PROGRAM 7

7. Develop a C program to simulate page replacement algorithms: a) FIFO b) LRU

DESCRIPTION:

Page Replacement Algorithms: 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.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
void FIFO(char [ ],char [ ],int,int);
void lru(char [ ],char [ ],int,int);
void opt(char [ ],char [ ],int,int);
int main()
{
        int ch,YN=1,i,l,f;
        char F[10],s[25];
        printf("\n\n\tEnter the no of empty frames: ");
        scanf("%d",&f);
        printf("\n\n\tEnter the length of the string: ");
        scanf("%d",&l);
        printf("\n\n\tEnter the string: ");
        scanf("%s",s);
        for(i=0;i<f;i++)
        F[i]=-1;
        do
        {
                printf("\n\n\t********** MENU **********");
                printf("\n\n\t1:FIFO\n\n\t2:LRU \n\n\t4:EXIT");
                printf("\n\n\tEnter your choice: ");
                scanf("%d",&ch);
```

```
                switch(ch)
                {
                        case 1:
                                for(i=0;i<f;i++)
                                {
                                        F[i]=-1;
                                }
                                FIFO(s,F,l,f);
                        break;
                        case 2:
                                for(i=0;i<f;i++)
                                {
                                        F[i]=-1;
                                }
                                lru(s,F,l,f);
                        break;
                        case 4:
                                exit(0);
                }
                printf("\n\n\tDo u want to continue IF YES PRESS 1\n\n\tIF NO PRESS 0 : ");
                scanf("%d",&YN);
        }while(YN==1);
        return(0);
}

//FIFO
void FIFO(char s[],char F[],int l,int f)
{
        int i,j=0,k,flag=0,cnt=1;
        printf("\n\tPAGE\t FRAMES\t FAULTS");
        for(i=0;i<l;i++)
        {
                for(k=0;k<f;k++)
                {
                        if(F[k]==s[i])
                        flag=1;
                }
                if(flag==0)
                {
                        printf("\n\t%c\t",s[i]);
                        F[j]=s[i];
                        j++;
                        for(k=0;k<f;k++)
                        {
                                printf(" %c",F[k]);
                        }
                        printf("\tPage-fault%d",cnt);
                        cnt++;
                }
                else
```

```
                {
                        flag=0;
                        printf("\n\t%c\t",s[i]);
                        for(k=0;k<f;k++)
                        {
                                printf(" %c",F[k]);
                        }
                        printf("\tNo page-fault");
                }
                if(j==f)
                        j=0;
        }
}
//LRU
void lru(char s[],char F[],int l,int f)
{
        int i,j=0,k,m,flag=0,cnt=1,top=0;
        printf("\n\tPAGE\t FRAMES\t FAULTS");
        for(i=0;i<l;i++)
        {
                for(k=0;k<f;k++)
                {
                        if(F[k]==s[i])
                        {
                                flag=1;
                                break;
                        }
                }
                printf("\n\t%c\t",s[i]);
                if(j!=f && flag!=1)
                {
                        F[top]=s[i];
                        j++;
                        if(j!=f)
                                top++;
                }
                else
                {
                        if(flag!=1)
                        {
                                for(k=0;k<top;k++)
                                {
                                        F[k]=F[k+1];
                                }
                                F[top]=s[i];
                        }
                        if(flag==1)
                        {
                                for(m=k;m<top;m++)
                                {
```

```
                                        F[m]=F[m+1];
                                }
                                F[top]=s[i];
                        }
                }
                for(k=0;k<f;k++)
                {
                        printf(" %c",F[k]);
                }
                if(flag==0)
                {
                        printf("\tPage-fault%d",cnt);
                        cnt++;
                }
                else
                        printf("\tNo page fault");
                flag=0;
        }
}
```

OUTPUT

```
Enter the no of empty frames: 3

Enter the length of the string: 12

Enter the string: 476176127271

********** MENU **********

1:FIFO

2:LRU

4:EXIT

Enter your choice: 1

PAGE    FRAMES  FAULTS
4       4 ◆ ◆  Page-fault1
7       4 7 ◆  Page-fault2
6       4 7 6  Page-fault3
1       1 7 6  Page-fault4
7       1 7 6  No page-fault
6       1 7 6  No page-fault
1       1 7 6  No page-fault
2       1 2 6  Page-fault5
7       1 2 7  Page-fault6
2       1 2 7  No page-fault
```

```
7       1 2 7  No page-fault
1       1 2 7  No page-fault

Do u want to continue IF YES PRESS 1

IF NO PRESS 0 : 1


********** MENU **********

1:FIFO

2:LRU

4:EXIT

Enter your choice: 2

PAGE    FRAMES  FAULTS
4       4 ? ?  Page-fault1
7       4 7 ?  Page-fault2
6       4 7 6  Page-fault3
1       7 6 1  Page-fault4
7       6 1 7  No page fault
6       1 7 6  No page fault
1       7 6 1  No page fault
2       6 1 2  Page-fault5
7       1 2 7  Page-fault6
2       1 7 2  No page fault
7       1 2 7  No page fault
1       2 7 1  No page fault

Do u want to continue IF YES PRESS 1

IF NO PRESS 0 : 0
```