Page fault

EAT (Effective access time) $= (1-p) \times$ mem. access time $+ p \times$ pg. fault overhead

overhead ↓
swap out
swap in
restarting inst

$0 < p < 1$ ← pg. fault for every reference

↑ no pg. fault probability of occurence ↑ pg. fault

$p = 0.6$ → fault → hard disk (ms)

$(1-p) = 0.4$ → hit → M.M (ns)

Eg) :- mem. access time = 200 ns
avg. pg. fault overhead = 8 ms
if 1 access out of 1000 causes pg. fault

$$p = \frac{1}{1000} = 0.001$$

$$EAT = (1 - 0.001) \times 200 \text{ ns} + (0.001) \times 8 \text{ ms}$$

$\times 10^{-9}$ $\times 10^{-3}$

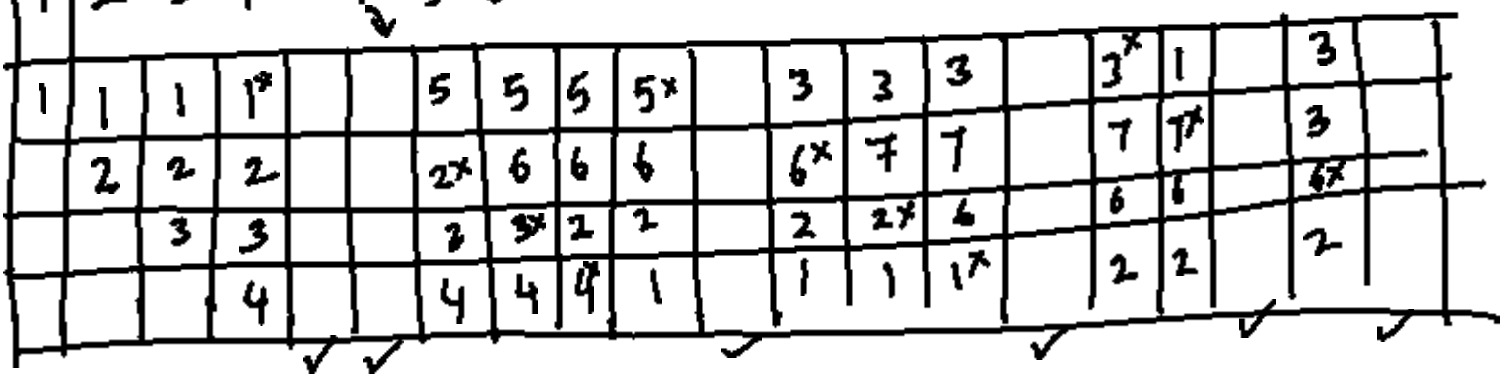
$$EAT = 82 \mu s$$

10^{-6}

FIFO — process – 8 pages – 4 frames

→ ref string

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6



Hits = 6

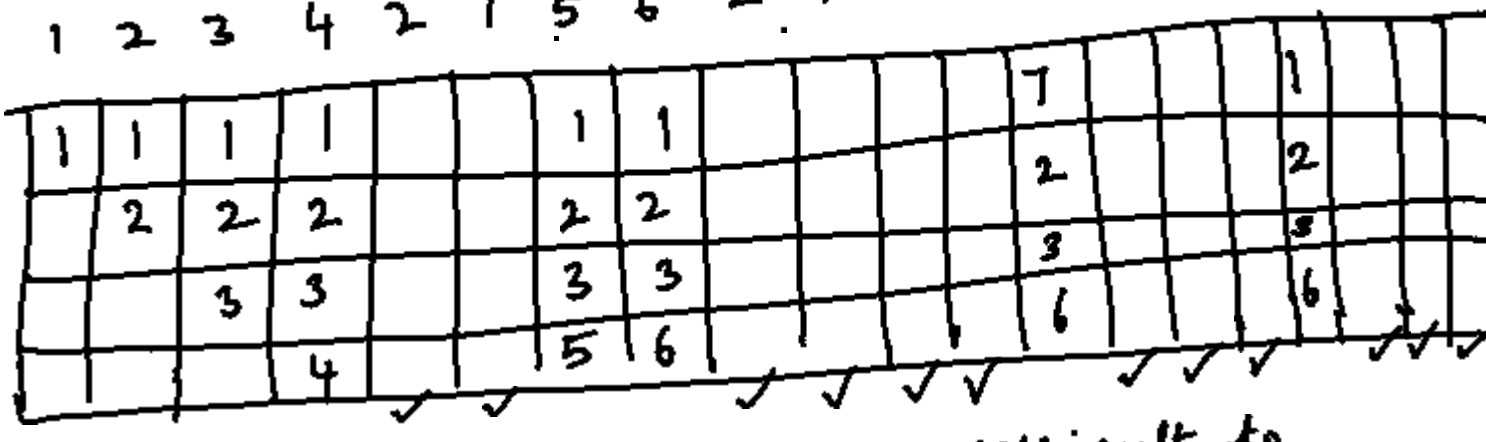
faults = 20 − 6 = (14)

Pg. fault ratio = $\frac{14}{20} \times 100 = 70\%$.

Optimal pg. replacement –

* future pg. references
* replace that page that is not used for the longest period of time

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6



Hits = 12

faults = 20 − 12 = (8)

Pg. fault ratio = $\frac{8}{20} \times 100 = 40$

* difficult to implement
* to predict

Least Recently used (LRU)

1	2	③	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1			1	1				1	1	6			6			
	2	2	2			2	2				2	2	2			2			
		3	3			5	5				3	3	3			3			
			4			4	6				6	7	7			1			
				✓	✓			✓	✓	✓				✓	✓		✓	✓	✓

hits = 10 = faults

pg. fault ratio = $\frac{10}{20} \times 100 = 50\%$

LRU < Stack, Counter

1	2	3	4	2	1	⑤	6	2	1	2	3	7	6	3	2	1	2	3	6
1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1			
	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2			
		1	2	3	4	2	1	5	6	6	1	2	3	7	6	3			
			1	1	3	4	2	1	5	5	6	1	2	2	7	6			
				✓	✓			✓	✓	✓				✓	✓		✓	✓	✓

hits = 10 = faults

counter 2 bit - 0, 1, 2, 3

counters —

Pg.	counter
1	~~1~~ ~~2~~ ~~3~~ ~~4~~ 7
2	~~1~~ ~~2~~ 3 4
3	~~1~~ ~~2~~ 3
4	1 2
5	1