unacademy

Memory Management Techniques - Part I

Comprehensive Course on Operating System

Khaleel Ur Rahman Khan • Lesson 27 • May 13, 2024

unacademy

CS & IT Engineering

Operating Systems
Memory Management

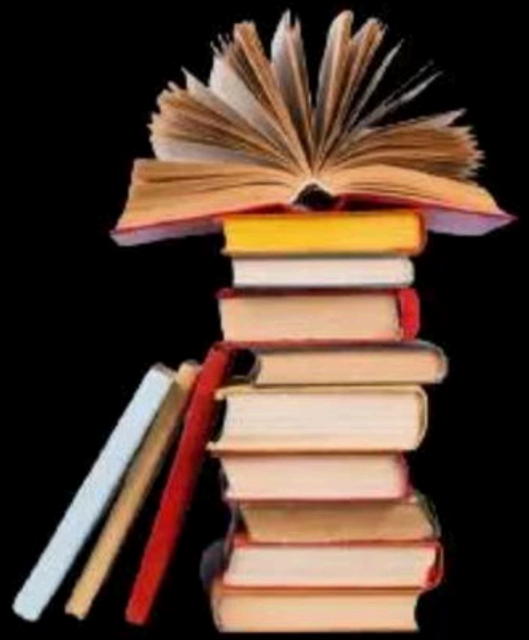
Lecture Number- 26

By- Dr. Khaleel Khan Sir

unacademy

Topics

to be covered



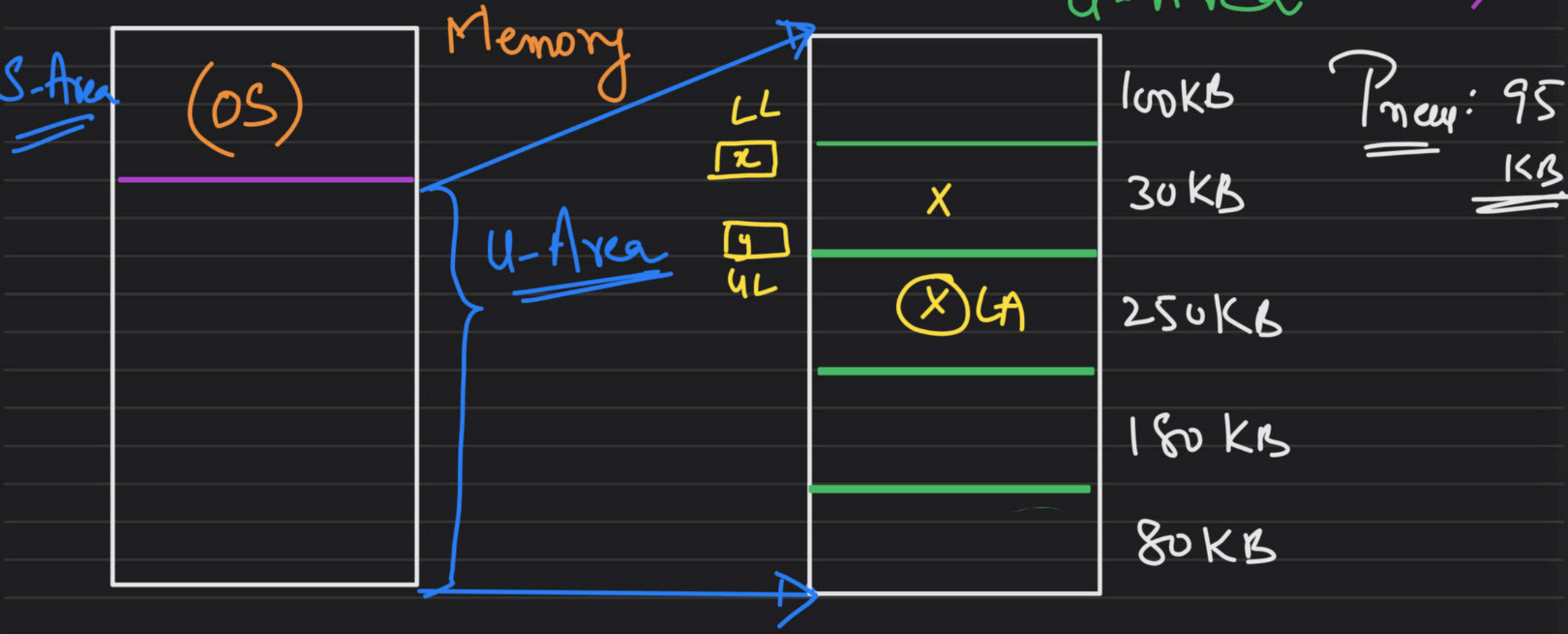
1. Partitioning ✓
2. Address Space ✓
3. Problem Solving ✓

L.A.S
P.A.S

II. PARTITIONING : CG Allocation

- Fixed Partitions (MFT)
  - M.P with Fixed (No.) Tasks
- Variable Partitions (MVT)
  - M.P. with variable (No.) Tasks

(i) Fixed Partitions (M.fT) (Static Partitioning)



→ The U-Area is divided into fixed no. of Partitions, may be of varying Sizes;

→ one partition can hold a single Program

[1:1]

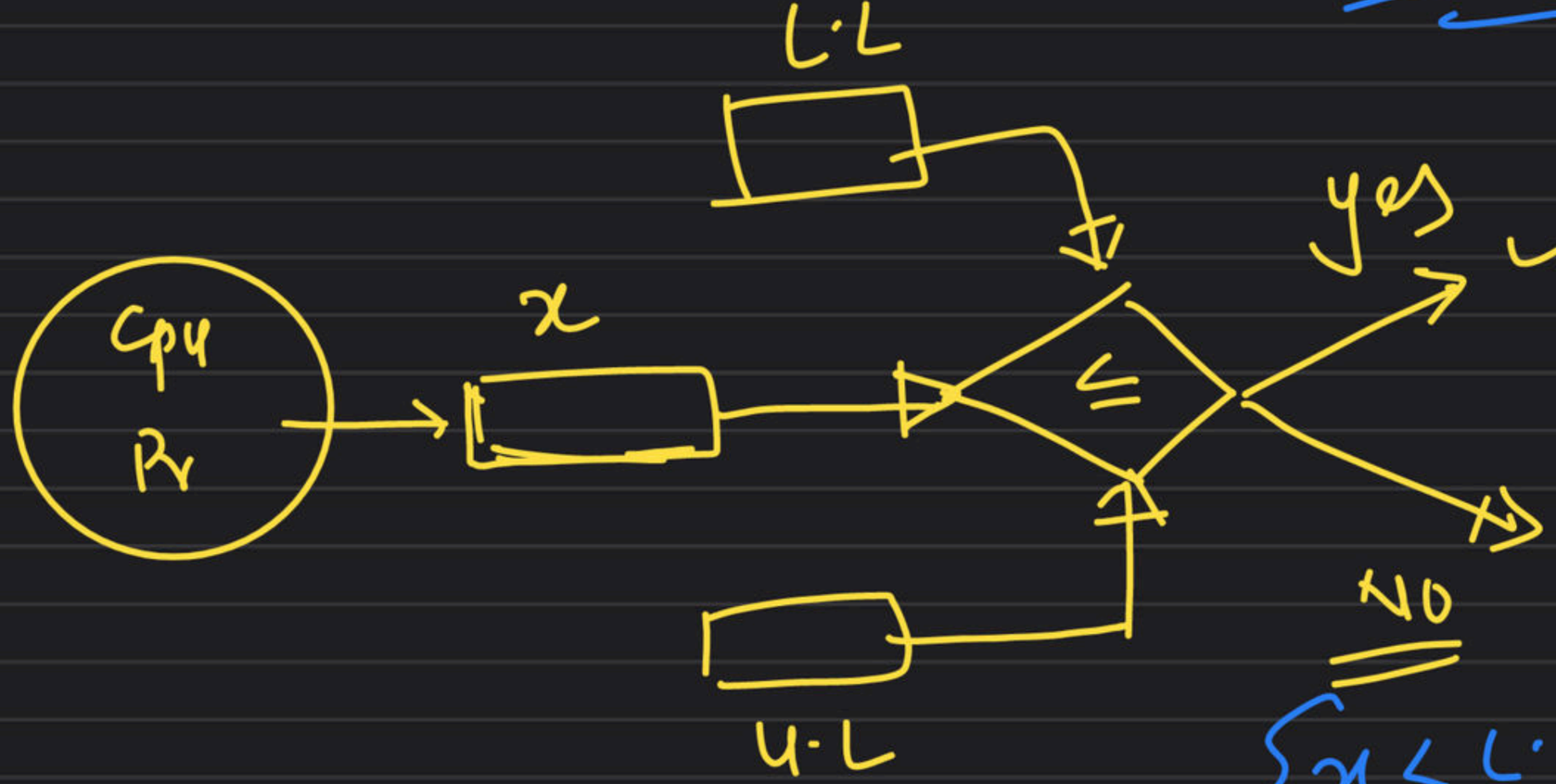
→ Protection is done by using limit registe

→ Allocation: Partition Allocation Policies

a) First fit (FF) b) Best Fit (BF
c) Next Fit (Nf) d) Worst Fit (Wf

$\{L.L \leq x \leq U.L\}$

Protection H/W.



yes ✓ Allowed to access Mem.

No → TRAP (error)

$\{x < L.L$
$x > U.L$

(i) First Fit (FF): First Free big enough.

(ii) Best Fit (BF): Smallest Free big enough

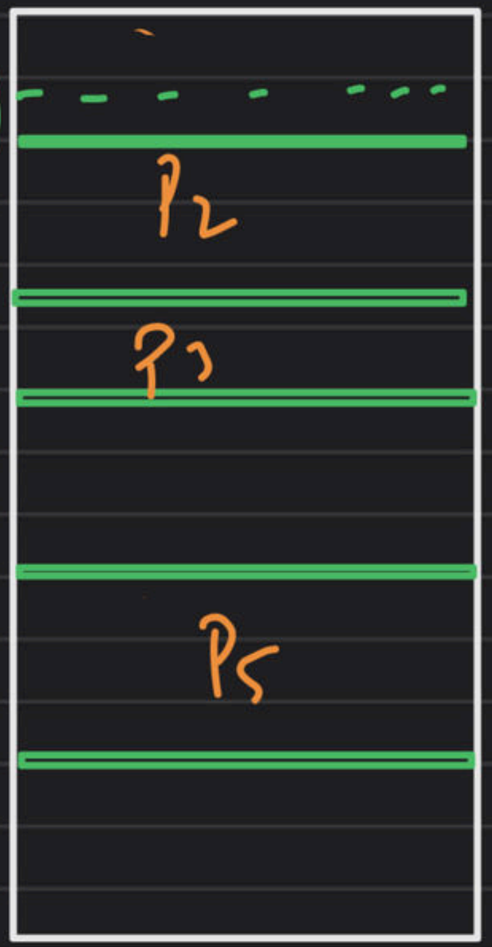
(iii) Worst Fit (WF): Largest Free big enough

(iv) Next Fit (NF): Works like first fit but Search Starts from Partition where Last Allocation was made; It may work faster

Performance :

1) Internal Fragmentation: ✓
   (If)
2) External fragmentation (EF): X
3) Degree of Multiprogramming: Limited to No. of Partitions
4) Max. Process Size: Limited to Max. Part. Size;
5) Alloc. Policy: Best Fit is superior due to Less I.F

2) Variable Partitions (MVT): (Dynamic Partitioning)



t: Process Req's

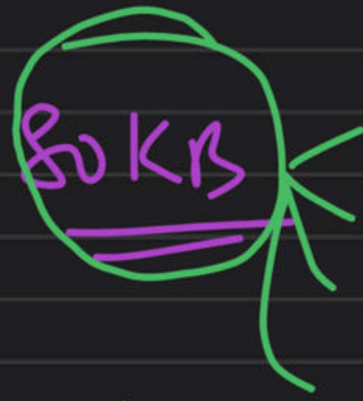
85KB; 120KB; 30KB; 400KB; 250KB

P1 P2 P3 P4 P5

Free Hole

{Coalescing}

Pmem:



FF
BF ✓
NF
WF

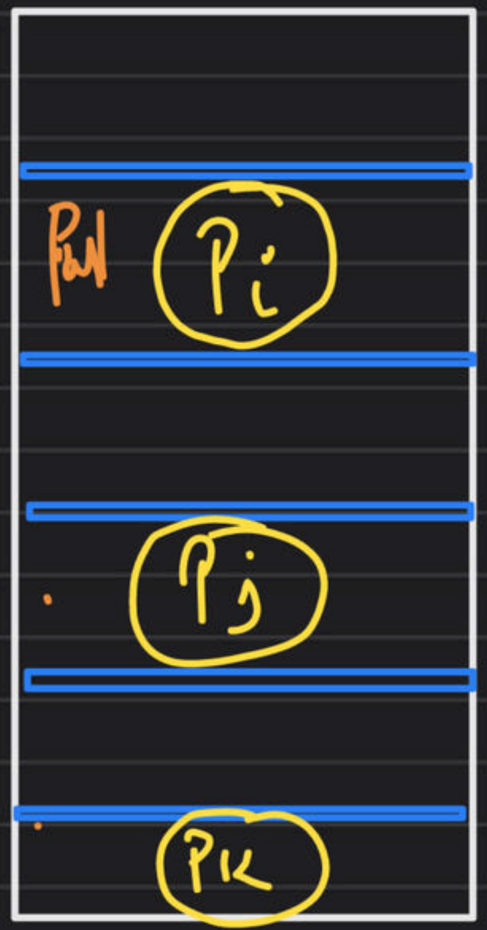
No Process size < 20KB

Performance:

1) Internal fragmentation (I.F): ✗
2) External fragmentation (E.F): ✓
3) Degree of M. Prog : Flexible
4) Max Proces Size : Flexible
5) Allocation Policy : Worst Fit will create larger size free holes that may accomodate new Process.

unacademy

Total free Memory Space. (190 KB)



't'

Pmem : [105KB] CG

(External fragmentation)

External fragmentation (EF):

→ Non-CG.

→ Paging

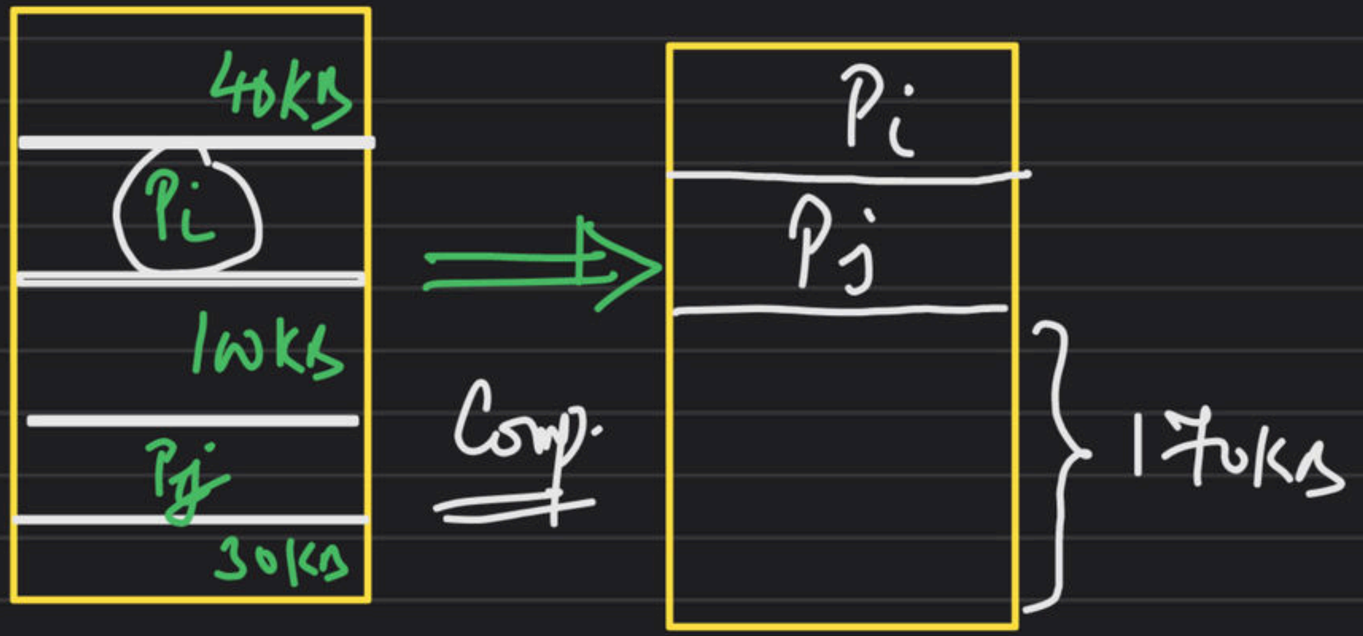
Defragmentation

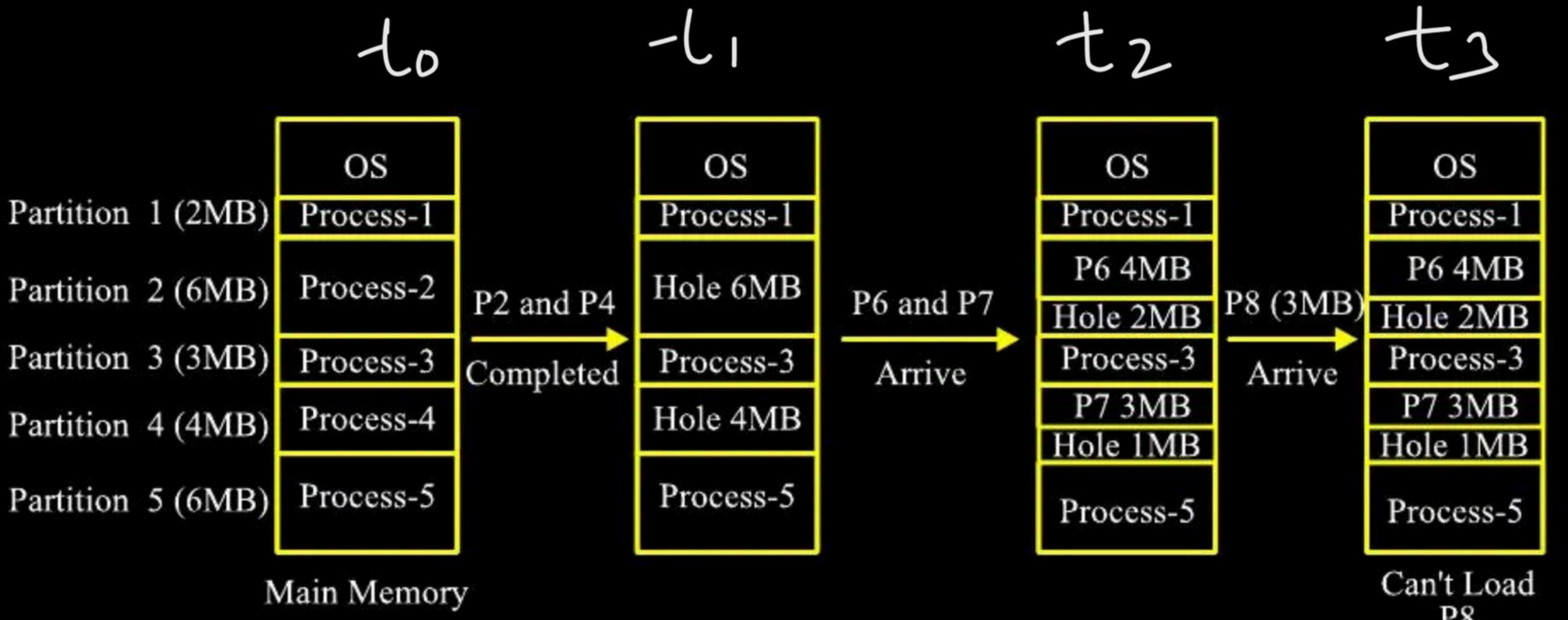
Compaction

→ Time Consuming (Ovhd)

→ R.T Address Binding

Merging of free holes at one end of Memory thru Dynamic Relocation.





Scenario of Variable Partition

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?

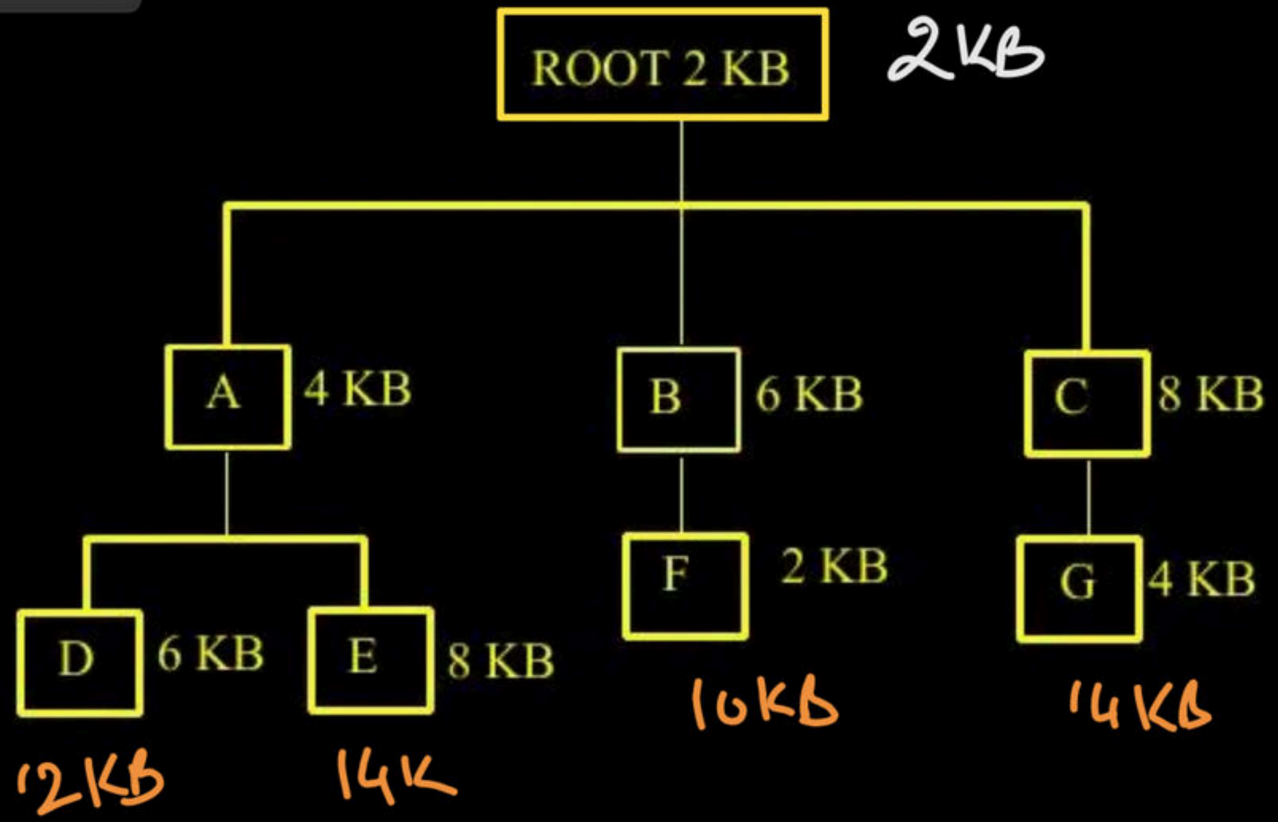
A 10 address, 16 data lines

B 11 address, 8 data lines

C 12 address, 16 data lines

D 12 address, 12 data lines

Q. The Overlay Tree for a Program is as shown below



PYQ

Prog-Size = 40KB

= Max { Path-length from Root to Leaf }

What will be the size of the partition (in Physical Memory) required to load (and run) this program?

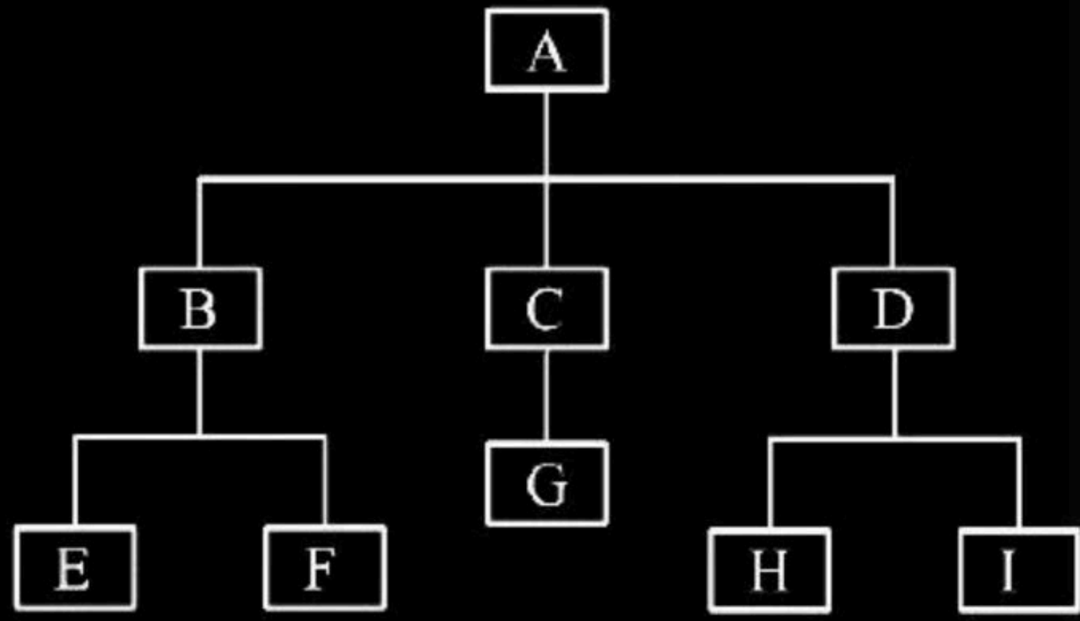
(A) 12 KB (B) 14 KB (C) 10 KB (D) 8 KB

Q. Consider the following graphical representation of program here horizontally connected blocks are independent modules and blocks connected vertically are dependent modules.

[MCQ]

Memory requirement of each block is as follows:

A : 10KB
B : 5KB
C : 3KB
D: 2KB
E: 5KB
F: 10KB
G: 3KB
H: 4KB
I: 6KB



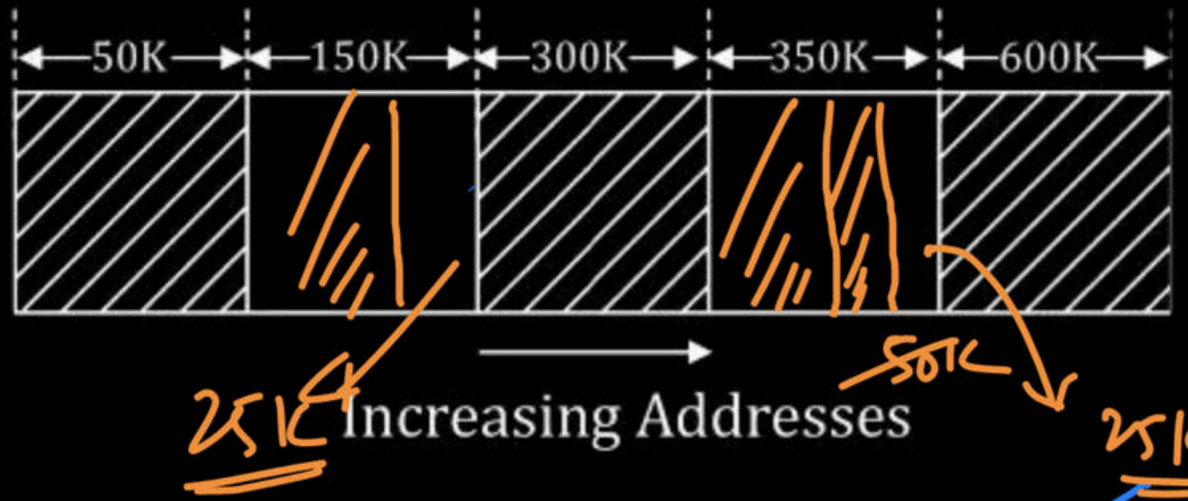
Which of the following memory space is sufficient enough to execute the given program using Overlays?

A. 48 KB
B. 25 KB
C. 50 KB
D. 24 KB

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?

(600, 300)

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.

F·F: ✓

B·F: X

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 ✓

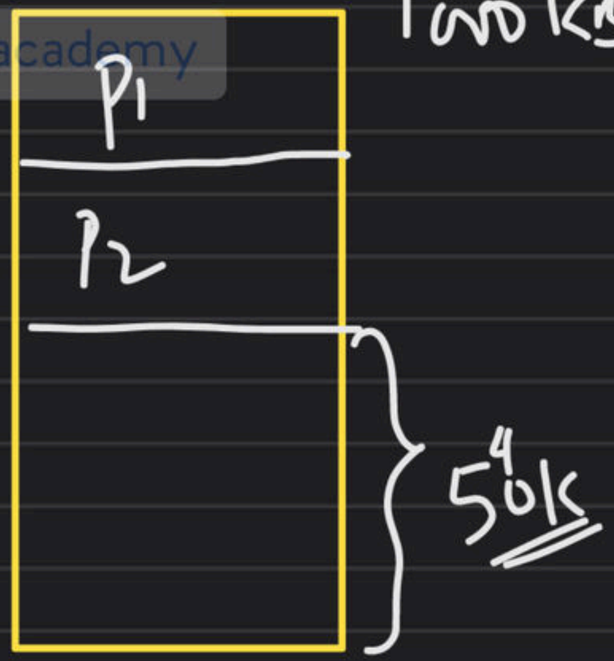
(ii) The smallest Allocation Request which could be denied is:

A. 131 K

B. 151 K

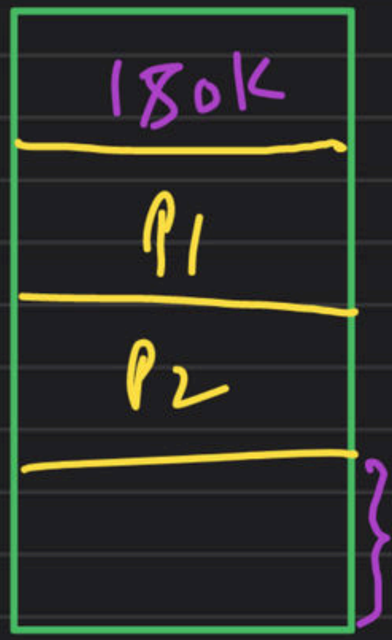
C. 181 K ✓

D. 541 K



(i) 1-free

541K



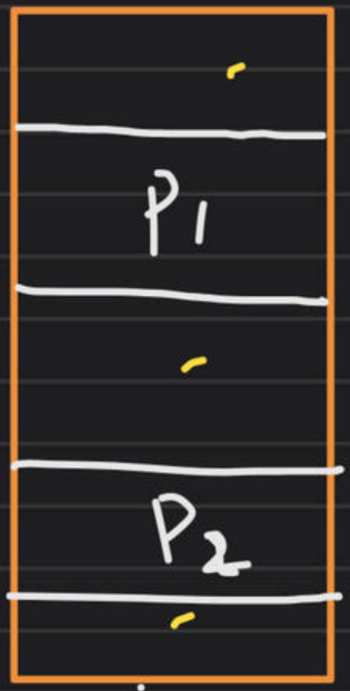
(ii) 2-free Holes

131K ×

151K ×

181K ×

541K ✓



$\frac{540}{3} = 180$

(iii) 3-free Holes

131K ×

151K ×

181K ✓

541K ✓

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. 3B

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

Memory: 2^{46} Bytes

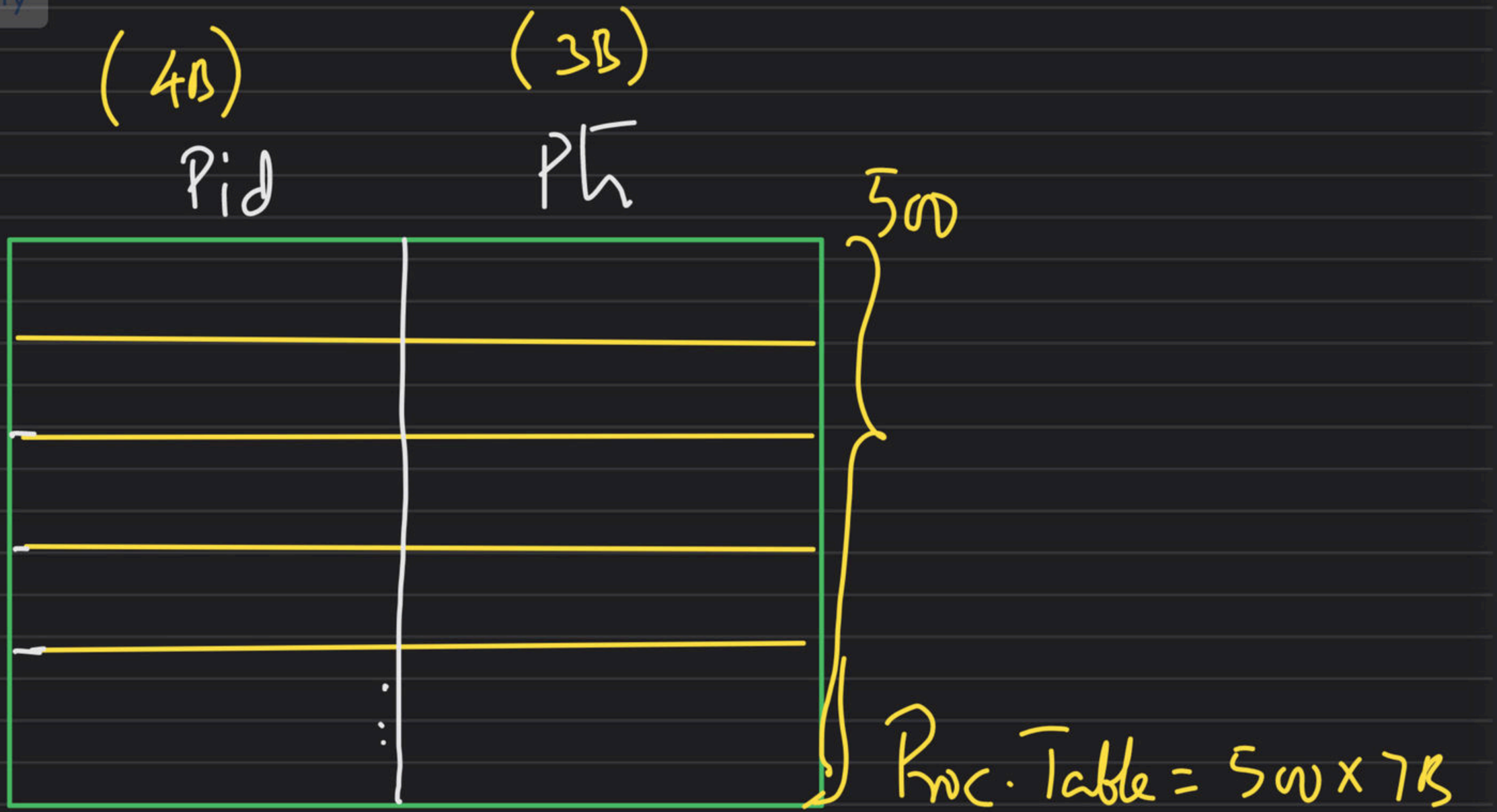
Part. Size = 2^{24}

No. of Partitions: $\frac{2^{46}}{2^{24}} = 2^{22}$

$= 4M$

Each part. is addressed with 22 bits (plz)

pth $= 3B$



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

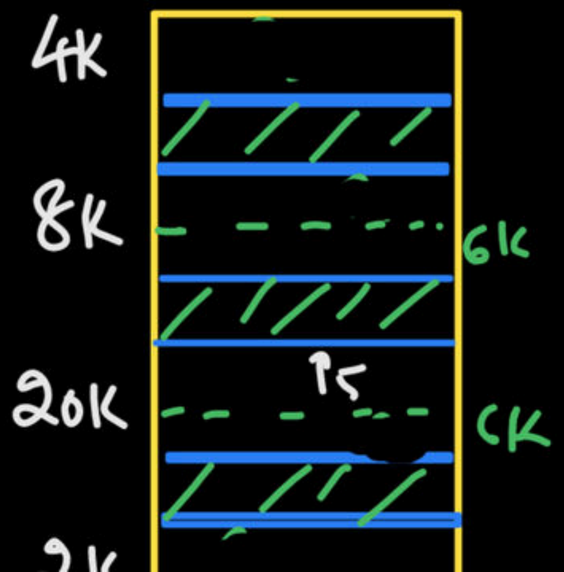
T.O.L = 0

B.T's

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 : ~~P1; P2; P3; P4; P7;~ t_{29} : P6

t_{14} : P5



FCFS

CPU

P1	P2	P3	P4	P7	P5	P6

0 4 14 16 17 25 29 30

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

Q. In a variable size fixed partition-memory management scheme, internal fragmentation occurs when: (MFT)

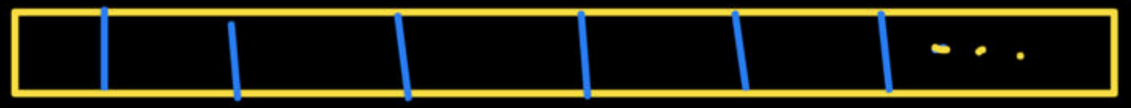
A. Sufficient memory is available to run a program, but it is scattered between existing partitions.

B. Insufficient memory is available to run a program.

C. The partition allocated to a program is larger than the memory required by the program.

D. A program is larger than the size of memory on the computer.

Q. A computer has 1 GB of RAM allocated in units of 64 KB. How many bits are needed if a bitmap is used to keep track of free memory?



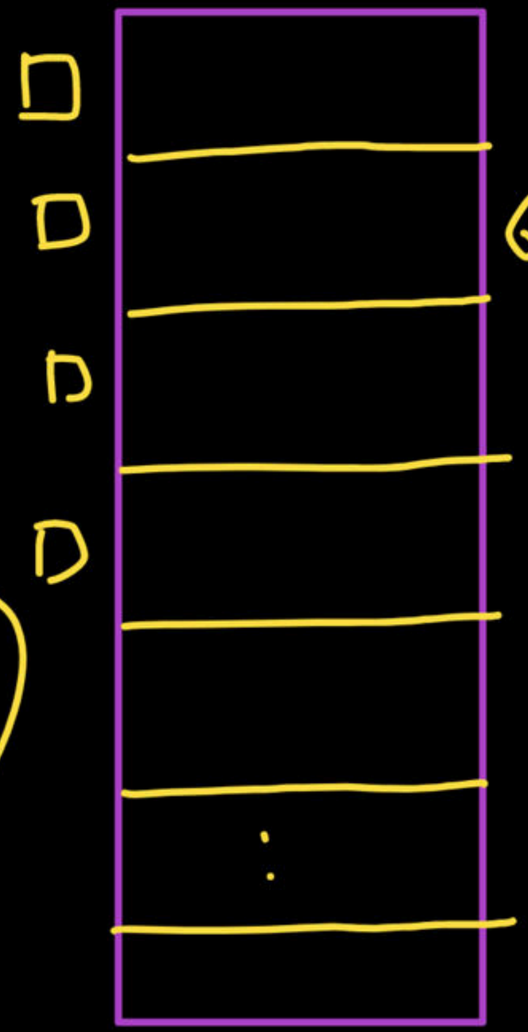
(A) 2^{14}

(B) 2^{13}

(C) 2^{15}

(D) 2^{24}

Memory = 1GB



No. of Part's $= \frac{1GB}{64KB} = \frac{2^{30}}{2^{16}} = 2^{14}$

Q. Let a Memory have four Free blocks of sizes 4K, 8K, 20K, 2K. These blocks are allocated following the Best-Fit strategy. The allocation requests are stored in a Queue as shown below.

Request No	Request sizes	Usage Time
J1	2 K	4
J2	14 K	10
J3	3K	2
J4	6K	8
J5	6K	4
J6	10K	1
J7	7K	8
J8	20K	6

Memory usage time — PYQ

Fixed

t_0:

t_2:

t_4:

t_8:

The time at which the request for J7 will be completed will be

A 16

B 19

C 20

D 37

H/W

Variable Partitions

Logical vs. Physical Address Space

- The concept of a logical address space that is bound to a separate **physical address space** is central to proper memory management
  - **Logical address** – generated by the CPU; also referred to as **virtual address**
  - **Physical address** – address seen by the memory unit
- Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme
- **Logical address space** is the set of all logical addresses generated by a program
- **Physical address space** is the set of all physical addresses generated by a program

What is the Difference Between Logical Address and Physical Address?

Logical Address vs Physical Address	
Logical address is the address at which an item appears to reside from the perspective of an executing application program.	Physical address is a memory address that is represented in the form of a binary number on the address bus circuitry in order to enable the data bus to access a *particular* storage cell of main memory, or a register of memory mapped I/O device.
Visibility	
The user can view the logical address of a program.	The user can not view physical address of program.
Method of Generation	
CPU generates the logical address.	MMU computes the Physical address.
Accessibility	
The user can use the logical address to access the physical address.	The user can not directly access physical address.

Paramenter	Logical Address	Physical Address
Basic	generated by CPU	location in a memory unit
Address Space	Logical Address Space is set of all logical addresses generated by CPU in reference to a program.	Physical Address is set of all physical addresses mapped to the corresponding logical addresses.
Visibility	User can view the logical address of a program.	User can never view physical address of program.
Generation	generated by the CPU	Computed by MMU
Access	The user can use the logical address to access the physical address.	The user can indirectly access physical address but not directly.

Logical Address vs. Physical Address

Logical Address	Physical Address
The process address space can be considered as a sequential list of bytes. Each byte has an address that is used to locate it. These addresses are called logical addresses	The entire physical memory can be considered as a sequential list of bytes. Each byte has an address that is used to locate it. These addresses are called physical addresses
Logical addresses are generated by the CPU.	Physical addresses are seen by main memory.
LAS-Logical Address Space • It is set of all logical addresses that can be referenced by a process • Addresses in LAS starts from zero and goes up to some maximum value based on the size of process	PAS-Physical Address Space • It is set of all physical addresses occupied by a process in main memory during its execution • Addresses in PAS may or may not be contiguous based upon the memory allocation method
Process can read and write addresses of its own logical address space.	Process can read and write only those physical addresses that belong to its own physical address space.
The logical addresses are limited by the address size of the processor Ex. 32-bit processor can generated up to 2^{32} addresses.	The physical addresses are limited to the amount of installed memory.

unacademy

THANK YOU!

Here’s to a cracking journey ahead!