Operating System Page Replacement Algorithms

Page Replacement

Page replacement takes the following approach. If no frame is free, we find one that is not currently being used and free it.

1. Find the location of the desired page on the disk.

2. Find a free frame:
    a. If there is a free frame, use it.
    b. If there is no free frame, use a page-replacement algorithm to select a victim frame.
    c. Write the victim frame to the disk; change the page and frame tables accordingly.

3. Read the desired page into the newly freed frame; change the page and frame tables.

4. Continue the user process from where the page fault occurred.

Page replacement is basic to demand paging. It completes the separation between logical memory and physical memory. With this mechanism, an enormous virtual memory can be provided for programmers on a smaller physical memory.

There are many different page-replacement algorithms. Every operating system probably has its own replacement scheme. How do we select a particular replacement algorithm? In general, we want the one with the lowest page-fault rate.

We evaluate an algorithm by running it on a particular string of memory references and computing the number of page faults. The string of memory references is called a reference string.

FIFO Page Replacement

The simplest page-replacement algorithm is a first-in, first-out (FIFO) algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen.

The FIFO page-replacement algorithm is easy to understand and program. However, its performance is not always good. On the one hand, the page replaced may be an initialization module that was used a long time ago and is no longer needed. On the other hand, it could contain a heavily used variable that was initialized early and is in constant use.

Even if we select for replacement a page that is in active use, everything still works correctly. After we replace an active page with a new one, a fault occurs almost immediately to retrieve the active page. Some other page must be replaced to bring the active page back into memory. Thus, a bad replacement choice increases the page-fault rate and slows process execution. It does not, however, cause incorrect execution.

Belady’s anomaly: for some page-replacement algorithms, the page-fault rate may increase as the number of allocated frames increases.

Optimal Page Replacement

One result of the discovery of Belady’s anomaly was the search for an optimal page-replacement algorithm—the algorithm that has the lowest page-fault rate of all algorithms and will never suffer from Belady’s anomaly. Such an algorithm does exist and has been called OPT or MIN .

It is simply this: Replace the page that will not be used for the longest period of time.

Use of this page-replacement algorithm guarantees the lowest possible page fault rate for a fixed number of frames.

LRU Page Replacement

If the optimal algorithm is not feasible, perhaps an approximation of the optimal algorithm is possible. The key distinction between the FIFO and OPT algorithms (other than looking backward versus forward in time) is that the FIFO algorithm uses the time when a page was brought into memory, whereas the OPT algorithm uses the time when a page is to be used. If we use the recent past as an approximation of the near future, then we can replace the page that has not been used for the longest period of time. This approach is the least recently used (LRU) algorithm.

LRU replacement associates with each page the time of that page’s last use. When a page must be replaced, LRU chooses the page that has not been used for the longest period of time.

The LRU policy is often used as a page-replacement algorithm and is considered to be good. The major problem is how to implement LRU replacement. An LRU page-replacement algorithm may require substantial hardware assistance. The problem is to determine an order for the frames defined by the time of last use.

Input 1 :

Number of References : 20
Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Number of frames : 3

Output :

```
nirish@16102008:~$ cd Desktop
nirish@16102008:~/Desktop$ gcc Page_Replacement.c
nirish@16102008:~/Desktop$ ./a.out
Enter the number of References : 20
Enter the Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Enter the number of frames : 3

First In First Out (FIFO) Page Replacement Policy :

7       -       -
7       0       -
7       0       1
2       0       1
No Page Fault
2       3       1
2       3       0
```

4 3 0
4 2 0
4 2 3
0 2 3
No Page Fault
No Page Fault
0 1 3
0 1 2
No Page Fault
No Page Fault
7 1 2
7 0 2
7 0 1

Analysis of First In First Out (FIFO) Page Replacement Policy for the given Reference String and Number of Frames :

No of Total References : 20
No of Page Hits : 5
No of Page Faults : 15
Hit Percentage : 25.000000

Optimal Page Replacement Policy :

7 - -
7 0 -
7 0 1
2 0 1
No Page Fault
2 0 3
No Page Fault
2 4 3
No Page Fault
No Page Fault
2 0 3
No Page Fault
No Page Fault
2 0 1
No Page Fault
No Page Fault
No Page Fault
7 0 1
No Page Fault
No Page Fault

Analysis of Optimal Page Replacement Policy for the given Reference String and Number of Frames :

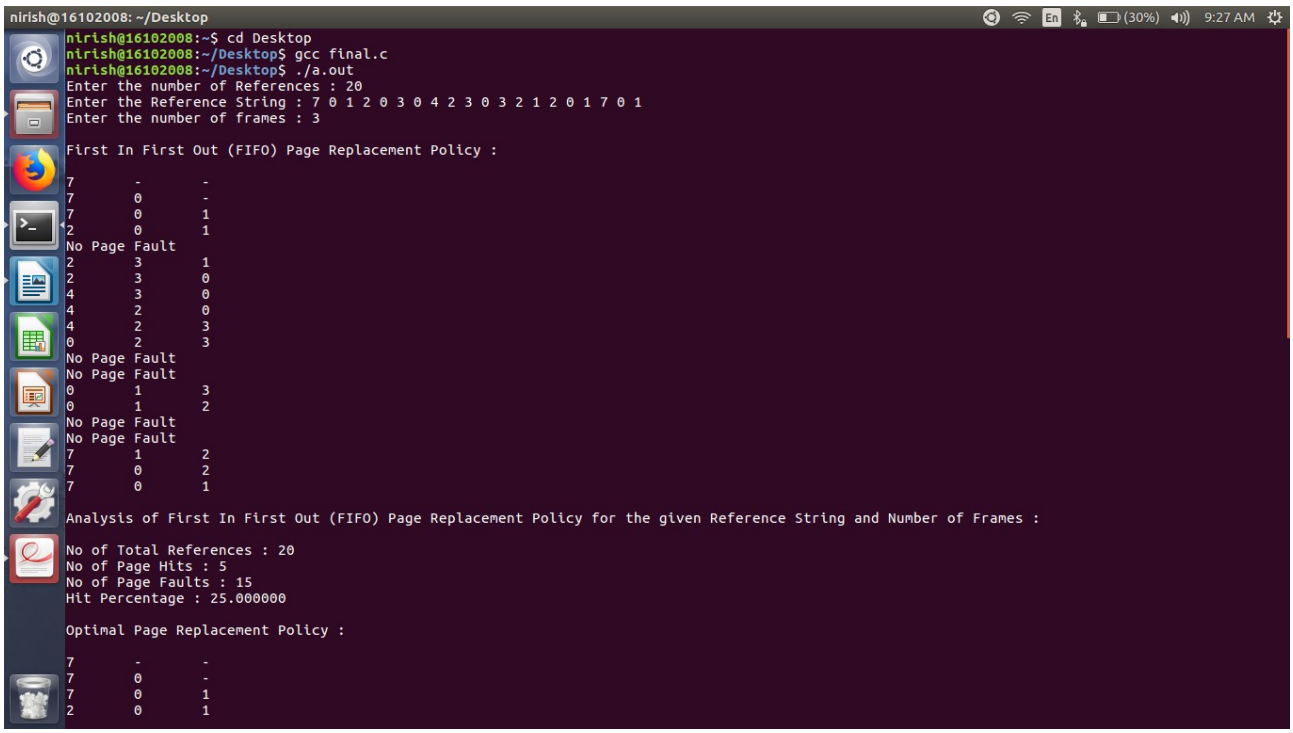
No of Total References : 20
No of Page Hits : 11
No of Page Faults : 9
Hit Percentage : 55.000000

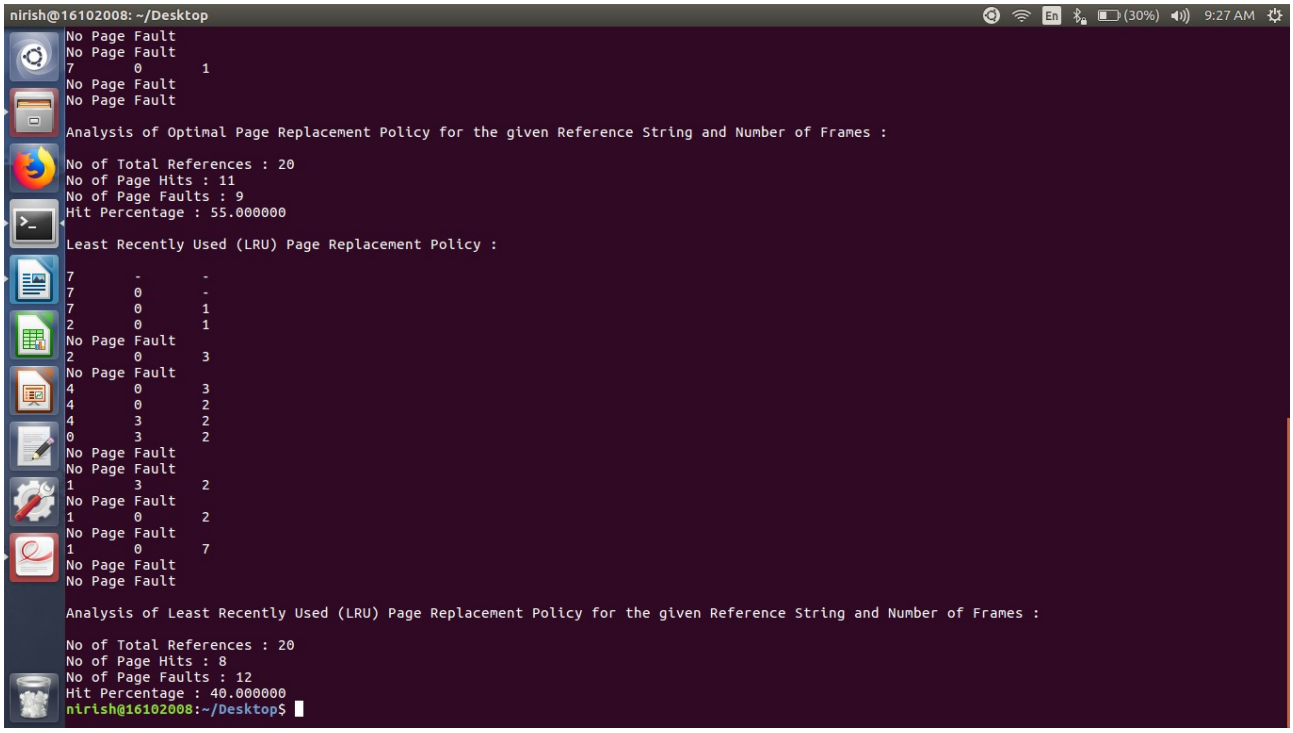
Least Recently Used (LRU) Page Replacement Policy :

```
7       -       -
7       0       -
7       0       1
2       0       1
No Page Fault
2       0       3
No Page Fault
4       0       3
4       0       2
4       3       2
0       3       2
No Page Fault
No Page Fault
1       3       2
No Page Fault
1       0       2
No Page Fault
1       0       7
No Page Fault
No Page Fault
```

Analysis of Least Recently Used (LRU) Page Replacement Policy for the given Reference String and Number of Frames :

```
No of Total References : 20
No of Page Hits : 8
No of Page Faults : 12
Hit Percentage : 40.000000
nirish@16102008:~/Desktop$
```





Input 2 :

Number of References : 20
Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Number of frames : 4

Output :

nirish@16102008:~$ cd Desktop

```
nirish@16102008:~/Desktop$ gcc final.c
nirish@16102008:~/Desktop$ ./a.out
Enter the number of References : 20
Enter the Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Enter the number of frames : 4

First In First Out (FIFO) Page Replacement Policy :

7       -       -       -
7       0       -       -
7       0       1       -
7       0       1       2
No Page Fault
3       0       1       2
No Page Fault
3       4       1       2
No Page Fault
No Page Fault
3       4       0       2
No Page Fault
No Page Fault
3       4       0       1
2       4       0       1
No Page Fault
No Page Fault
2       7       0       1
No Page Fault
No Page Fault

Analysis of First In First Out (FIFO) Page Replacement Policy for the given Reference String and Number of Frames :

No of Total References : 20
No of Page Hits : 10
No of Page Faults : 10
Hit Percentage : 50.000000

Optimal Page Replacement Policy :

7       -       -       -
7       0       -       -
7       0       1       -
7       0       1       2
No Page Fault
3       0       1       2
No Page Fault
3       0       4       2
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
```

1 0 4 2
No Page Fault
No Page Fault
No Page Fault
1 0 7 2
No Page Fault
No Page Fault

Analysis of Optimal Page Replacement Policy for the given Reference String and Number of Frames :

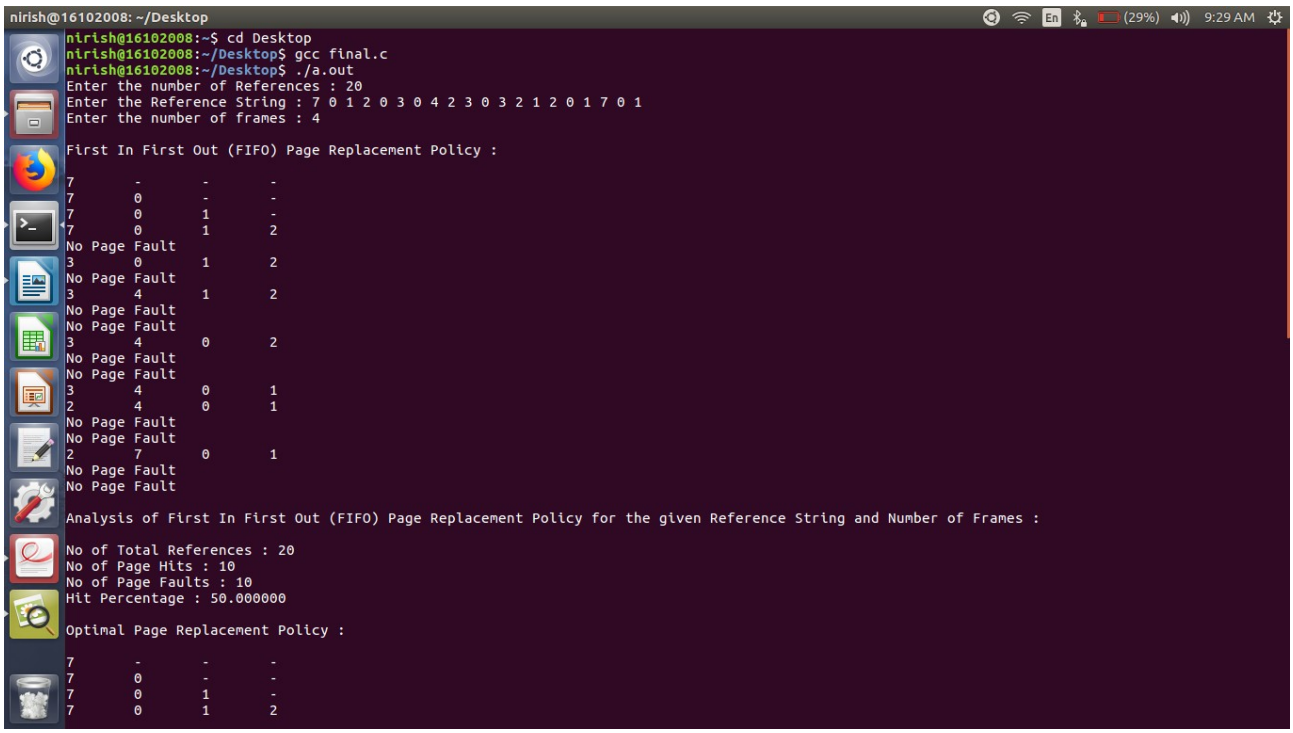
No of Total References : 20
No of Page Hits : 12
No of Page Faults : 8
Hit Percentage : 60.000004

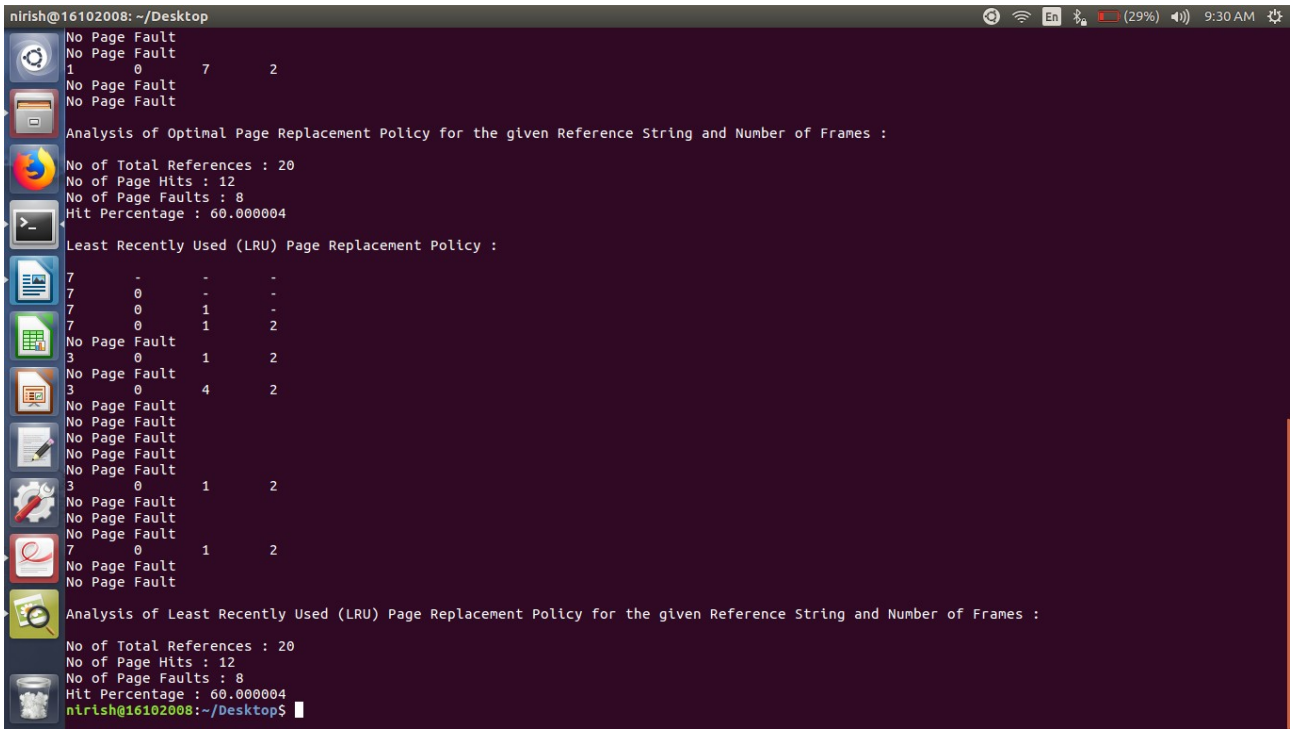
Least Recently Used (LRU) Page Replacement Policy :

7 - - -
7 0 - -
7 0 1 -
7 0 1 2
No Page Fault
3 0 1 2
No Page Fault
3 0 4 2
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
3 0 1 2
No Page Fault
No Page Fault
No Page Fault
7 0 1 2
No Page Fault
No Page Fault

Analysis of Least Recently Used (LRU) Page Replacement Policy for the given Reference String and Number of Frames :

No of Total References : 20
No of Page Hits : 12
No of Page Faults : 8
Hit Percentage : 60.000004
nirish@16102008:~/Desktop$





Input 3 Test Case :

Number of References : 20
Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Number of frames : 5

Output :

nirish@16102008:~$ cd Desktop
nirish@16102008:~/Desktop$ gcc final.c

```
nirish@16102008:~/Desktop$ ./a.out
Enter the number of References : 20
Enter the Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Enter the number of frames : 5

First In First Out (FIFO) Page Replacement Policy :

7       -       -       -       -
7       0       -       -       -
7       0       1       -       -
7       0       1       2       -
No Page Fault
7       0       1       2       3
No Page Fault
4       0       1       2       3
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
4       7       1       2       3
4       7       0       2       3
4       7       0       1       3

Analysis of First In First Out (FIFO) Page Replacement Policy for the given Reference String and
Number of Frames :

No of Total References : 20
No of Page Hits : 11
No of Page Faults : 9
Hit Percentage : 55.000000

Optimal Page Replacement Policy :

7       -       -       -       -
7       0       -       -       -
7       0       1       -       -
7       0       1       2       -
No Page Fault
7       0       1       2       3
No Page Fault
4       0       1       2       3
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
```

No Page Fault
No Page Fault
No Page Fault
7 0 1 2 3
No Page Fault
No Page Fault

Analysis of Optimal Page Replacement Policy for the given Reference String and Number of Frames :

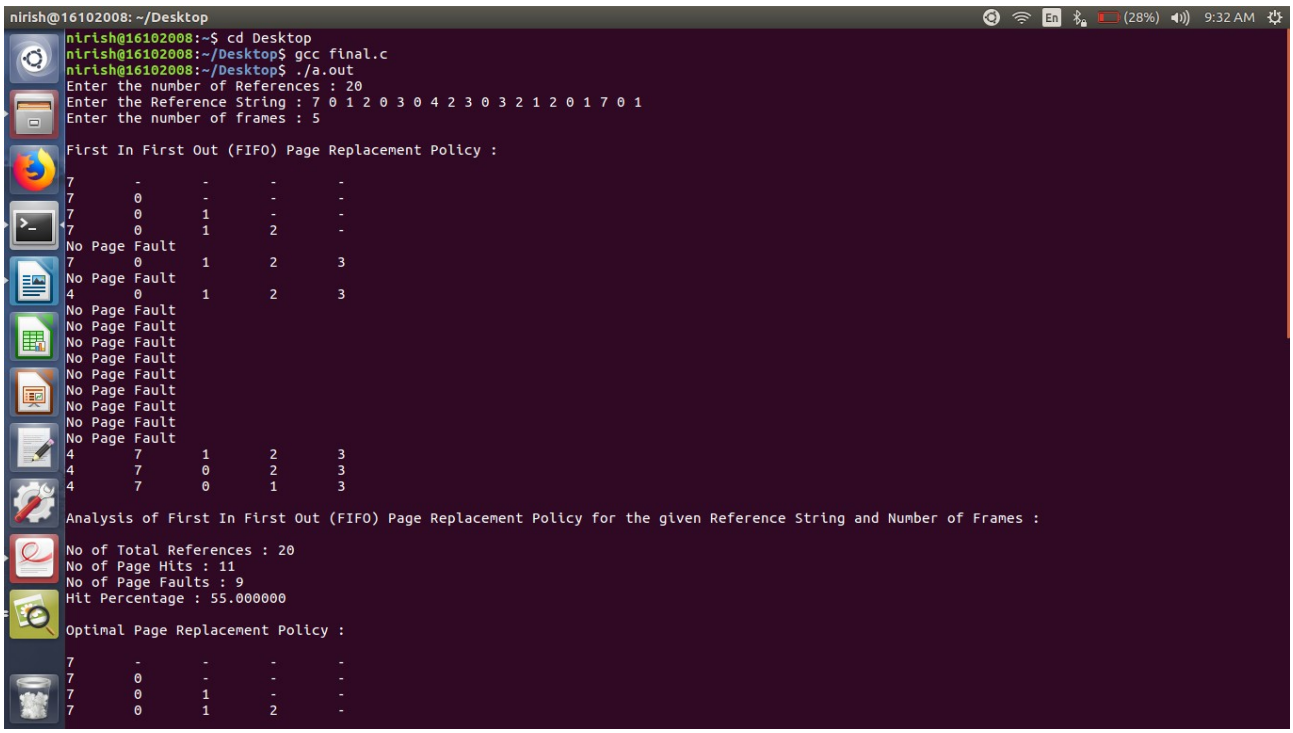
No of Total References : 20
No of Page Hits : 13
No of Page Faults : 7
Hit Percentage : 65.000000

Least Recently Used (LRU) Page Replacement Policy :

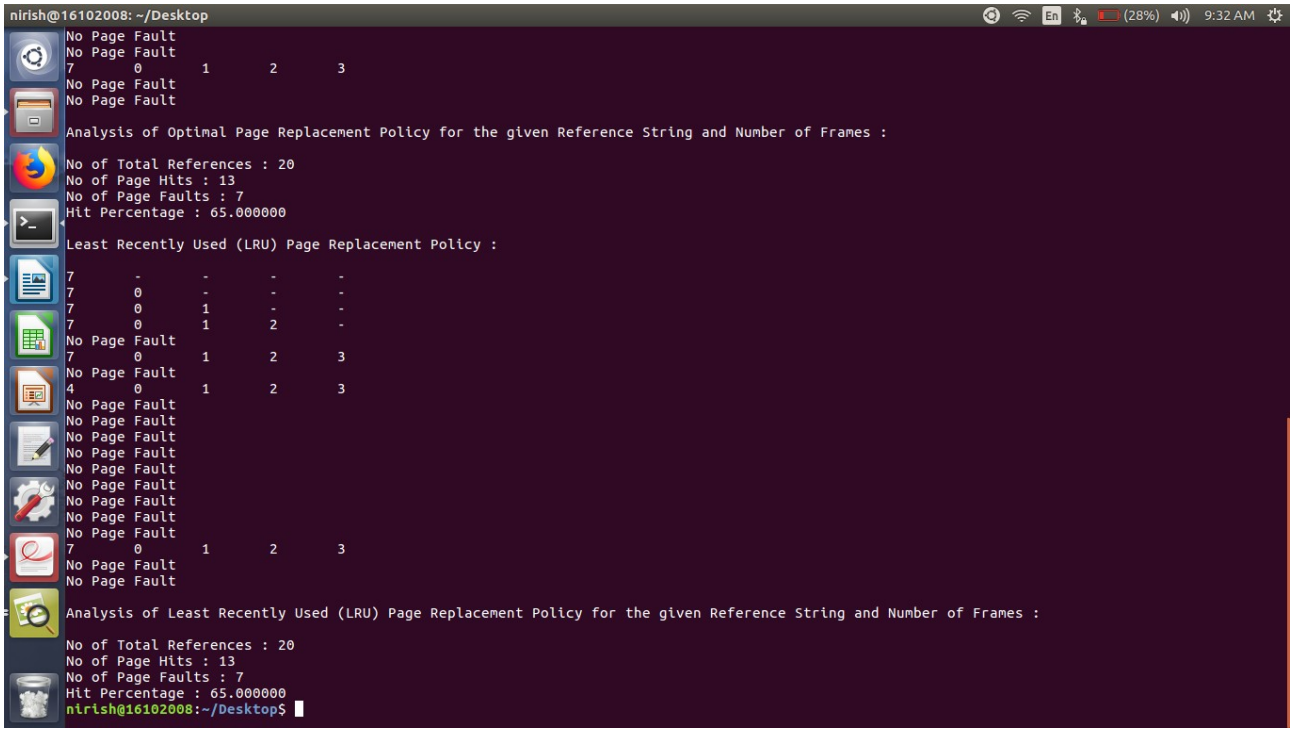
7 - - - -
7 0 - - -
7 0 1 - -
7 0 1 2 -
No Page Fault
7 0 1 2 3
No Page Fault
4 0 1 2 3
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
No Page Fault
7 0 1 2 3
No Page Fault
No Page Fault

Analysis of Least Recently Used (LRU) Page Replacement Policy for the given Reference String and Number of Frames :

No of Total References : 20
No of Page Hits : 13
No of Page Faults : 7
Hit Percentage : 65.000000
nirish@16102008:~/Desktop$



Soving on Paper :

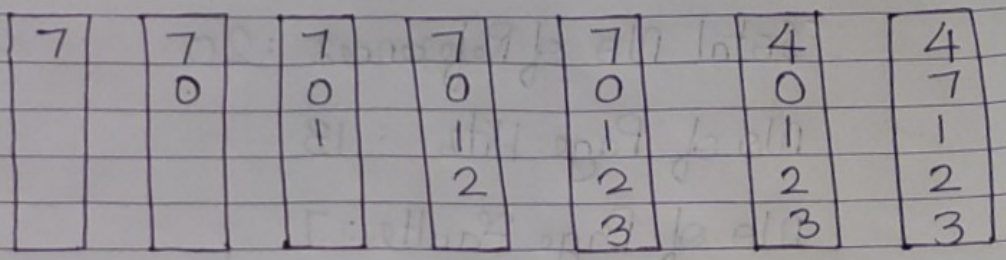


- No of memory Pages : 5

Reference String : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

1) First In First Out (FIFO) Page Replacement Policy

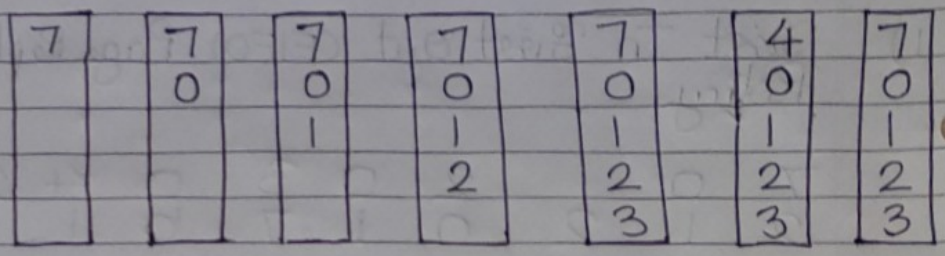
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1



No of References = 20
No of Hits = 11
No of Page Faults = 9

2) Optimal Page Replacement Policy :

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



Total No of References : 20

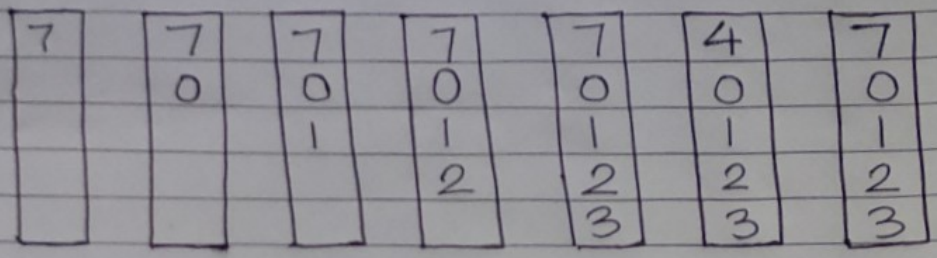
No of Page Hits : 13

No of Page Faults : 7.

3) Least Recently Used (LRU) Page Replacement Policy

Reference String

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



Total No of References : 20

No of Page Hits : 13

No of Page Faults : 7

Conclusion :

Thus , we have successfully studied and implemented Page Replacement Policy Mini Project.