PAGE REPLACEMENT ALGORITHMS:

* The Longest time period repeated pages are to be replaced first.

* page fault: The CPU will demand for the particular page number; If it is not present in the reference String (RAM/main memory). Then it is going to be Page Fault.

* Page Hit : The CPU will demand for the particular page number; If it is present in the RAM/main memory. Then it is going to be Page Hit.

TYPES: → FIFO : First In First Out. ⎫ with the help of
→ LRU : Least Recently Used. ⎬ this ⊗ , we can
→ Optimal Algorithm. ⎭ find page fault, Page Hit, Hit ratio.

FIFO:.

* First In First Out.

* From the name itself we can recognize that which page has to be replaced first and which page has to be Deleted first.

* Replace the pages that has been in the RAM/main memory for the longest time period.

* The Example as follows,

Reference String:

7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

* These page numbers are present in the RAM/memory

* Because the main memory size is lesser than the logical memory.

* The main memory can't store all the pages at a time.

* If the particular demanded page number not present in main memory. It has to be borrowed from secondary memory, by deleting the longest time repeated page number.

* If the page faults are more than RAM gets slower.   (left side).

7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

Page Frames	7	0	1	2	0	3	0	4	2	3	0	3	1	2	0
F1	⑦	7	7	②	2	2	2	④	4	4	0	0	0	0	0̲ Present
F2		⓪	0	0	0̲ Present	③	3	3	②	2	2	2	1	1	1
F3			①	1	1	1	⓪	0	0	③	3	3̲ Present	3	2	2
	*	*	*	*	Hit	*	*	*	*	*	*	Hit	*	*	Hit

Page Hit : Hits → 3

pag faults : * → 12

Reference string → 15.

Explanation for FIFO:

* Starting RAM / Main memory is free. It does not contains any page numbers.

* It contains (RAM) only three frames for Replacing the longest time period page numbers.

* The CPU is demanding for Page no : 7.. The RAM doesn't contains any page no in the starting. So CPU will borrow the particular page number from RAM. So page no. 7 is not there in RAM. Then it is going to be 'Page fault *'

* The process continuous like this.............

* Now the CPU is demanding for page '0'.

$$7 \quad 0 \quad 1 \quad 2 \quad \overset{\downarrow}{0} \quad 3 \quad 0 \quad 4 \quad 2 \quad 3 \quad 0 \quad 3 \quad 1 \quad 2 \quad 0$$

⇒ It is present in the given Reference String

Then it is going to be 'page Hit - Hit'.

Hit ratio = $\dfrac{\text{No. of Hits}}{\text{total no. of. Reference String}}$

$= \dfrac{3}{15} \times 100 = 20\%$ (approx)

Page fault = $\dfrac{\text{No. of fault}}{\text{Ref. string}}$

$= \dfrac{12}{15} \times 100 = 80\%$ (approx).

LRU:

* Least Recently Used.

* Note : 3 Frames = 3 page numbers. (compare)

* Left side page numbers are replaced with another.

Example :

(consider two zero's as 1)

(consider two 3's as one)

7	0	1	2	0	3	0	4	2	3	0	3	1	2	0

	7	7	(7)	2	2	2	(2)	4	4	(4)	0	0	(0)	2	2
F₁	7	7	(7)	2	2	2	(2)	4	4	(4)	0	0	(0)	2	2
F₂		0	0	0	0	0	0	(0)	3	3	3	3	(3)	0
F₃			1	1	(1)	3	3	(3)	2	2	2	(2)	1	1	1

| * | * | * | * | Hit | * | Hit | * | * | * | * | Hit | * | * | * |

* Step 1 : 7 0 1 (frame) = 7 0 1 (page no). check the Arrow [For your Reference].

* Step 2 : Compare the page numbers and frames in left direction.

* Don't compare the Frame F₂ for the particular time period.

* Check the frames with given page numbers.

<u>For example</u> : 7 0 1 (1 is most recently used. 7 is Least recently used. so select '7' and replace with next page number '2'.)

The process continuous . . . . . .

* Now, check the frame : 4 0 2 and page number (0 & 2). In frames 2 is very most recently used ; 4 is mostly recently used (only 2 times repeated. we should replace the frame with another page number only if particular page number repeats 3 & above times). so In this condition you just delete the frame (F₂) - '0'. with '3'.

The process Continuous · · · · · ·

page Hit = 3
page Fault = 12.

OPTIMAL ALGORITHM:

* This algorithm represents the longest time period page number in Future (Right Direction).

Example: longest time in future. →

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7
7	7	⑦	2	2	2	2	2	2	2	2	2	2	2	2	2	②	7
	0	0	0	0	0	⑩	4	4	④	0	0	0	0	0	0	0	0
		1	1	①	3	3	3	3	3	3	3	③	1	1	1	1	1
*	*	*	*	Hit	*	Hit	*	Hit	Hit	*	Hit	Hit	*	Hit	Hit	Hit	*

Explanation:

* The process starts as usual ;

* Now, check for longest time in future.

⇒ 7 0 1 (frames) ; comparing these numbers in given Reference string . 7 is most longest traveled page number (i.e., Repeated in future).

These processes are continues - - - - - - -

* Now, page number 4 is not there in future reference string. In this condition just replace the longest time repeated frame with another page number (i.e) '0' is replaced with 4.

* Then process repeats. After that 4 is not there in future reference string . So if it is in frames also no use. So just delete that page number and replace '0'.

2 ④ 3   replace with   2 0 3

* The process continuous. Last 7 is not there in future . So just replace with longest traveled page number (2) with 7.

Page Hit — 9
Page fault — 9.