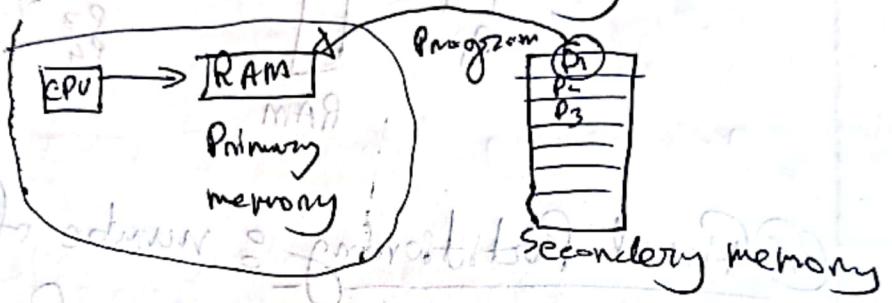
Memory Management

→ Method of managing (memories → ram, h/d, registers.)
Primary memory.

→ goal: Efficient utilization of memory.



• Program must brought secondary memory into Primary memory. ⇒ degree of multiprogramming
more & more processes in Ram.

✓ Degree of multiprogramming:

CPU Utilization = $1 - p^n$

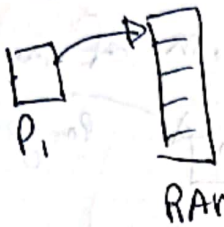
n → num of process.
P → I/O block time.
(অর্থাৎ সময় CPU ব্যস্ত)

ex: [3 process @ memory], 80% I/O time,

∴ CPU utilization = $1 - (0.8)^3$
= 49%

(iv) Memory management technique → Contiguous / Non-contiguous

Contiguous Allocation: (i) Fixed Partition (Static)
(ii) Variable " (Dynamic)

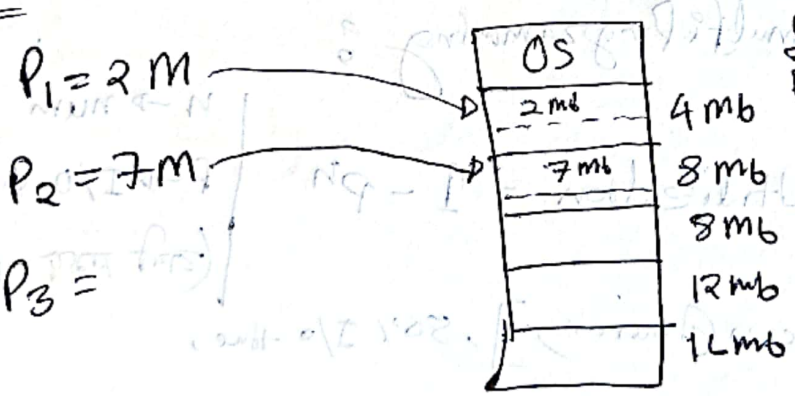


P1
P2
P3
P4
↓ Serially.

(✓) Fixed Partitioning: numbe of partition are fixed.

(i) Number of partitions are fixed.
(ii) Size of each partition not same.
(iii) Spanning not allowed.

ex-

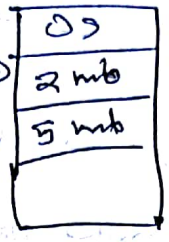


cons:
[2mb waste হলো, অন্য কেউ ব্যবহার করতে পারে না - internal fragmation.]

✓ fragmation হচ্ছে,
✓ Limit আছে Process size.
✓ Partition এর থেকে ছোট Process আসতে পারবে না।

Variable Partitioning: ✓ no partitions ✓ will be created according to the Procan size.

$P_1 = 2$ mb
$P_2 = 5$ mb

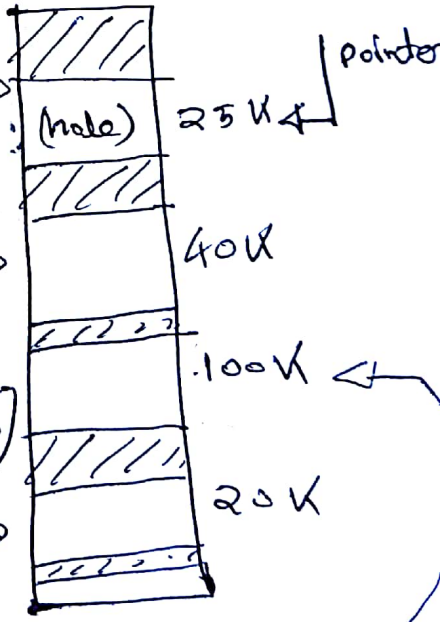


Cons:
✓ hole create if procan execute

Memory Allocation –

✓ First fit: Allocate the first hole that is big enough. $P_1 = 15K$

✓ next fit: Same as first fit but start's searching from last allocated. $P_2 = 18$



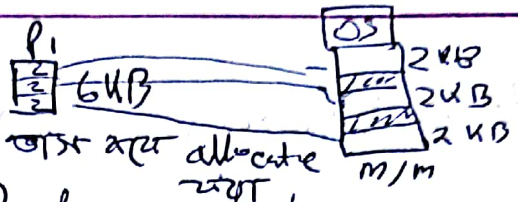
✓ Best fit: (min internal fragmentation) will search all holes.

$P_3 = 15$ — best fit

✓ Worst fit: (large hole search)

$P = 15$

Non-Contiguous -



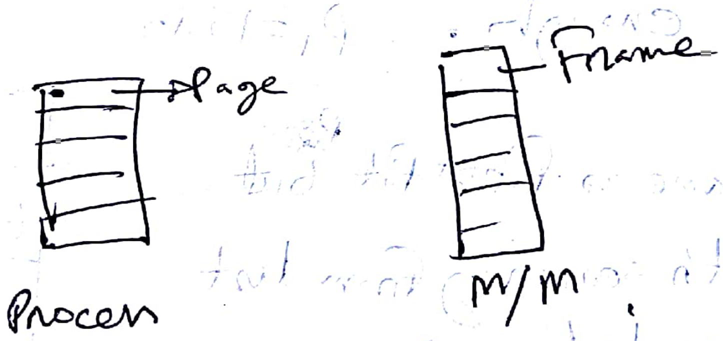
6KB তাকে ভেঙে allocate করা।

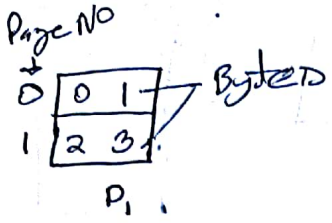
Paging

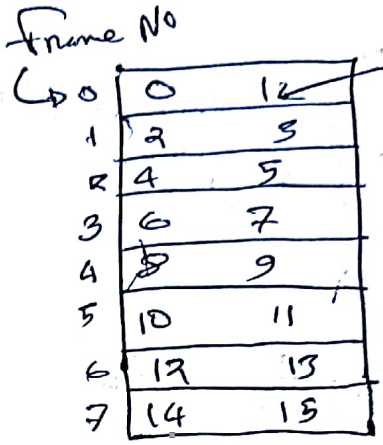
✓ Secondery memory এ Process divide করার ফলে Page এ ভাগ করতে হয়,

✓ Main memory divide করে ফলে Frame এ ভাগ করে

[Page size = frame size]







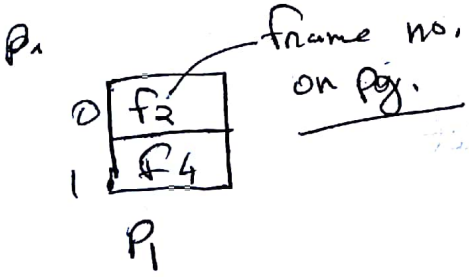
Procen Size = 4 KB

Page size = 2B.

No. of pages/Procen = $4/_2$KB = 2B

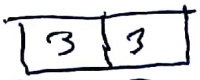
[

Page Table : revery procen has it's own page table.



calculation

Pg number — Pg offset

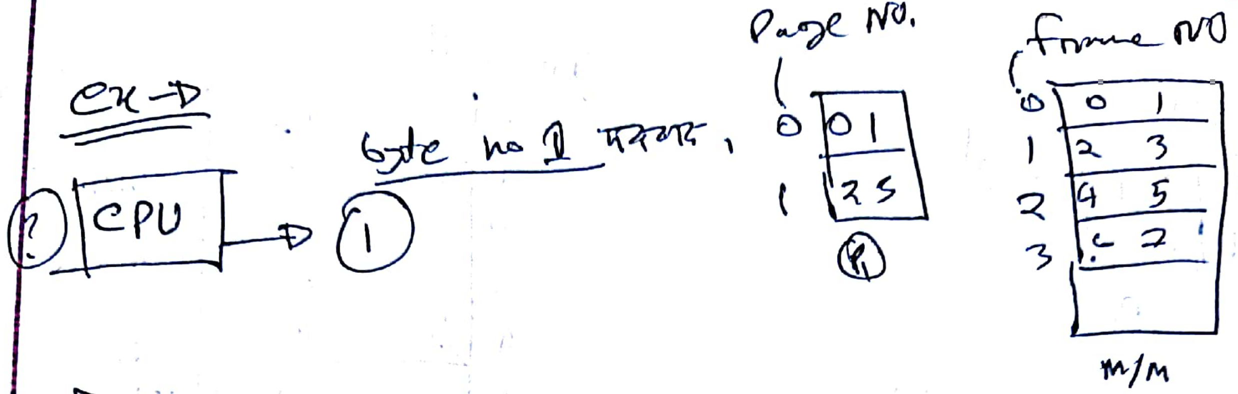


frame num — frame offset

logical address

physical address.

ex ->



=>

Logical address

0	1

Page no. — Page offset

(Logical to physical address convert)

Page table

0	f_2
1	f_4

Physical address

<-3-> <-1->

010	1

frame offset

[illegible]

Bin -> 0101

$2^{10} = 1K$

$\Rightarrow 2^{20} = 1M, 2^{30} = 1G, 2^{40} = 1T.$

Memory in bytes addressable

\#

Logical adhen space / size of procen = 4 GB

physical addrun space / = 64 MB

Page size = 4 MB

(?) No of pages ? $\Rightarrow$

(?) No. of frames ? $\Rightarrow$

(?) No. of page table $\Rightarrow$

\# | 4 | 3 |

Logical addrun

| 3 | 3 |

Phy addrun

→ no. of pages $= 2^4 = 16$

→ no. of frames $= 2^3 = 8$

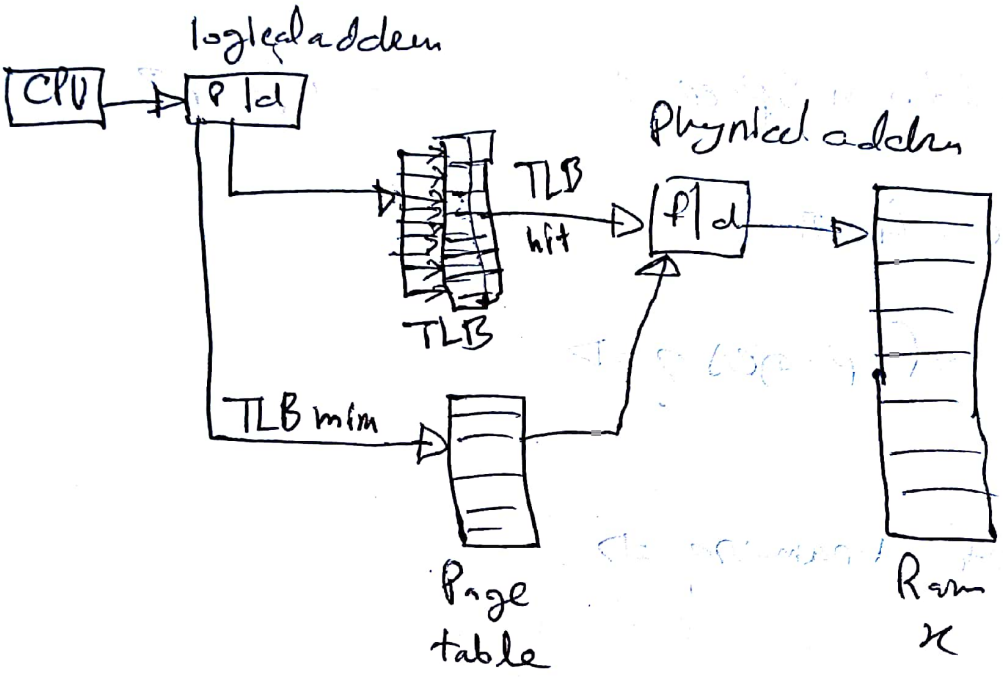
no. of

→ Logical addrun space $= 2^7 = 128$

→ physical " " $= 2^6 = 64$

Translation Lookaside Buffer

(TLB) cash memory



Effective acces time ⇒

(*) [TLB Hit × (TLB acces time + memory acces time)

+

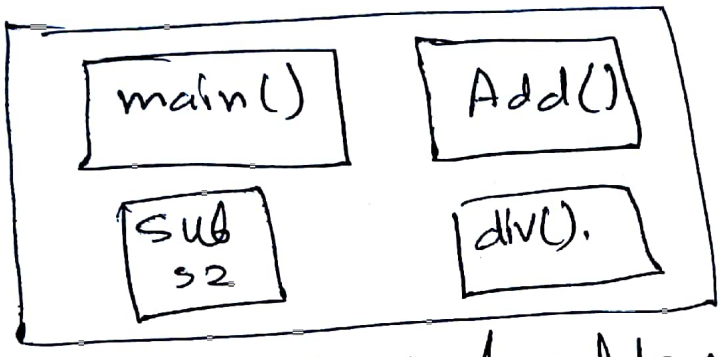
(*) TLB miss × (TLB acces time + Page table acces time + memory acces time)].

(ns)

Segmentation

✓ don't devide the process without knowing like paging.

✓ devides into diffrent unit → function, obj, main() method etc.



← various size.

logical address.

✓ ✓