Page No.
Date

## Virtual memory

In order to increase the CPU utilisation, degree of multiprogramming should be increased, i.e., no. of processes in the main memory (MM) should be increased. Since, size of MM is limited, so, all the pages of the processes cannot be kept in the MM.
When CPU generates a logical address, it assumes that whole process is present in MM, i.e., all the pages of running process is are present in MM. However, practically it is not possible. So, concept of virtual memory takes place which works as extented RAM so that CPU feels that all pages of the running process are present in MM and generates the logical address according. Logical address is also called virtual address bcoz of virtual memory.

virtual memory space is ~~reserved~~ the space of the hard disk reserved for virtual memory in swapping. we need to separately allocate the space for VM bcoz ~~a data stored~~ ~~saved in hard disk is not same as that of RAM. So, a separate space is reserved for~~ the pattern in which data stored in RAM is not same as that of hard disk. So, a separate space of the hard disk is reserved for VM in place of directly using the hard disk space.

Two terms are associated with virtual memory implementation
(i) Frame allocation (ii) page replacement

In frame allocation, it is decided that how many frames in MM are allocated to be allocated to a process.

Let 4 frames of MM are allocated to the a processes which have accommodated 4 pages of the process. Now, if

5th page of the process is required by the CPU then in this case, one of the 4 frames will be evicted and 5th page will be fetched from hard disk and kept in the evicted frame. Here, page replacement policy decides that out of 4 frames, which frame will be evicted when the requirement of 5th page is generated by CPU.

## Frame Allocation

minimum no. of frames to be allocated to a process depends on the architecture of the processor or CPU. It actually depends on the maxi. no. of frames involved in the execution of an instruction. For example, in the worst case, if an instruction needs to refer 4 frames for its execution then mini. no. of frames must be allocated to that process is 4. ~~if those four pages are not~~ p. Whenever a page fault occurs, the instruction is restarted. Let mini. no. of frames needed to execute the instruction is 4 but frames allocated to a process is 3. In that case, page fault will occur and instruction will restart but it will always face a page fault but all pages needed by this instruction will never be in the mm.

maxi. no. of frames to be allocated to a process is size of the process.

So, frames to be allocated to a process should be between mini. and maxi. no. of frames to be allocated to a process.

(i) Equal allocation:- If no. of frames in mm is m and no. of processes are n then each process will be allocated m/n no. of frames.
However, equal allocation is not efficient will
but a smaller process not need $\frac{m}{n}$ frames whereas a bigger process will need more than $\frac{m}{n}$ frames.

(ii) Weighted allocation:- Frames are allocated depending on the size of the process.

Let, Size of Process $P_i$ is $S_i$ pages
no. of processes = n
total no. of frames = F
then no. of frames allocated to Process $P_i$ is

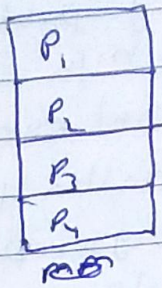
$$= \left(\frac{S_i}{S}\right) F$$

where $S = \sum_{i=1}^{n} S_i$

(iii) Dynamic allocation:- In this method, a fixed no. of frames are not allocated to a process. A process is allocated frames depending on its Priority and requirement. For example, initially Process $P_1$ and $P_2$ are allocated 2 frames each. Now, $P_1$ wants 2 more pages then if its Priority is higher than $P_2$ then frames of $P_2$ will be replaced by demanded pages of $P_1$.

## Page Replacement Algorithms

Let a process gets 4 frames in mm to get it executed, but it has 5 pages.




Let $P_1, P_2, P_3$ and $P_4, P_5$ are the pages of the process. Out of these 5 pages, $P_1, P_2, P_3, P_4$ have got frames in mm.

So, whenever $P_5$ will be needed by CPU, one of $P_1, P_2, P_3, P_4$ will have to be ~~swapped out~~ evicted so that $P_5$ ~~will be~~ can be accommodated. Now, which page out of $P_1, P_2, P_3, P_4$ should be replaced by $P_5$ will depend on the page replacement algorithm being used.

Generally Page replacement Algo (PRA) are of two types:-

(i) **local**:- In this, the page to be replaced will be among the pages of the running process which are in the mm. In this case, $P_1, P_2, P_3$ and $P_4$. Most of the PRAs are ~~local~~ based on local.

(ii) **global**:- In this case, the page to be replaced can be any frame in the mm, depending on various parameters such as priority of other process, percentage of allocated memory used by other processes, etc.

Page No.
Date

(a) Optimal PRA :- Replace the page which will not be refered longest.

(b) LRU (Least recently used) :- Replace the page which has not been refereed for a long time.

(c) FIFO :- First in First out :- Replace the oldest page.

In case of optimal, it is impossible to implement but it will give least no. of page faults.

Question: Following is the reference string

4 7 6 1 7 6 1 2 7 2

no. of frames allocated to the process = 3

Find the no. of page faults?

Here, reference string means that the sequence in which requirement of pages are generated. Here, numbers show the page number.

Optimal: Initially, no page is in mm. So, initial 3 requests will be considered as page fault.

Request for 4 → Not present (HP) in mm, so,
no. of page fault or (PF) = 1

| 4 | 1 |
|---|---|
| 7 | |
| 6 | |

Request for 7 - HP, PF = 2

Request for 6 - HP, PF = 3

" " 1 - HP, 4 will be replaced but it will not be refereed even after
PF = 4

Request for 7 - P.

Date

4761761272

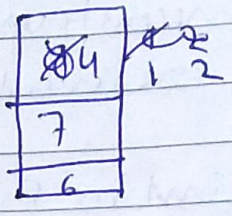
Request for 6 - P

Request for 1 - P

Request for 2 - MP, any among 1 and 6

Can be replaced

- PF = 5



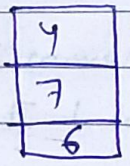

Request for 7 - P

Request for 2 - P

∴ total no. of PF = 5

LRU

for 4, 7, 6, PF = 3




Request for 1, MP,

4 will be replaced but it is least recently used.

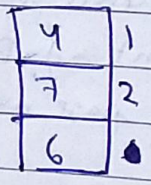
∴ PF = 4

Request for 7, 6, 1, P

Request for 2, MP,

7 will be replaced but it is LRU

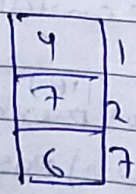
∴ PF = 5




Request for 7, MP

6 will be replaced but it is LRU

PF = 6




Request for 2, P

∴ total no. of page fault = 6

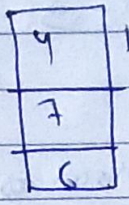
4 7 6 1 7 6 1 2 7 2

47 6 1 7 6 1 2 7 2

Page No. / Date

FIFO

for 4, 7, 6, PF = 3




Request for 1, MP

4 will be replaced bcz it came 1st

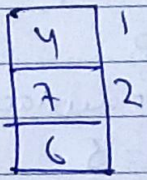
PF = 4

Request for 7, 6, 1, P,

Request for 2, MP

7 will be replaced bcz it came 1st

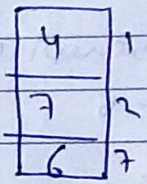
PF = 5




Request for 7, MP

6 will be replaced bcz it came 1st

PF = 6




Request for 3 2, P

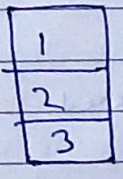
∴ total no. of PF = 6

Question: 1 2 3 2 4 1 3 2 4 1

total no of frames = 3

Optimal

~~if~~ for 1, 2, 3 no. of PF = 3

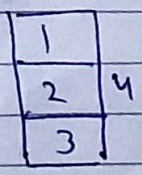



for 2, P

for 4, MP

2 will be replaced bcz it is not ~~needed~~ referred for longest time

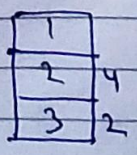
PF = 4




for 1, 3, P

for 2, MP, 3 will be replaced, PF = 5

for 4, 1 P

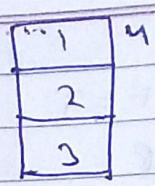


Page No.
Date

1 2 3 2 4 1 3 2 4 1

LRU

for 1, 2, 3, PF = 3



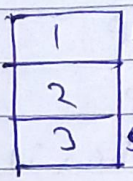

for 2, HP

for 4, HP

1 will be replaced but it is LRU

PF = 4

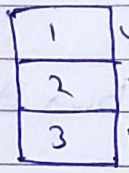
for 1, HP




3 will be replaced

PF = 5

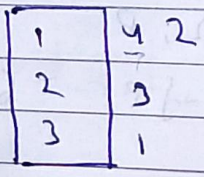
for 3, HP




2 will be replaced

PF = 6

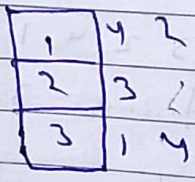
for 2, HP




4 will be replaced

PF = 7

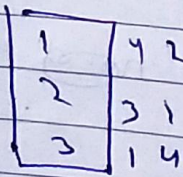
for 4, HP




1 will be replaced

PF = 8

for 1, HP




3 will be replaced

PF = 9

1 2 3 2 4 1 3 2 4 1

1 2 3 2 4 1 3 2 4 1

Page No. _____
Date

FIFO

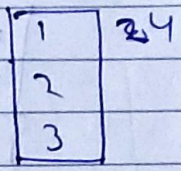
for 1, 2, 3 PF = 3


for 2, P

for 4, NP

1 will be replaced

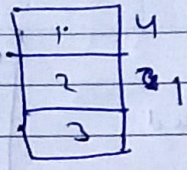
PF = 4




for 1, NP

2 will be replaced

PF = 5

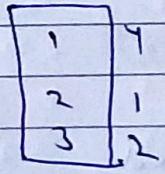



for 3 P

for 2 NP

3 will be replaced

PF = 6




for 4 P

for 1 P

∴ total no. of PF = 6

Date

## Belady's Anamoly

0 1 2 3 0 1 4 0 1 2 3 4

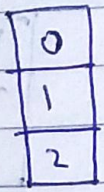
no. of frames = 3

optimal

For 0, 1, 2,

MP, ~~RF~~

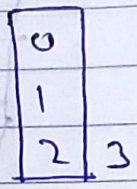
PF = 3




for 3, MP

2 will be replaced

PF = 4

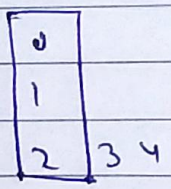



for 0, 1 P

for 4, MP

3 will be replaced

PF = 5



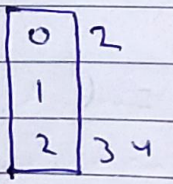

for 0 1 0 P

for 2 MP

either 0 or 1 can be replaced

~~ket~~ 0 is replaced

PF = 6




for 3, MP

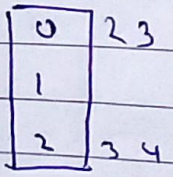
either 2 or 1 is replaced

PF = 7

~~for~~

for 4, P

∴ total no. of PF = 7

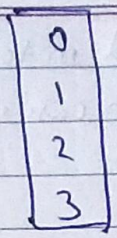


Page No.
Date

If no. of frames = 4

for 0, 1, 2, 3, MP

PF = 4

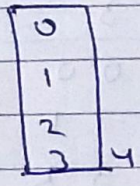



for 0, 1 P

for 4 MP

3 will be replaced

PF = 5

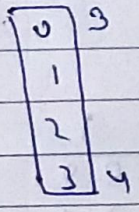



for 0, 1, 2 P

for 3, MP

any among 0 1 2 can be replaced

PF = 6




Here, we can see that as total no. of frames to be allocated to a process is increasing, Page faults are decreasing.

Similarly for LRU

In case of 3 frames, PF = 10

4 " PF = 8

In case of FIFO

for 3 frames PF = 9

4 frames, PF = 10

Through experiments it is found that in case of Optimal and LRU, as the no. of frames allocated to a process increases, Page fault always decreases. But in case of FIFO, it is not always true. This behaviour of FIFO is called Belady's anomaly.

Date

Analysis of 'optimal' in terms of presence of page numbers in mm in case of no. of frames allocated as 3 and 4.

Reference string

| 0 | 1 | 2 | 3 | 0 | 1 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 1 | 1 | 1 | 1 |
| | | 2 | 3 | 2 | 2 |
| | | | | 3 | 3 |

| 4 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 3 | 3 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 4 | 4 | 4 | 4 | 4 |
| | 4 | 4 | 4 | 4 | 4 |

Here, we can see that the pages numbers which are present in case of 3 frame nos. are always present in 4 frame no. also. This property is called as Stack property.

If an algo. follows the Stack property, it does not fall in Belady Anamoly.

Similarly, we can check it for FIFO

| 0 | 1 | 2 | 3 | 0 | 1 | 4 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 3 | 3 | 4 |
| 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 2 | 2 | 2 | 2 | 2 |
| | | 2 | 2 | 2 | 2 | 2 |
| | | | 3 | 3 | 3 | 3 |
| | | | | | | X |

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 2 | 1 | 1 | 2 |
| | 3 | 1 | 2 | 3 |
| | | 3 | 2 | 1 |
| | | | X | 2 |
| X | | | | |

In case of 1st request of 4,
Page no. 0 is present in 3 frames but not present in 4 frames. So, it is not following the Stack property.
So, it will suffer Belady's Anamoly.