

## UNIT-IV

Virtual m/m - It is a separation of user logical m/m from physical m/m.

In this method, we keep only a part of the user's point of view, process in the m/m and other part on the disk (secondary storage).

Advantages - i) - Only part of the prog. needs to be in m/m for execution.

ii) - Need to allow pages swap in & swap out when needed.

iii) - No Boundation for main m/m.

We need to maintain entire image of process in disk storage. It uses paging scheme.

User's point of view - (By developer)

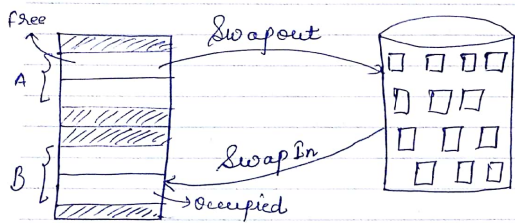
Programmers or developers are relieved of trying to fit a prog. into limited m/m.

System point of view - Reduces external fragmentation.

Reid & Taylor

Demand Paging - In this technique, a page is brought into the m/m for its execution only when it is demanded.

Combination of Paging and Swapping.  
It requires complete process in disk area in form of pages.  
Pages which is less needed is swapped like (Exception handling).



lazy swapper :- Because it swaps the page only when it is needed.

Advantages - \* Reduces m/m requirement  
\* Swap time is also reduced.  
\* Degree of multiprogramming ↑.

Disadvantages - \* It occurs page fault.  
(Whenever there is a requirement of page and that particular page not present in main m/m, is called page fault).

\* Page fault is used in performance measurement

Effective m/m access time  $\Rightarrow$

$$E_a = [ (P * \text{"page fault time"}) + (1-P) * m_a ]$$

$P$  = probability of P.F

$m_a$  = general m/m access time.

Page Replacement Algorithms :-

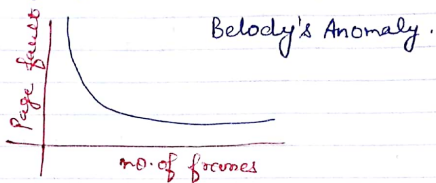
- ↳ It is the technique used by o.s to decide which m/m pages to swap out.
- It is also decided that in m/m, how many frames to allocate to each process.
- When page replacement is required, we must select the frames that are to be replaced.

Reference String - String of m/m references is called reference string.

7, 0, 1, 2, ...

Raid & Taylor

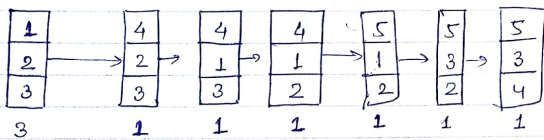
Graph of Page fault v/s. no. of frames.



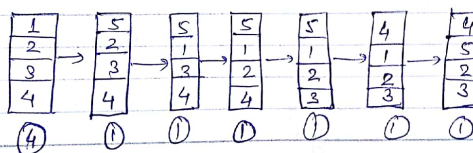
1) - First In First Out -

- ↳ very simple to implement.
- ↳ Oldest page is replaced for replacement.
- ↳ Performance is not always good.

ex- 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 (4 frames) (3 frames)



⇒ 9 page faults.



⇒ 10 page fault.

Belady's Anomaly.

Belady's Anomaly.

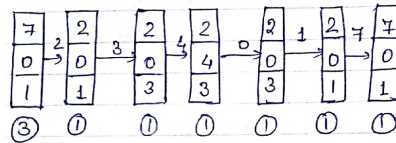
For some page replacement algorithm, page fault increases as the no. of allocated frames increases.

② - Optimal Page replacement - [Looks for the page in future]

- i) - Lowest Page fault
- ii) - Replace the page that will not be used for the longest period of time.
- iii) - Very difficult to implement.

Ex- 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

frame = 3.



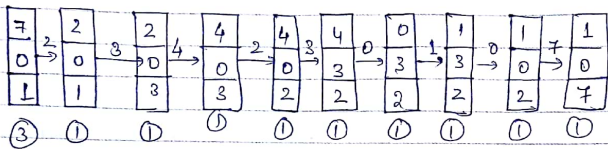
page fault ⇒ 9

③ - Least Recently Used Algo.

- i) - Page which has not been used for the longest time in main mem the one which will be selected for replacement.
- ii) - It is unlike Optimal page-replacement algo looking backwards in time.

Reid's Taylor

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



page fault  $\Rightarrow$  12

**Thrashing** - It is a situation, if the system spends more time in paging instead of their execution.

or

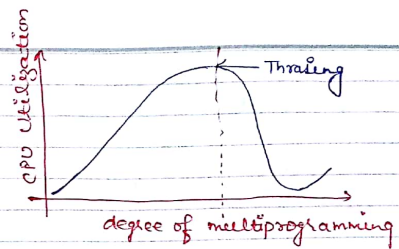
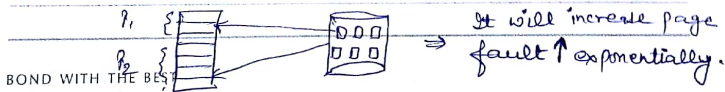
A process is busy swapping pages in and out.

\* Process does not have enough pages for the execution then the page fault will occur.

↳ Low utilization of CPU.

↳ O.S need to think that degree of multiprogramming should increase.

↳ Another process is added to the system.



Reid & Taylor