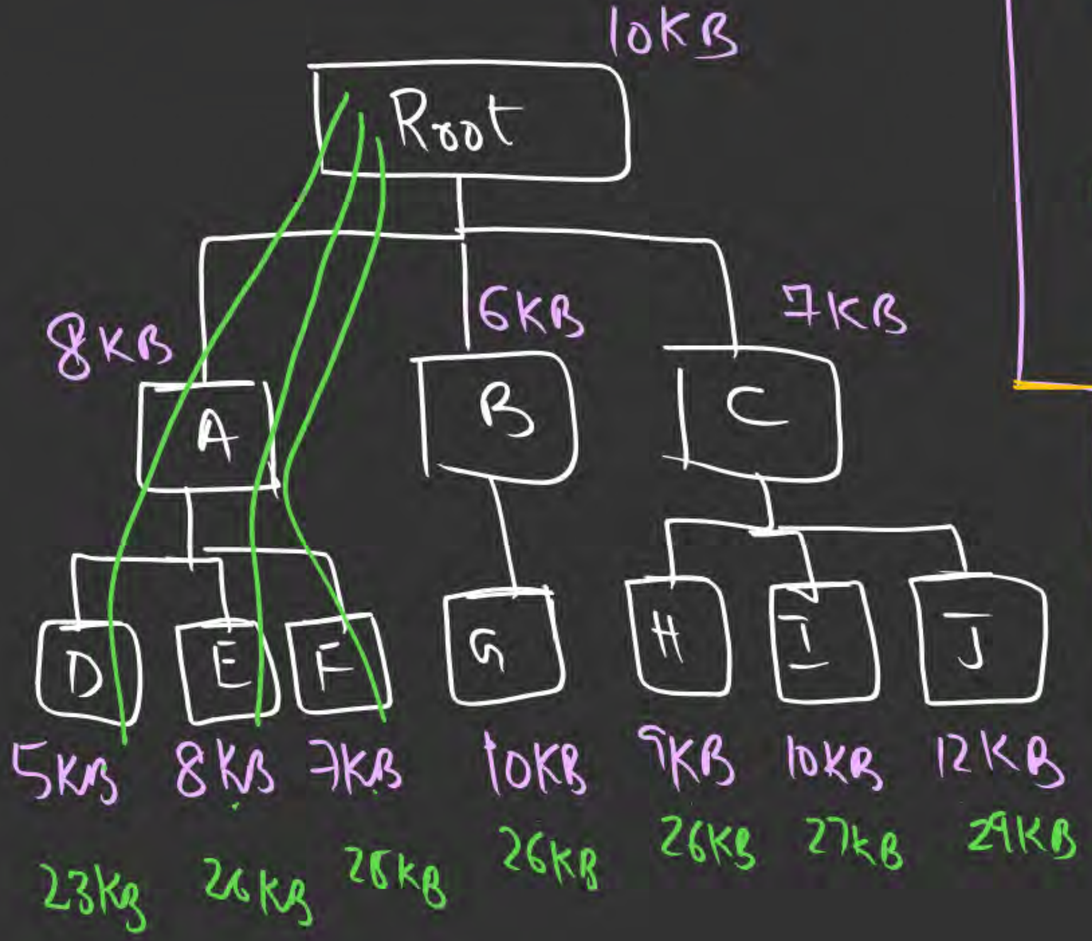
(Pass 1 & Pass 2 are Independent)



overlay = replace

(overlaying Concept is possible only if Program is divisible into independent Modules)

Consider the following Program expressed as an overlay Tree.



Prog-Size : 92KB (on disk)

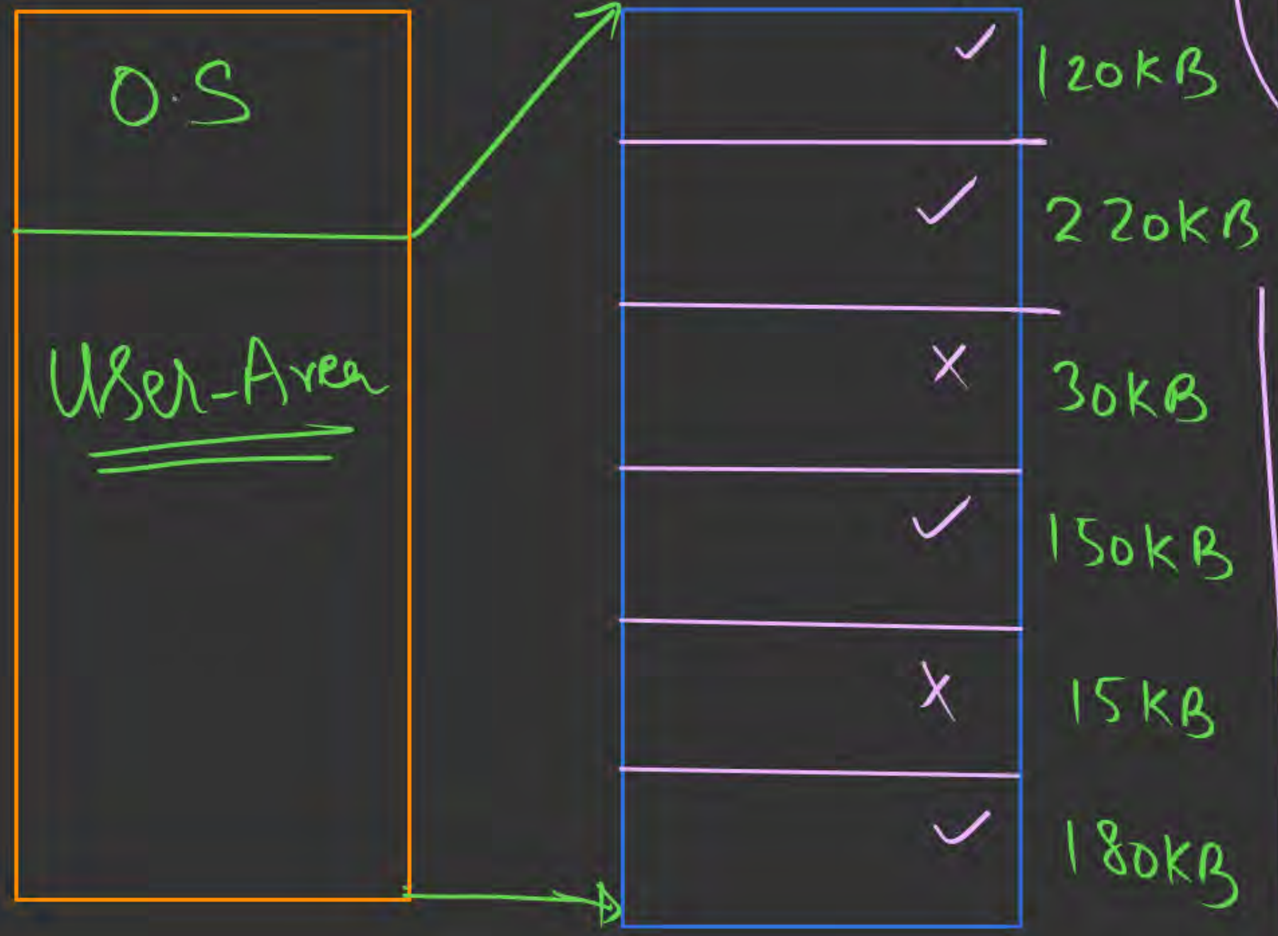
What is Min. Mem Reqd to Successfully execute this Program using overlays?

Min. Mem Reqd = Max { Path-lengths from root to leaf }

= 29KB

* II PARTITIONS:

(i) Fixed Partitions (MFT) → Multiprogramming with Fixed Tasks / Program



Limit Registers → U-Area will be divided into fixed No. of Partitions.

→ 1 Partition = 1 Program

Partition Alloc. Policies

→ Partitions May be of different Sizes,

(Protection)

Allocation

Prem: 80KB

as TQ ~ ∞

R.R ~ FCFS

Starvation Problem

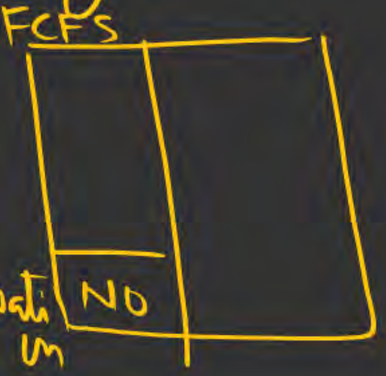
Remote Cases

1) FCFS
2) R.R
3) SJF
4) Priority

(MSQ)

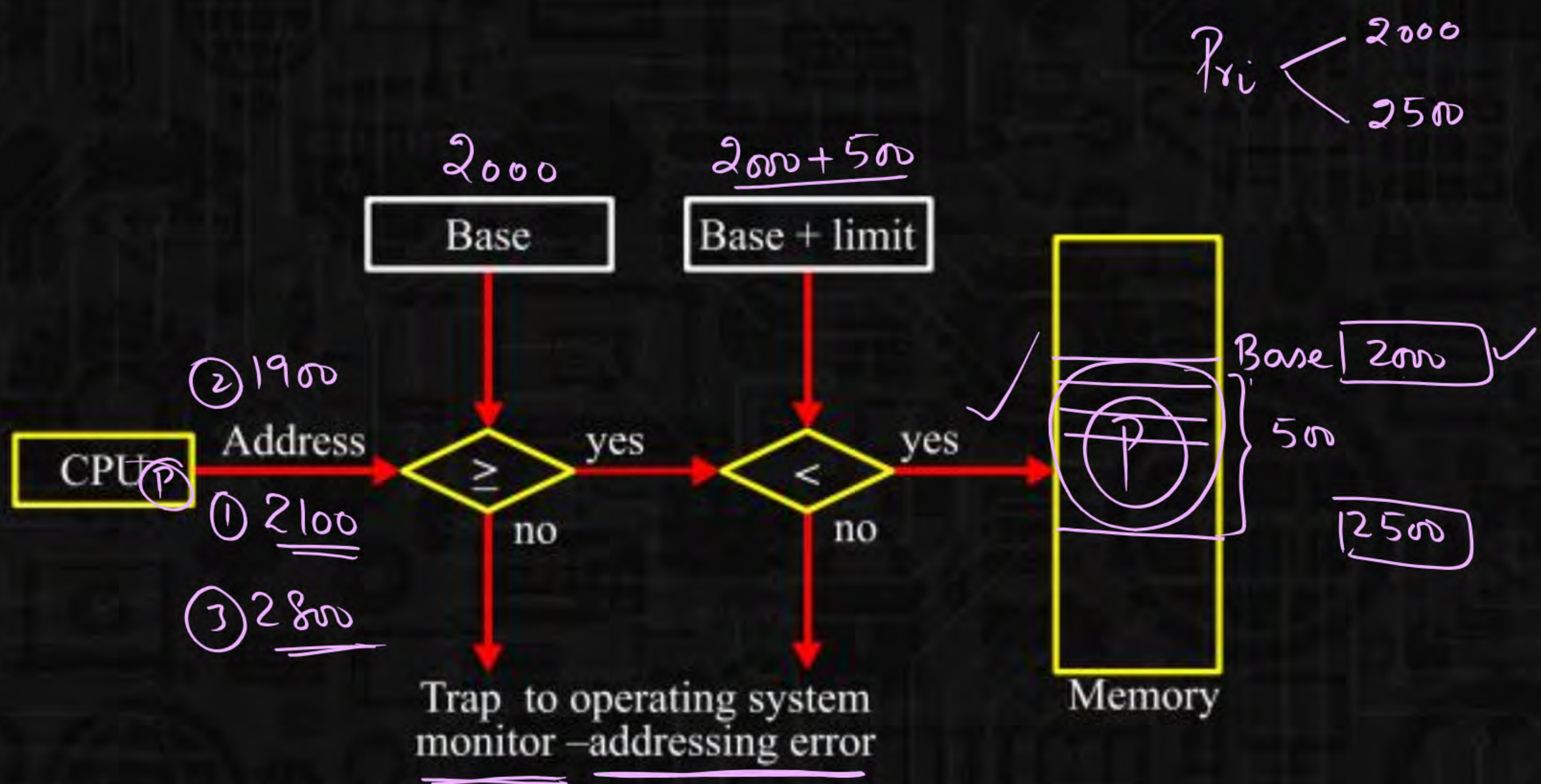
<1,3,4> | <3,4>

① Stallings



② Galvin

③


Pri < 2000
250
2000
2000+500
Base
Base + limit
2100 Address
CPU P
yes
yes
no
no
Trap to operating system
monitor –addressing error
Memory
Base 2000
500
2500


THANK YOU GW
SOLDIERS !