Contigous Allocation.

. Main Memory must support → OS and user process.

[Memory allocation মধ্যে Contigous Allocation.]

(a process must be given specific amount of memory to its needs.)

Needs এর উপরে ভিত্তি করে একটি single contiguous piece of memory blocks assigned হবে.

Main Memory তে enter করতে চাইলে একটি Contigous segment allocate হবে

(problem)

(Strategies used in categories)

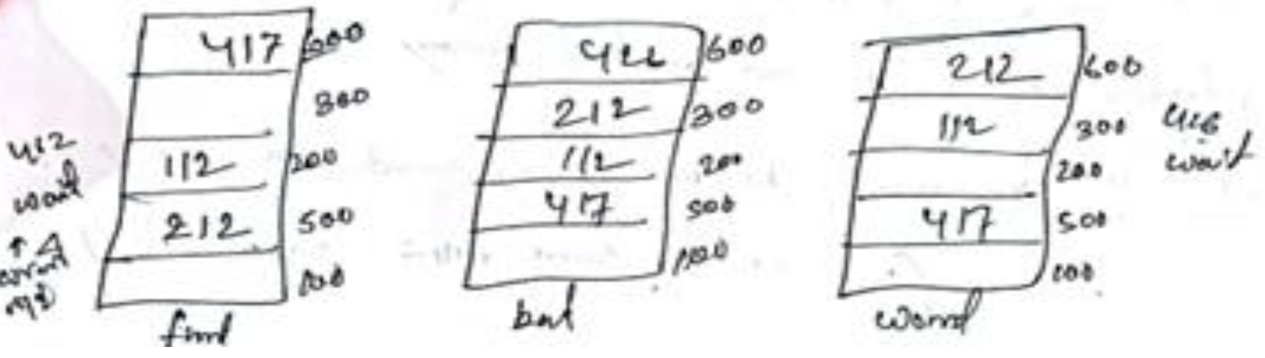
Memory allocation Methods
(hole allocation)

first fit → same value বা বড় হলে সেটাই সুবিধা বসাইদিবে
তাও যদি না যদি মেলে না
না পেলে থাকবে wait

best → ন্যূনতম জায়গা ফাঁকা থাকলে space বসাবে
যেমন 112 হলে, 200তে জায়গা left (200-112)
= 88 space.

worst - memory wastage হয় যেমন 112 জায়গা 600তে
রাখে বেশি waste হয়

Ex - 100, 500, 200, 300, 600 put the value 212, 417,
112, 426 in f. b. w. fit (to place)



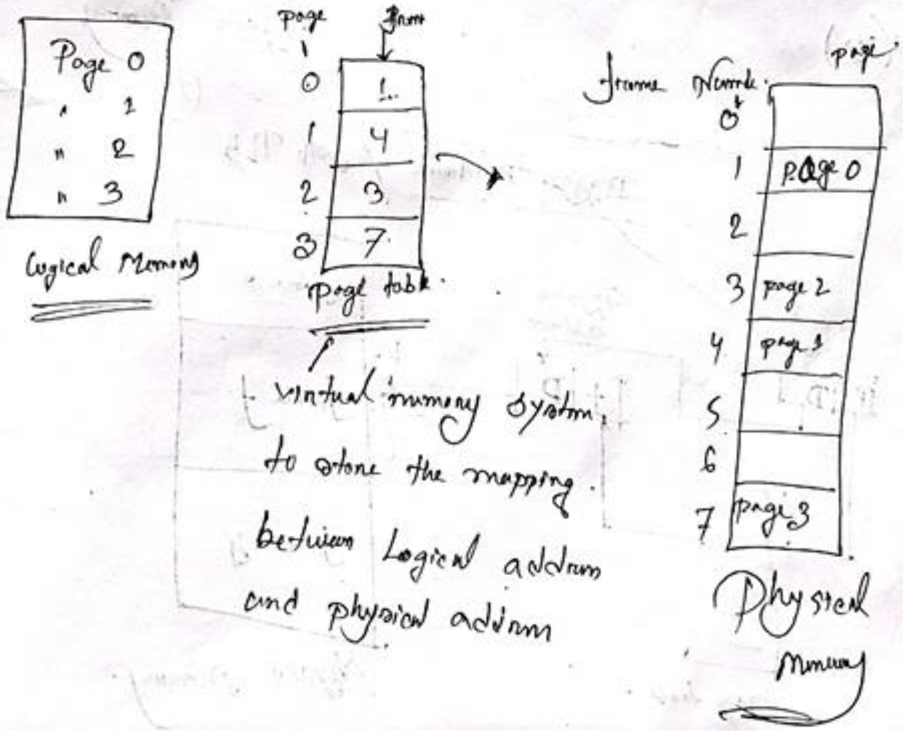
Paging

In os. a storage machonism used to retrive process from Secondary Storage into the main memory in the form of. pages.

[Divide each process foram of pages]

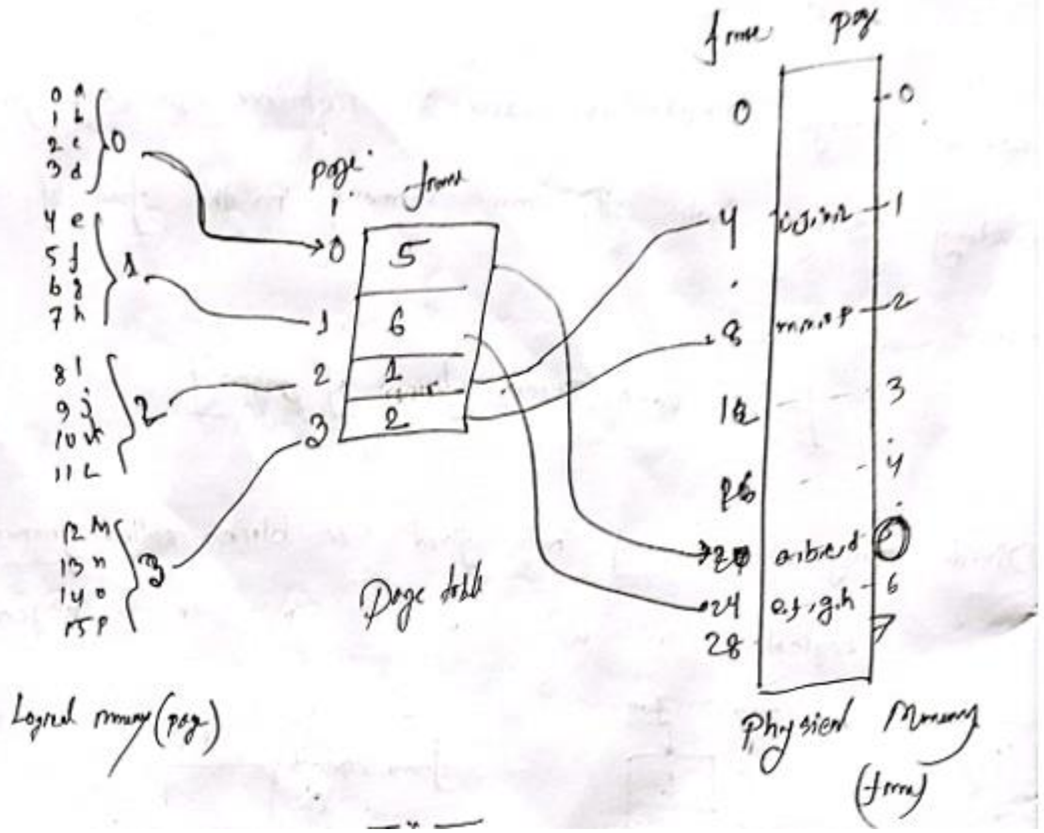
Divide physical memory into fixed size blocks called frames

" Logical " " " " " " pages.



virtual memory system, to store the mapping between Logical address and physical address

Paging Example



* Effective Access Time. (EAT)

Hit reatio ⇒ % of. times that a page Num is found in the TLB.

⊛ Assume hit reatio is 90%. Now Calculte the EAT, one (1) memory accem need $\boxed{10ns.}$

$(100-90)\% = 10\%$

$EAT = (0.90 * 10ns) + (0.10 + 1ns)$

90%.

$= 9.1 Ns.$

memory

Effect of page size

Larger page size	Smaller page size
ADJ	ADJ
· reduced overhead	· reduced internal fragmentation.
· TLB Efficiency	· Lower page fault
Dis ADJ	Dis adj
· Internal fragmentation	· increased overhead
· Increased Page fault -	· potential TLB Misses.

Hashed page table.

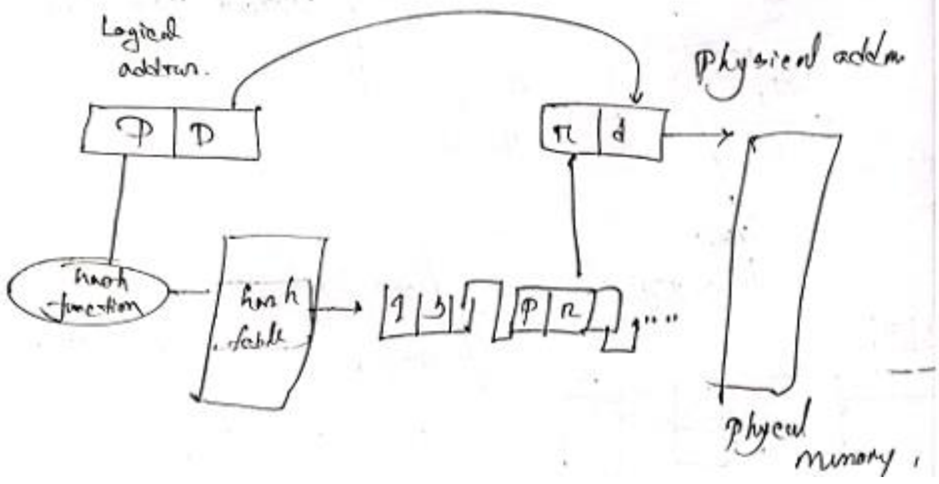
- address space > 32bits.

[এর chain of elements having যে same location)
জায়গা

Each element contains.

- virtual page number . the value of the mapped page frame
- A pointer of next element.

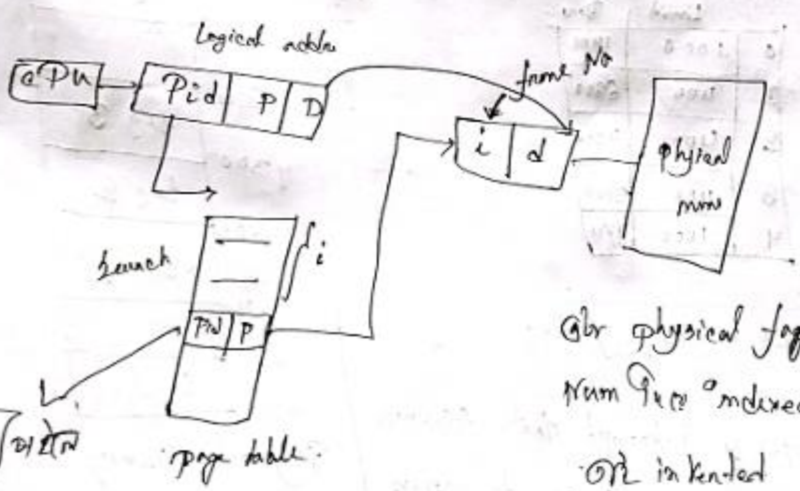
এর chain searching করে
আমাদের page frame remove করে দেই



Inverted page table (8)

এখানে combine করে 1 টা page table এবং frame table.

- One entry for each virtual page Number and real page of Memory.
- It increase the time that is needed to search the table. whenever a page reference occours.



এর physical frame Num গুলো indexed হয় তাই inverted

Pid P খুঁজে

(i) frame No find হবে.

Logical addres space 256 pages with 4KB page size mapped into physical memory of 64 frames.

ⓐ How many bit required in logical addres

$\Rightarrow$ 256 pages = 2^8.

4 KB page-size = 4×1024 ← KB মান

$\Rightarrow 4096 = 2^{12}$.

so = $(12+8) = 20$ bits. / $2^{12} \times 2^{8} = 2^{20} = 20$ bits

(b) How many bits required for physical addres.

4KB p-size = $4 \times 1024 = 4096 = 2^{12}$

64 frames = 2^6

total $(12+6) = 18$ bits.

{ $2^{(n)}$ value calcule করব only for find bit }

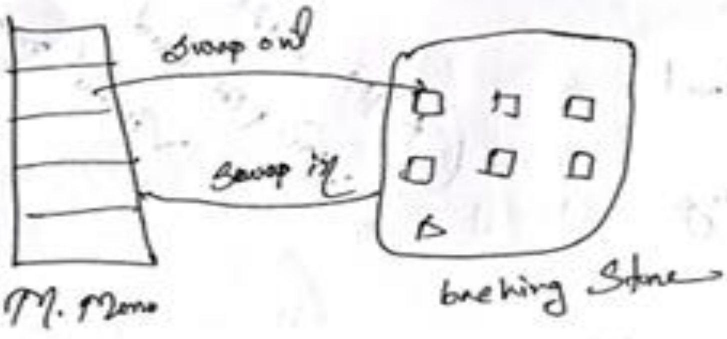
Demand pagging *

· Page না থাকলে page fault হয় তখন Secondary থেকে নিয়ে এসে page table fulfill করে।

Lazy swapper: Never swaps a page into memory unless page Will be needed.

(page যারা deal করে pager)

· Bring a page into memory only When needed.



Adv

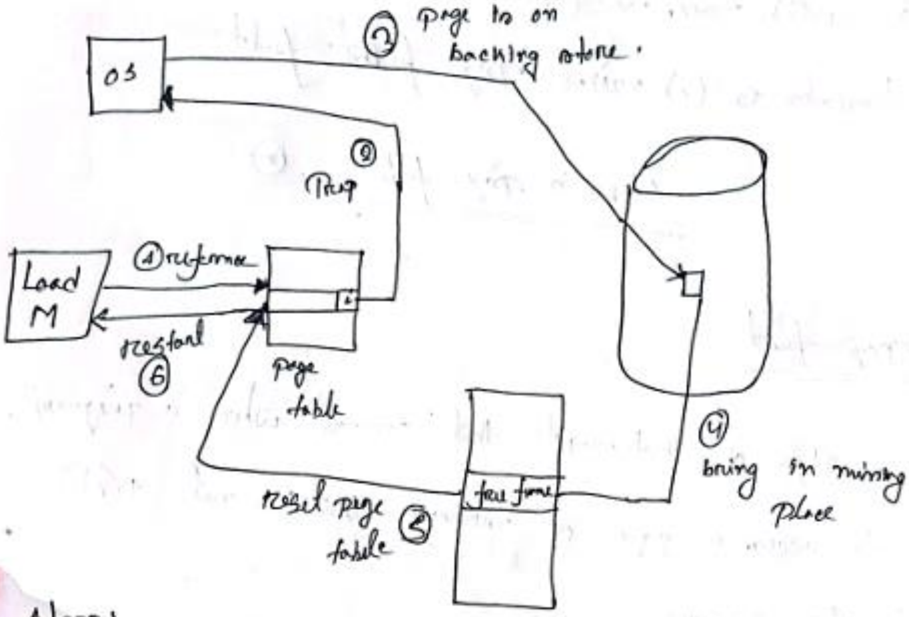
- less I/O need
- " Memory "
- faster response
- More users.

(page set করে MMU)

Steps for handling

Page fault *

· Page fault होता $\boxed{\text{user} \rightarrow \text{OS}}$ shift हो control.



Steps:

· reference that page will trap to operating system (page fault)

· ck करो invalid or not

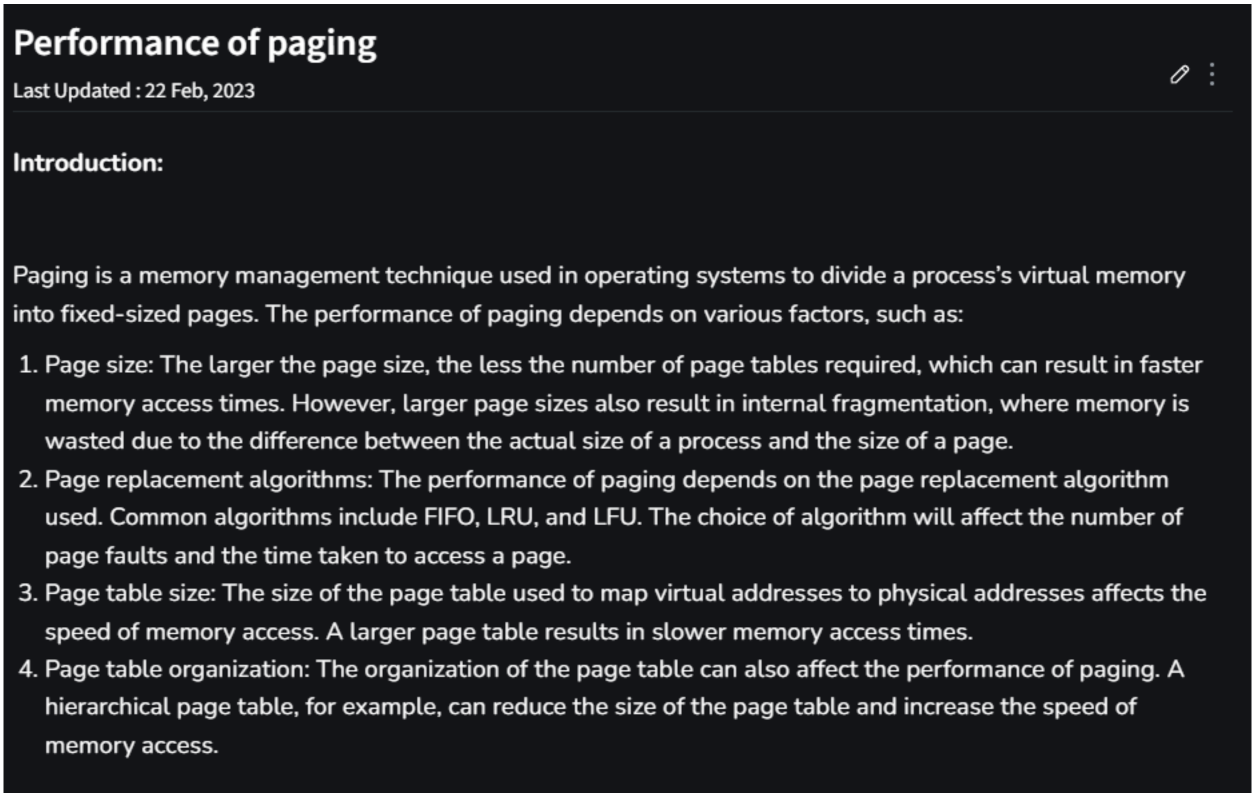
√ free frame find करो

· swap page into free frame

· reset table (page table) and change bit ⓥ

· read the instruction accur go to users

# ~~[illegible]~~ induction



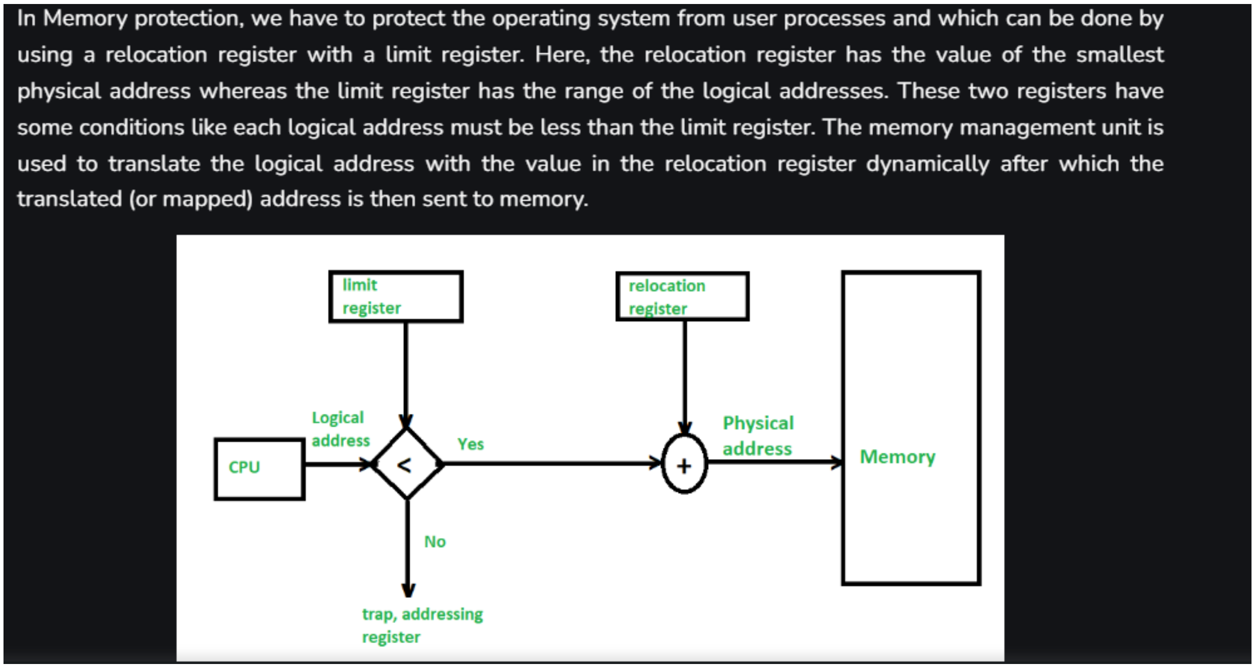
Performance of paging

Last Updated : 22 Feb, 2023

Introduction:

Paging is a memory management technique used in operating systems to divide a process's virtual memory into fixed-sized pages. The performance of paging depends on various factors, such as:

1. Page size: The larger the page size, the less the number of page tables required, which can result in faster memory access times. However, larger page sizes also result in internal fragmentation, where memory is wasted due to the difference between the actual size of a process and the size of a page.
2. Page replacement algorithms: The performance of paging depends on the page replacement algorithm used. Common algorithms include FIFO, LRU, and LFU. The choice of algorithm will affect the number of page faults and the time taken to access a page.
3. Page table size: The size of the page table used to map virtual addresses to physical addresses affects the speed of memory access. A larger page table results in slower memory access times.
4. Page table organization: The organization of the page table can also affect the performance of paging. A hierarchical page table, for example, can reduce the size of the page table and increase the speed of memory access.



In Memory protection, we have to protect the operating system from user processes and which can be done by using a relocation register with a limit register. Here, the relocation register has the value of the smallest physical address whereas the limit register has the range of the logical addresses. These two registers have some conditions like each logical address must be less than the limit register. The memory management unit is used to translate the logical address with the value in the relocation register dynamically after which the translated (or mapped) address is then sent to memory.

page replacement

Fifo

· যদি Main memory মধ্যে (না) থাকলে Page fault (x)

· " " " " " Hit ← মেমোরিতে আগে থেকে ছিল

String . 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0

frame set (3)

f3			1	1	1	~~1~~	0	0	~~0~~	3	3	3	~~3~~	2	2
f2		0	0	0	~~0~~	3	3	~~3~~	2	2	2	~~2~~	1	1	1
f1	7	7	~~7~~	2	2	2	~~2~~	4	4	~~4~~	0	0	0	0	0
	x	x	x	x	Hit	x	x	x	x	x	x	Hit	x	x	Hit

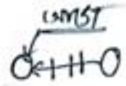
tepnique → যেটা আগে এসেছে f1, f2, then f3 যাবে

here hit = 3 → ratio = $\frac{3}{15} \times 100 = 20\%$

P-fault = 12 → ratio = $\frac{12}{15} \times 100 = 80\%$.

7 ## LRU 11

রেসেন্ট



যে recently আগে এসেছে

f4				2	2	2	2	2	2	~~2~~ 2	2	2	2		
f3			1	1	1	1	~~1~~	4	4	4	~~4~~	1	1		
f2		0	0	0	0	0	0	0	0	0	0	0	0		
f1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	3		
	·	·	·	×	H	×	H	H	H	H	H×	H	H		

String -. 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, ~~0 1 7 0 1~~

optimal

f3			1	1	1	4	4							
f2		0	0	0	0	0	0							
f1	7	7	~~7~~	2	2	2	2							

ভবিষ্যৎ হতে Last porjonto

String 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0 1, 7, 0, 1,

অনেক Last এ দূরে দিবে

সময় দেরি বেশি আছে

2nd Change Chance Algo (clock page)

~~Ref~~ Ref

- ~~ref~~ bit 1 [illegible] 2nd chance [illegible]
- fifo style.

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2.

5
2
5
5
5
0
1
2
change
[illegible]
5
[illegible]

ref bit 0 [illegible]
0
f1
f2
f3
2
2
0
2
1
0
3
0
3
0
0
M
M
H

G [illegible] change [illegible] and 2nd chance [illegible] and 1 → 0 [illegible] next 3 [illegible] 3 [illegible] [illegible] chg

	2	3	2	1	5	2	4	5	3	2	5	2
f1	2 0	2 0	2 1	2 1	2 0	2 1	2 0	2 0	2 0	3 0	3 0	3 0
f2	● 0	3 0	3 0	3 0	5 0	5 0	5 0	5 1	~~3~~5 0 1	5 0	5 1	5 1
f3	0		0	1 0	1 0	1 0	4 0	4 0	4 0	2 0	2 0	2
	h	M	H	M	M	H	M	H	M	M	H	

ref bit Same [illegible] 0 [illegible] [illegible]

1 [illegible] [illegible] 2 [illegible] Main Memory [illegible]

[Meory full [illegible] fifo [illegible] [illegible]]

[illegible] . ~~R~~ bit 1 [illegible] Tuplee [illegible] [illegible] [illegible]

No hits – 5 ; No of miss – 7.

$= \frac{5}{12} \times 100$ $= \frac{7}{12} \times 100$

[যদি 1 আবার ref bit (0) কে convert করি oo]
then cond বেশি ক্রেটর change.

Belady's anomaly in Fifo

Ref – 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

f3			3	3	~~3~~	2	2	2	2	~~2~~	4	4		
f2		2	2	~~2~~	1	1	1	1	~~1~~	3	3	3		
f1	~~1~~	1	~~1~~	4	4	~~4~~	5	5	5	5	5	5		
	×	×	×	×	×	×	×	H	H	×	×	H		

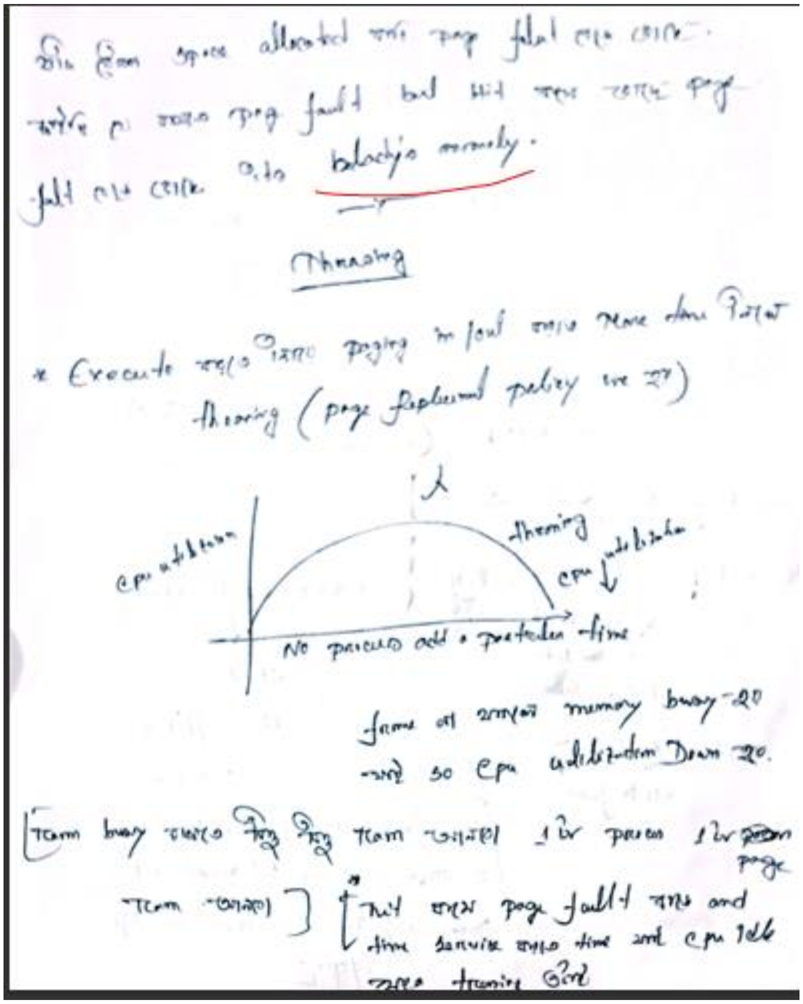
initially কোনো বেশি লোড

no Hit 3
page fault = 9.

এবার 4 page নিই

f4				4	4	4	4	4	~~4~~	3	3	3	
f3			3	~~3~~	3	3	3	~~3~~	2	2	2	2	
f2		2	2	2	2	2	~~2~~	1	1	1	~~1~~	5	
f1	1	1	1	~~1~~	1	~~1~~	5	5	5	~~5~~	4	4	
	×	×	×	×	H	H	×	×	×	×	×	×	

Hit = 2
Page fault – 10



ঠিক [illegible] space allocated [illegible] page fault [illegible] [illegible].
[illegible] [illegible] [illegible] page fault bad hit [illegible] [illegible] page
fault [illegible] [illegible] [illegible] belady's anomaly.

Thrashing

* Execute [illegible] [illegible] paging in fault [illegible] more time [illegible]
thrashing (page replacement policy use [illegible])

λ
cpu utilization
thrashing
cpu ↓ utilization
No process add · protection time

frame এর [illegible] memory busy হয়
[illegible] so cpu utilization Down হয়.

[Team busy [illegible] [illegible] [illegible] team [illegible] 1 টা process 1 টা page
team [illegible]] [[illegible] [illegible] page fault [illegible] and
time service [illegible] time and cpu idle
[illegible] thrashing [illegible]

handle thrasing

- Main memory size increased
- Long term sheduler use করে।
  - Degree of progm কম করে [আগে বলা]

remove করে Working set model.

→ নির্দিষ্ট সময় পর পর CU করে process কতগুলো frame use করেছে ⓸ বলে

working set model $\Delta t = 10$

$P_0 \rightarrow 1 \; 1\,2\,3\,1\,2\,3\,3\,2\,2\,4\,3\,2\,2\,1\,3\,3\,3\,1\,2\,2\,1\,7\,7\,1\,1\,7\,1\,2$

t_1 t_2

working model এর জন্য 10 টা unige value

$w_s t_1 = \langle 1,2,3,4 \rangle$
$= 4$ টা frame

$\Delta t = 6$ টা

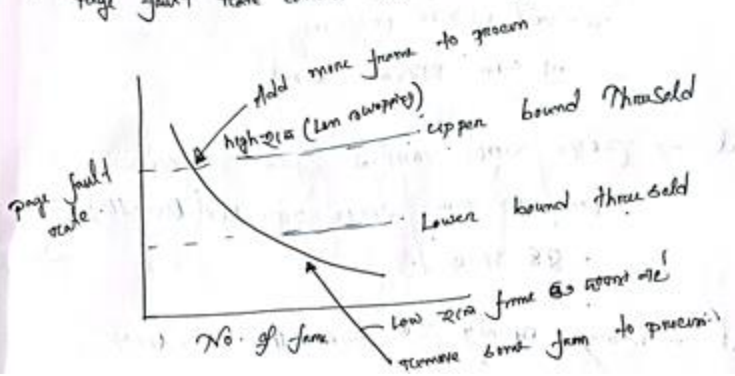
$w_s t_2 = (1,2,7)$
$= 3$ frame

2nd frame ⑤ টা আছে তার, 4, টা না- ই তখন working set model ai frame গুলো Process কে দিয়ে দিতে হয় [illegible]

Techniques of avoid thrasing

→ page fault frequency.

- Page fault rate control করলে thrasing handle হবে


Add more frame to process
high হলে (Less swapping)
upper bound Thresold
page fault rate
Lower bound threshold
low হলে frame এর দরকার নাই
remove some fram to process
No. of frame