

Assignment 6

code:

```
#include<stdio.h>
```

```
void menu()
```

```
{
    printf("\n\n\t\t**MAIN MENU**");
    printf("\n\n\tPAGE REPLACEMENT ALGORITHMS");
    printf("\n\n\t1.FIRST IN FIRST OUT");
    printf("\n\n\t2.LEAST RECENTLY USED");
    printf("\n\n\t3.OPTIMAL");
    printf("\n\n\t4.QUIT");
}
```

```
void accept(int ref[],int n)
```

```
{
    int i;
    printf("\nENTER ELEMENT OF REFERENCE STRING ONE BY ONE::");
    for(i=0;i<n;i++)
    {
        scanf("%d",&ref[i]);
    }
}
```

```
void display(int ref[],int n)
```

```
{
    int i;
    //printf("\n\n");
    for(i=0;i<n;i++)
    {
        printf("%3d ",ref[i]);
    }
}
```

```
int search(int frame[],int x,int fs)
```

```
{
    int i;
    for(i=0;i<fs;i++)
    {
        if(frame[i]==x)
            return 1;
    }
    return 0;
}
```

```

void fifo(int ref[],int n)
{
    int frame[10];
    int i,k,fs,temp,cnt;
    printf("\n\nENTER FRAME SIZE::");
    scanf("%d",&temp);

    cnt=k=fs=0;

    for(i=0;i<n;i++)
    {
        if(search(frame,ref[i],fs)==0)
        {
            frame[k]=ref[i];
            k=(k+1)%temp;
            cnt++;
            if(cnt<temp)
                fs=cnt;
            else
                fs=temp;
            printf("\n\nFOR PAGE %d ::\t",ref[i]);
            display(frame,fs);
        }
        else
        {
            printf("\n\nFOR PAGE %d ::\t",ref[i]);
            printf(" NO PAGE FAULT");
        }
    }
    printf("\n\nTOTAL NO. OF PAGE FAULTS = %d ",cnt);
}

/*-----*/

```

```

void lru(int ref[],int n)
{
    int frame[30],pos[30];
    int i,j,k,fs,temp,cnt,min;
    printf("\n\nENTER FRAME SIZE::");
    scanf("%d",&fs);

    cnt=k=0;

    for(i=0;i<fs && i<n;i++)
    {
        if(search(frame,ref[i],i)==0)
        {
            frame[k++]=ref[i];
            cnt++;
        }
    }
}

```

```

        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        display(frame,k);    //frame size=k
    }
    else
    {
        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        printf(" NO PAGE FAULT");
    }
}

for(;i<n;i++)
{
    if(search(frame,ref[i],fs)==0)
    {
        for(j=0;j<fs;j++)
        {
            for(k=i-1;k>=0;k--)
            {
                if(frame[j]==ref[k])
                    break;
            }
            pos[j]=k;
        }
        k=0;
        min=pos[0];
        for(j=1;j<fs;j++)
        {
            if(min>pos[j])
            {
                min=pos[j];
                k=j;
            }
        }
        frame[k]=ref[i];
        cnt++;
        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        display(frame,fs);
    }
    else
    {
        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        printf(" NO PAGE FAULT");
    }
}

printf("\n\nTOTAL NO. OF PAGE FAULTS = %d ",cnt);
}

/*-----*/

```

```

void optimal(int ref[],int n)
{
    int frame[30],pos[30];
    int i,j,k,fs,temp,cnt,max;
    printf("\n\nENTER FRAME SIZE::");
    scanf("%d",&fs);

    cnt=k=0;

    for(i=0;i<fs && i<n;i++)
    {
        if(search(frame,ref[i],i)==0)
        {
            frame[k++]=ref[i];
            cnt++;
            printf("\n\nFOR PAGE %d ::\t",ref[i]);
            display(frame,k);    //frame size=k
        }
        else
        {
            printf("\n\nFOR PAGE %d ::\t",ref[i]);
            printf(" NO PAGE FAULT");
        }
    }

    for(;i<n;i++)
    {
        if(search(frame,ref[i],fs)==0)
        {
            for(j=0;j<fs;j++)
            {
                for(k=i+1;k<n;k++)
                {
                    if(frame[j]==ref[k])
                        break;
                }
                pos[j]=k;
            }
            k=0;
            max=pos[0];
            for(j=1;j<fs;j++)
            {
                if(max<pos[j])
                {
                    max=pos[j];
                    k=j;
                }
            }
            frame[k]=ref[i];
        }
    }
}

```

```

        cnt++;
        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        display(frame,fs);
    }
    else
    {
        printf("\n\nFOR PAGE %d ::\t",ref[i]);
        printf(" NO PAGE FAULT");
    }
}
printf("\n\nTOTAL NO. OF PAGE FAULTS = %d ",cnt);
}

```

```

/*-----*/

```

```

int main()
{
    int ref[50],n,ch;

    printf("\n\nENTER SIZE OF REFERENCE STRING::");
    scanf("%d",&n);
    accept(ref,n);

    while(ch!=4)
    {
        menu();
        printf("\n\nENTER YOUR CHOICE::");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                printf("\nYOUR ENTERED REFERENCE STRING IS::\n\n");
                display(ref,n);
                fifo(ref,n);
                break;
            case 2:
                printf("\nYOUR ENTERED REFERENCE STRING IS::\n\n");
                display(ref,n);
                lru(ref,n);
                break;
            case 3:
                printf("\nYOUR ENTERED REFERENCE STRING IS::\n\n");
                display(ref,n);
                optimal(ref,n);
                break;

```

```
}
```

```
}
```

```
return(0);
```

```
}
```

Output:

1:

```
ak-linux-computer@fedora-a... [ak-linux-computer@ak ~]$ gcc 6.c -o 6
[ak-linux-computer@ak ~]$ ./6

ENTER SIZE OF REFERENCE STRING::20

ENTER ELEMENT OF REFERENCE STRING ONE BY ONE::7
0
1
2
0
3
0
4
2
3
0
3
2
1
2
0
1
7
0
1

**MAIN MENU**

PAGE REPLACEMENT ALGORITHMS

1.FIRST IN FIRST OUT
2.LEAST RECENTLY USED
3.OPTIMAL
4.QUIT

ENTER YOUR CHOICE::1

YOUR ENTERED REFERENCE STRING IS::

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

ENTER FRAME SIZE::3

FOR PAGE 7 ::      7
FOR PAGE 0 ::      7    0
FOR PAGE 1 ::      7    0    1
```

2:

```
ak-linux-computer@fedora-a...  
FOR PAGE 7 ::      7  
FOR PAGE 0 ::      7      0  
FOR PAGE 1 ::      7      0      1  
FOR PAGE 2 ::      2      0      1  
FOR PAGE 0 ::      NO PAGE FAULT  
FOR PAGE 3 ::      2      3      1  
FOR PAGE 0 ::      2      3      0  
FOR PAGE 4 ::      4      3      0  
FOR PAGE 2 ::      4      2      0  
FOR PAGE 3 ::      4      2      3  
FOR PAGE 0 ::      0      2      3  
FOR PAGE 3 ::      NO PAGE FAULT  
FOR PAGE 2 ::      NO PAGE FAULT  
FOR PAGE 1 ::      0      1      3  
FOR PAGE 2 ::      0      1      2  
FOR PAGE 0 ::      NO PAGE FAULT  
FOR PAGE 1 ::      NO PAGE FAULT  
FOR PAGE 7 ::      7      1      2  
FOR PAGE 0 ::      7      0      2  
FOR PAGE 1 ::      7      0      1  
TOTAL NO. OF PAGE FAULTS = 15
```


3:

3:

```
ak-linux-computer@fedora-a...
**MAIN MENU**

PAGE REPLACEMENT ALGORITHMS

1.FIRST IN FIRST OUT
2.LEAST RECENTLY USED
3.OPTIMAL
4.QUIT

ENTER YOUR CHOICE::2

YOUR ENTERED REFERENCE STRING IS::

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

ENTER FRAME SIZE::3

FOR PAGE 7 ::      7
FOR PAGE 0 ::      7   0
FOR PAGE 1 ::      7   0   1
FOR PAGE 2 ::      2   0   1
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 3 ::      2   0   3
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 4 ::      4   0   3
FOR PAGE 2 ::      4   0   2
FOR PAGE 3 ::      4   3   2
FOR PAGE 0 ::      0   3   2
FOR PAGE 3 ::      NO PAGE FAULT
FOR PAGE 2 ::      NO PAGE FAULT
FOR PAGE 1 ::      1   3   2
FOR PAGE 2 ::      NO PAGE FAULT
FOR PAGE 0 ::      1   0   2
```

4:

```
ak-linux-computer@fedora-a...
FOR PAGE 1 ::      NO PAGE FAULT
FOR PAGE 7 ::      1    0    7
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 1 ::      NO PAGE FAULT
TOTAL NO. OF PAGE FAULTS = 12

      **MAIN MENU**

      PAGE REPLACEMENT ALGORITHMS

      1.FIRST IN FIRST OUT
      2.LEAST RECENTLY USED
      3.OPTIMAL
      4.QUIT

ENTER YOUR CHOICE::3

YOUR ENTERED REFERENCE STRING IS::

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

ENTER FRAME SIZE::3

FOR PAGE 7 ::      7
FOR PAGE 0 ::      7    0
FOR PAGE 1 ::      7    0    1
FOR PAGE 2 ::      2    0    1
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 3 ::      2    0    3
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 4 ::      2    4    3
FOR PAGE 2 ::      NO PAGE FAULT
FOR PAGE 3 ::      NO PAGE FAULT
FOR PAGE 0 ::      2    0    3
```

5:

```
FOR PAGE 3 ::      NO PAGE FAULT
FOR PAGE 2 ::      NO PAGE FAULT
FOR PAGE 1 ::      2      0      1
FOR PAGE 2 ::      NO PAGE FAULT
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 1 ::      NO PAGE FAULT
FOR PAGE 7 ::      7      0      1
FOR PAGE 0 ::      NO PAGE FAULT
FOR PAGE 1 ::      NO PAGE FAULT
TOTAL NO. OF PAGE FAULTS = 9
```

MAIN MENU

PAGE REPLACEMENT ALGORITHMS

1.FIRST IN FIRST OUT

2.LEAST RECENTLY USED

3.OPTIMAL

4.QUIT

ENTER YOUR CHOICE::4

[ak-linux-computer@ak ~]\$