Experiment No. 9

9. Implement e all page replacement algorithms:

FIFO

LRU

LFU

DESCRIPTION

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.

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.

If the recent past is used as an approximation of the near future, then the page that has not been used for the longest period of time can be replaced. 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.

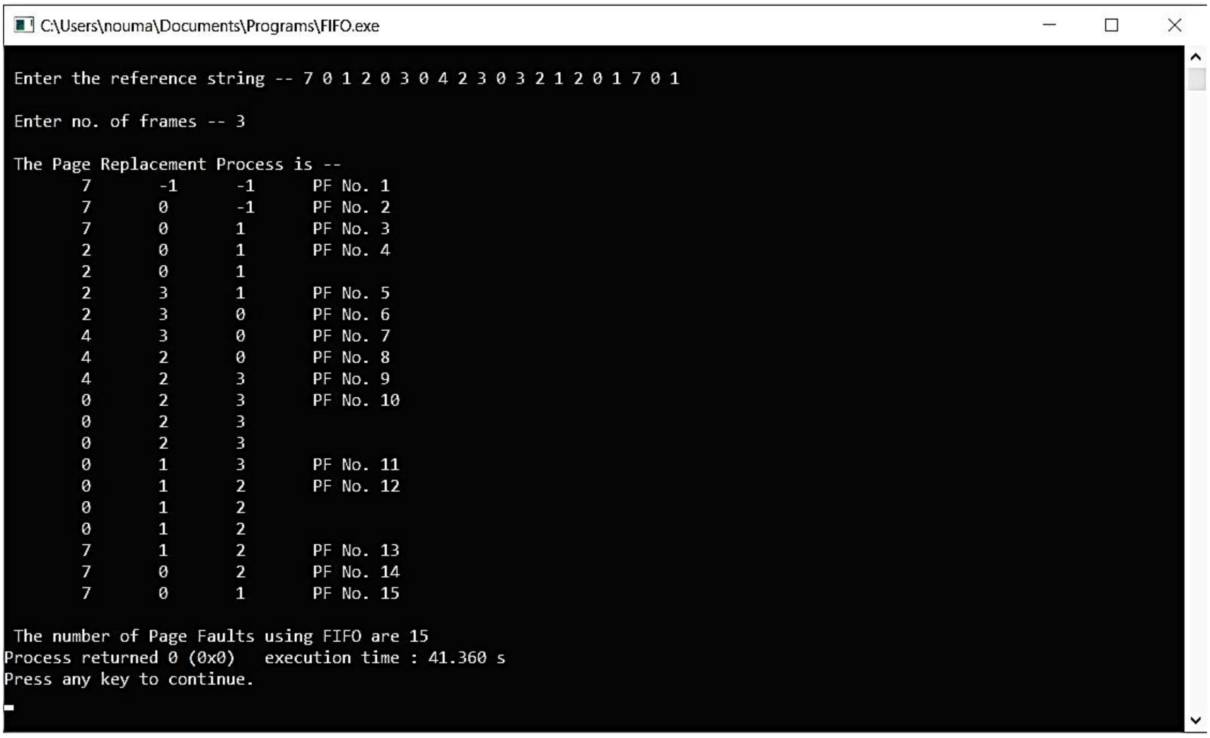
Least frequently used (LFU) page-replacement algorithm requires that the page with the smallest count be replaced. The reason for this selection is that an actively used page should have a large reference count.

FIFO

Program:

```
#include<stdio.h>

main()
{
   int i, j, k, f, pf=0, count=0, rs[25], m[10], n;
   printf("\n Enter the length of reference string -- ");
   scanf("%d",&n);
   printf("\n Enter the reference string -- ");

   for(i=0;i<n;i++)
      scanf("%d",&rs[i]);

   printf("\n Enter no. of frames -- ");
   scanf("%d",&f);

   for(i=0;i<f;i++)
      m[i]=-1;
   printf("\n The Page Replacement Process is -- \n");

   for(i=0;i<n;i++)
      {
         for(k=0;k<f;k++)
         {
            if(m[k]==rs[i])
            break;
         }
         if(k==f)
            {
               m[count++]=rs[i];
               pf++;
            }
         for(j=0;j<f;j++)
            printf("\t%d",m[j]);
         if(k==f)
            printf("\tPF No. %d",pf);
         printf("\n");
         if(count==f)
            count=0;
         }
   printf("\n The number of Page Faults using FIFO are %d",pf);
}
```

Output:



```
 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 no. of frames -- 3

 The Page Replacement Process is --
        7       -1      -1      PF No. 1
        7       0       -1      PF No. 2
        7       0       1       PF No. 3
        2       0       1       PF No. 4
        2       0       1
        2       3       1       PF No. 5
        2       3       0       PF No. 6
        4       3       0       PF No. 7
        4       2       0       PF No. 8
        4       2       3       PF No. 9
        0       2       3       PF No. 10
        0       2       3
        0       2       3
        0       1       3       PF No. 11
        0       1       2       PF No. 12
        0       1       2
        0       1       2
        7       1       2       PF No. 13
        7       0       2       PF No. 14
        7       0       1       PF No. 15

 The number of Page Faults using FIFO are 15
Process returned 0 (0x0)   execution time : 41.360 s
Press any key to continue.
```

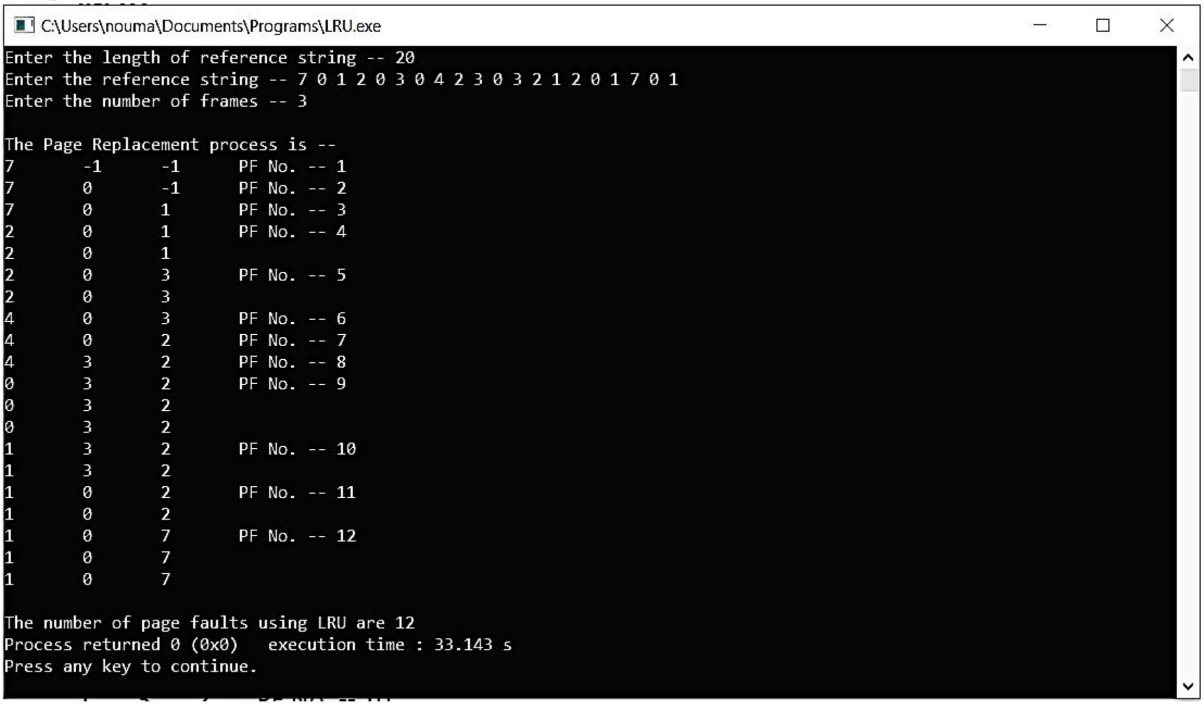
LRU:

Program:

```
#include<stdio.h>

main()
{
   int i, j , k, min, rs[25], m[10], count[10], flag[25], n, f, pf=0, next=1;
   printf("Enter the length of reference string -- ");
   scanf("%d",&n);
   printf("Enter the reference string -- ");

   for(i=0;i<n;i++)
      {
         scanf("%d",&rs[i]);
         flag[i]=0;
      }
   printf("Enter the number of frames -- ");
   scanf("%d",&f);

   for(i=0;i<f;i++)
      {
         count[i]=0; m[i]=-1;
      }
   printf("\nThe Page Replacement process is -- \n");

   for(i=0;i<n;i++)
      {
         for(j=0;j<f;j++)
         {
            if(m[j]==rs[i])
            {
               flag[i]=1;
               count[j]=next;
               next++;
            }
         }
      if(flag[i]==0)
         {
            if(i<f)
            {
               m[i]=rs[i];
               count[i]=next; next++;
            }
            else
```

```
               {
                  min=0;
                  for(j=1;j<f;j++)
                     if(count[min] > count[j])
                     min=j; m[min]=rs[i];
                     count[min]=next; next++;
               }
            pf++;
         }
      for(j=0;j<f;j++)
         printf("%d\t", m[j]);
      if(flag[i]==0)
         printf("PF No. -- %d" , pf);
      printf("\n");
   }
   printf("\nThe number of page faults using LRU are %d",pf);
}
```

Output:



```
Enter the length of reference string -- 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

The Page Replacement process is --
7       -1      -1      PF No. -- 1
7       0       -1      PF No. -- 2
7       0       1       PF No. -- 3
2       0       1       PF No. -- 4
2       0       1
2       0       3       PF No. -- 5
2       0       3
4       0       3       PF No. -- 6
4       0       2       PF No. -- 7
4       3       2       PF No. -- 8
0       3       2       PF No. -- 9
0       3       2
0       3       2
1       3       2       PF No. -- 10
1       3       2
1       0       2       PF No. -- 11
1       0       2
1       0       7       PF No. -- 12
1       0       7
1       0       7

The number of page faults using LRU are 12
Process returned 0 (0x0)   execution time : 33.143 s
Press any key to continue.
```

LFU:

Program:

```
#include<stdio.h>

main()
{
   int rs[50], i, j, k, m, f, cntr[20], a[20], min, pf=0;
   printf("\nEnter number of page references -- ");
   scanf("%d",&m);
   printf("\nEnter the reference string -- ");

   for(i=0;i<m;i++)
      scanf("%d",&rs[i]);

   printf("\nEnter the available no. of frames -- ");
   scanf("%d",&f);

   for(i=0;i<f;i++)
      {
         cntr[i]=0; a[i]=-1;
      }

   Printf("\nThe Page Replacement Process is – \n");

   for(i=0;i<m;i++)
      {
         for(j=0;j<f;j++)
            if(rs[i]==a[j])
            {
               cntr[j]++;
               break;
            }
         if(j==f)
         {
            min = 0;
            for(k=1;k<f;k++)
               if(cntr[k]<cntr[min])
                  min=k;

               a[min]=rs[i];
               cntr[min]=1;
               pf++;
         }
      printf("\n");
      for(j=0;j<f;j++)
         printf("\t%d",a[j]);
      if(j==f)
         printf("\tPF No. %d",pf);
      }
   printf("\n\n Total number of page faults -- %d",pf);
}
```