Ex. No.: 11a)
Date: 12/4/25

FIFO PAGE REPLACEMENT

Aim:

To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

Algorithm:

1. Declare the size with respect to page length
2. Check the need of replacement from the page to memory
3. Check the need of replacement from old page to new page in memory 4. Form a queue to hold all pages
5. Insert the page require memory into the queue
6. Check for bad replacement and page fault
7. Get the number of processes to be inserted
8. Display the values

Program Code:

```
n = int(input("Enter no. of frames"))
s = [ ]
n1 = int(input("Enter size of reference string"))
s1 = [ ]
top = 0
for i in range(len(s)):
    if s[i] not in s1:
        if(len(s1) < n):
            s1.append(s[i])
            print(s[i], "->", s1, "Page Fault")
            top += 1
        else:
            s1[(top)%n] = s[i]
            print(s[i], "->", s1, "Page Fault")
            top += 1
    else:
        print("No Page Fault")
```


PDF Scanner ACE Scanner

Sample Output:

```
[root@localhost student]# python fifo.py

Enter the size of reference string: 20
Enter [ 1] : 7
Enter [ 2] : 0
Enter [ 3] : 1
Enter [ 4] : 2
Enter [ 5] : 0
Enter [ 6] : 3
Enter [ 7] : 0
Enter [ 8] : 4
Enter [ 9] : 2
Enter [10] : 3
Enter [11] : 0
Enter [12] : 3
Enter [13] : 2
Enter [14] : 1
Enter [15] : 2
Enter [16] : 0
Enter [17] : 1
Enter [18] : 7
Enter [19] : 0
Enter [20] : 1

Enter page frame size : 3

7 -> 7 - -
0 -> 7 0 -
1 -> 7 0 1
2 -> 2 0 1
0 -> No Page Fault


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


ACE Scanner

```
1 -> 7 0 1
Total page faults: 15.
[root@localhost student]#
```

Result :

Thus the program to implement solution the number of page faults that Occur using First-in-First Out page ~~how~~ replacement technique has been executed successfully.

Ex. No.: 11b)
Date: 12/4/25

LRU

Aim:

To write a c program to implement LRU page replacement algorithm.

Algorithm:

1: Start the process
2: Declare the size
3: Get the number of pages to be inserted
4: Get the value
5: Declare counter and stack
6: Select the least recently used page by counter value
7: Stack them according the selection.
8: Display the values
9: Stop the process

Program Code:

```
n=int(input("Enter no. of frames"))
n1=int(input("Enter size of reference string"))
s=[]
for i in range(n1):
    s.append(int(input("Enter the Page No:")))
s1=[]
for i in range(len(s)):
    if s[i] not in s1:
        if(len(s1)<n):
            s1.append(s[i])
            print(s[i],"->",s1,"Page Fault")
        else:
            temp[]
            for j in range(i-1,-1,-1):
                if(s[j] not in temp and len(temp)<n:
                    temp.append(s[j])
                elif len(temp)>n-1:
                    break
            s[s1.index(temp[-1])] = s[i]
            print(s[i],"->",s1,"Page Fault")
else:
    print("No Page Fault")
```

Enter no. of frames 3
Enter size of reference string 20
Enter the Page No. 7
Enter the Page No. 0
Enter the Page No. 1
Enter the Page No. 2
Enter the Page No. 0
Enter the Page No. 3
Enter the Page No. 0
Enter the Page No. 4
Enter the Page No. 2
Enter the Page No. 3
Enter the Page No. 0
Enter the Page No. 3
Enter the Page No. 2
Enter the Page No. 1
Enter the Page No. 2
Enter the Page No. 0
Enter the Page No. 1
Enter the Page No. 7
Enter the Page No. 0
Enter the Page No. 1

7 -> [7] Page Fault
0 -> [7, 0] Page Fault
1 -> [7, 0, 1] Page Fault
2 -> [2, 0, 1] Page Fault
No Page Fault
3 -> [2, 0 3] Page Fault
No Page Fault
4 -> [4, 0, 3] Page Fault
2 -> [4, 0, 2] Page Fault
3 -> [4, 3, 2] Page Fault
0 -> [0, 3, 2] Page Fault

PDF Scanner ACE Scanner

No Page Fault

No Page Fault

1 -> [1, 3, 2] Page Fault

No Page Fault

0 -> [1, 0, 2] Page Fault

No Page Fault

7 -> [1, 0, 7] Page Fault

No Page Fault

No Page Fault

PDF Scanner ACE Scanner

Sample Output :

Enter number of frames: 3
Enter number of pages: 6
Enter reference string: 5 7 5 6 7 3
5 -1 -1
5 7 -1
5 7 -1
5 7 6
5 7 6
3 7 6
Total Page Faults = 4

Result:

Thus. the program to Implement solution ~~to~~ of LRU page replacement algorithm technique has been executed successfully.

ACE Scanner

Ex. No.: 11c)
Date: 18/4/25

Optimal

Aim:
To write a c program to implement Optimal page replacement algorithm.

ALGORITHM:

1. Start the process
2. Declare the size
3. Get the number of pages to be inserted
4. Get the value
5. Declare counter and stack
6. Select the least frequently used page by counter value
7. Stack them according the selection.
8. Display the values
9. Stop the process

PROGRAM:

```
n=int(input("Enter no. of frames"))
s=[]
n1=int(input("Enter the size of the reference string"))
for i in range(n1):
    s.append(int(input("Enter the PageNo.")))

s1=[]
for i in range(len(s)):
    if s[i] not in s1:
        if(len(s1)<n):
            s1.append(s[i])
            print(s[i],"->",s1,"Page Fault")
        else:
            temp=[]
            for j in range(len(s1)):
                try:
                    temp.append(s[i:].index(s1[j]))
                except:
                    temp.append(-1)
                    break
```


PDF Scanner ACE Scanner

Enter No. of frames 3
Enter the size of reference string 20
Enter the Page No. 7
Enter the Page No. 0
Enter the Page No. 1
Enter the Page No. 2
Enter the Page No. 0
Enter the Page No 3
Enter the Page No. 0
Enter the Page No. 4
Enter the Page No. 2
Enter the Page No. 3
Enter the Page No. 0
Enter the Page No. 3
Enter the Page No. 2
Enter the Page No. 1
Enter the Page No. 2
Enter the Page No. 0
Enter the Page No. 1
Enter the Page No. 7
Enter the Page No. 0
Enter the Page No. 1

7 -> [7] Page Fault
0 -> [7, 0] Page Fault
1 -> [7, 0, 1] Page Fault
2 -> [2, 0, 1] Page Fault
No Page Fault
3 -> [2, 0, 3] Page Fault
No Page Fault
4 -> [2, 4, 3] Page Fault
No Page Fault
No Page Fault
0 -> [2, 0, 3] Page Fault
No Page Fault
No Page Fault

PDF Scanner ACE Scanner

1 -> [2, 0, 1] Page Fault

No Page Fault

No Page Fault

No Page Fault

7 -> [7, 0, 1] Page Fault

No Page Fault

No Page Fault

PDF Scanner ACE Scanner

```
    if -1 in temp
        a = temp.index(-1)
    else:
        a = temp.index(max(temp))

    s1[a] = s[i]
    print(s[i], "->", s1, "Page Fault")
else:
    print("No Page Fault")
```


PDF Scanner ACE Scanner

Output:

Result:

Thus the program to implement solution of Optimal page replacement algorithm. technique. has been executed sucessfully.

PDF Scanner ACE Scanner