VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE 2005: OPERATING SYSTEMS

DIGITAL ASSIGNMENT 5

Page Replacement Algorithms

Anushka Dixit
[19BCE0577]

Question: Implement FIFO, Optimal, LRU, LFU and MFU Page Replacement Algorithms with 3, 5, 7 frames with some reference string

Code:

```
#include<stdio.h>
int n,nf;
int in[100];
int p[50];
int hit=0;
int i,j,k;
int pgfaultcnt=0;

void getData()
{
    printf("\nEnter length of page reference sequence:");
    scanf("%d",&n);
    printf("\nEnter the page reference sequence:");
    for(i=0; i<n; i++)
        scanf("%d",&in[i]);
    printf("\nEnter no of frames:");
    scanf("%d",&nf);
}

void initialize()
{
    pgfaultcnt=0;
    for(i=0; i<nf; i++)
        p[i]=9999;
}

int isHit(int data)
{
    hit=0;
    for(j=0; j<nf; j++)
    {
        if(p[j]==data)
        {
            hit=1;
            break;
        }

    }

    return hit;
}

int getHitIndex(int data)
{
```

```
    int hitind;
    for(k=0; k<nf; k++)
    {
        if(p[k]==data)
        {
            hitind=k;
            break;
        }
    }
    return hitind;
}

void dispPages()
{
    for (k=0; k<nf; k++)
    {
        if(p[k]!=9999)
            printf(" %d",p[k]);
    }

}

void dispPgFaultCnt()
{
    printf("\nTotal no of page faults:%d",pgfaultcnt);
}

void fifo()
{
    initialize();
    for(i=0; i<n; i++)
    {
        printf("\nFor %d :",in[i]);

        if(isHit(in[i])==0)
        {

            for(k=0; k<nf-1; k++)
                p[k]=p[k+1];

            p[k]=in[i];
            pgfaultcnt++;
            dispPages();
        }
        else
            printf("No page fault");
    }
    dispPgFaultCnt();
```

```
}


void optimal()
{
    initialize();
    int near[50];
    for(i=0; i<n; i++)
    {

        printf("\nFor %d :",in[i]);

        if(isHit(in[i])==0)
        {

            for(j=0; j<nf; j++)
            {
                int pg=p[j];
                int found=0;
                for(k=i; k<n; k++)
                {
                    if(pg==in[k])
                    {
                        near[j]=k;
                        found=1;
                        break;
                    }
                    else
                        found=0;
                }
                if(!found)
                    near[j]=9999;
            }
            int max=-9999;
            int repindex;
            for(j=0; j<nf; j++)
            {
                if(near[j]>max)
                {
                    max=near[j];
                    repindex=j;
                }
            }
            p[repindex]=in[i];
            pgfaultcnt++;

            dispPages();
        }
```

```
        else
            printf("No page fault");
    }
    dispPgFaultCnt();
}

void lru()
{
    initialize();

    int least[50];
    for(i=0; i<n; i++)
    {

        printf("\nFor %d :",in[i]);

        if(isHit(in[i])==0)
        {

            for(j=0; j<nf; j++)
            {
                int pg=p[j];
                int found=0;
                for(k=i-1; k>=0; k--)
                {
                    if(pg==in[k])
                    {
                        least[j]=k;
                        found=1;
                        break;
                    }
                    else
                        found=0;
                }
                if(!found)
                    least[j]=-9999;
            }
            int min=9999;
            int repindex;
            for(j=0; j<nf; j++)
            {
                if(least[j]<min)
                {
                    min=least[j];
                    repindex=j;
                }
            }
            p[repindex]=in[i];
```

```
            pgfaultcnt++;

            dispPages();
        }
        else
            printf("No page fault!");
    }
    dispPgFaultCnt();
}

void lfu()
{
    int usedcnt[100];
    int least,repin,sofarcnt=0,bn;
    initialize();
    for(i=0; i<nf; i++)
        usedcnt[i]=0;

    for(i=0; i<n; i++)
    {

        printf("\n For %d :",in[i]);
        if(isHit(in[i]))
        {
            int hitind=getHitIndex(in[i]);
            usedcnt[hitind]++;
            printf("No page fault!");
        }
        else
        {
            pgfaultcnt++;
            if(bn<nf)
            {
                p[bn]=in[i];
                usedcnt[bn]=usedcnt[bn]+1;
                bn++;
            }
            else
            {
                least=9999;
                for(k=0; k<nf; k++)
                    if(usedcnt[k]<least)
                    {
                        least=usedcnt[k];
                        repin=k;
                    }
                p[repin]=in[i];
                sofarcnt=0;
```

```
                for(k=0; k<=i; k++)
                    if(in[i]==in[k])
                        sofarcnt=sofarcnt+1;
                usedcnt[repin]=sofarcnt;
            }

            dispPages();
        }

    }
    dispPgFaultCnt();
}



int main()
{
    int choice;
    while(1)
    {
        printf("\nPage Replacement Algorithms\n1.Enter data\n2.FIFO\n3.Optimal
\n4.LRU\n5.LFU\n6.Second Chance\n7.Exit\nEnter your choice:");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
            getData();
            break;
        case 2:
            fifo();
            break;
        case 3:
            optimal();
            break;
        case 4:
            lru();
            break;
        case 5:
            lfu();
            break;

        default:
            return 0;
            break;
        }
    }
}
```

Output:

Part 1: Number of frames =3

```
anushka--os@LAPTOP-6G5U0QLQ:~$ ./page

Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:1

Enter length of page reference sequence:15

Enter the page reference sequence:7
0
1
2
0
3
0
4
2
3
0
3
2
0
1

Enter no of frames:3
```

FIFO Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:2

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 0 1 2
For 0 :No page fault
For 3 : 1 2 3
For 0 : 2 3 0
For 4 : 3 0 4
For 2 : 0 4 2
For 3 : 4 2 3
For 0 : 2 3 0
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 : 3 0 1
Total no of page faults:11
```

Optimal Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:3

For 7 : 7
For 0 : 0
For 1 : 0 1
For 2 : 0 1 2
For 0 :No page fault
For 3 : 0 3 2
For 0 :No page fault
For 4 : 4 3 2
For 2 :No page fault
For 3 :No page fault
For 0 : 0 3 2
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 : 1 3 2
Total no of page faults:8
```

LRU Algorithm

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:4

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 2 0 1
For 0 :No page fault!
For 3 : 2 0 3
For 0 :No page fault!
For 4 : 4 0 3
For 2 : 4 0 2
For 3 : 4 3 2
For 0 : 0 3 2
For 3 :No page fault!
For 2 :No page fault!
For 0 :No page fault!
For 1 : 0 1 2
Total no of page faults:10
```

LFU Algorithm

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:5

 For 7 : 7
 For 0 : 7 0
 For 1 : 7 0 1
 For 2 : 2 0 1
 For 0 :No page fault!
 For 3 : 3 0 1
 For 0 :No page fault!
 For 4 : 4 0 1
 For 2 : 2 0 1
 For 3 : 2 0 3
 For 0 :No page fault!
 For 3 :No page fault!
 For 2 :No page fault!
 For 0 :No page fault!
 For 1 : 1 0 3
Total no of page faults:9
```

Part 2: Number of frames =5

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:1

Enter length of page reference sequence:15

Enter the page reference sequence:7
0
1
2
0
3
0
4
2
3
0
3
2
0
1

Enter no of frames:5
```

FIFO Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:2

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 7 0 1 2
For 0 :No page fault
For 3 : 7 0 1 2 3
For 0 :No page fault
For 4 : 0 1 2 3 4
For 2 :No page fault
For 3 :No page fault
For 0 :No page fault
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 :No page fault
Total no of page faults:6
```

Optimal Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:3

For 7 : 7
For 0 : 0
For 1 : 0 1
For 2 : 0 1 2
For 0 :No page fault
For 3 : 0 1 2 3
For 0 :No page fault
For 4 : 0 1 2 3 4
For 2 :No page fault
For 3 :No page fault
For 0 :No page fault
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 :No page fault
Total no of page faults:6
```

LRU Algorithm

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:4

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 7 0 1 2
For 0 :No page fault!
For 3 : 7 0 1 2 3
For 0 :No page fault!
For 4 : 4 0 1 2 3
For 2 :No page fault!
For 3 :No page fault!
For 0 :No page fault!
For 3 :No page fault!
For 2 :No page fault!
For 0 :No page fault!
For 1 :No page fault!
For 0 :No page fault!
For 1 :No page fault!
Total no of page faults:6
```

LFU Algorithm

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:5

 For 7 : 7
 For 0 : 7 0
 For 1 : 7 0 1
 For 2 : 7 0 1 2
 For 0 :No page fault!
 For 3 : 7 0 1 2 3
 For 0 :No page fault!
 For 4 : 4 0 1 2 3
 For 2 :No page fault!
 For 3 :No page fault!
 For 0 :No page fault!
 For 3 :No page fault!
 For 2 :No page fault!
 For 0 :No page fault!
 For 1 :No page fault!
 For 0 :No page fault!
 For 1 :No page fault!
Total no of page faults:6
```

Part 3: Number of frames =7

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:1

Enter length of page reference sequence:15

Enter the page reference sequence:7
0
1
2
0
3
0
4
2
3
0
3
2
0
1

Enter no of frames:7
```

FIFO Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:2

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 7 0 1 2
For 0 :No page fault
For 3 : 7 0 1 2 3
For 0 :No page fault
For 4 : 7 0 1 2 3 4
For 2 :No page fault
For 3 :No page fault
For 0 :No page fault
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 :No page fault
Total no of page faults:6
```

Optimal Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.Exit
Enter your choice:3

For 7 : 7
For 0 : 0
For 1 : 0 1
For 2 : 0 1 2
For 0 :No page fault
For 3 : 0 1 2 3
For 0 :No page fault
For 4 : 0 1 2 3 4
For 2 :No page fault
For 3 :No page fault
For 0 :No page fault
For 3 :No page fault
For 2 :No page fault
For 0 :No page fault
For 1 :No page fault
Total no of page faults:6
```

LRU Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:4

For 7 : 7
For 0 : 7 0
For 1 : 7 0 1
For 2 : 7 0 1 2
For 0 :No page fault!
For 3 : 7 0 1 2 3
For 0 :No page fault!
For 4 : 7 0 1 2 3 4
For 2 :No page fault!
For 3 :No page fault!
For 0 :No page fault!
For 3 :No page fault!
For 2 :No page fault!
For 0 :No page fault!
For 1 :No page fault!
Total no of page faults:6
```

LFU Algorithm:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.LFU
6.Exit
Enter your choice:5

 For 7 : 7
 For 0 : 0 7
 For 1 : 0 1 7
 For 2 : 0 1 2 7
 For 0 :No page fault!
 For 3 : 0 1 2 3 7
 For 0 :No page fault!
 For 4 : 0 1 2 3 4 7
 For 2 :No page fault!
 For 3 :No page fault!
 For 0 :No page fault!
 For 3 :No page fault!
 For 2 :No page fault!
 For 0 :No page fault!
 For 1 :No page fault!
Total no of page faults:6
```

Most Frequently Used Algorithm

Code:

```
#include<stdio.h>
#define SIZE 3
int full=0;//To check whether all frames are filled
int a[21],n;//To take the input
int frame[SIZE];
int ctr[SIZE]={0};
static int f;
int repptr;
int count=0;
int display()
{int i;
printf("\nThe elements in the frame are\n");
for(i=0;i<full;i++)
printf("%d\n",frame[i]);

}
int Longestopt()
{int i,max;
    max=0;
    for(i=0;i<SIZE;i++)//The page with maximum frequency is selected
    if(ctr[max]<ctr[i])
    max=i;

    repptr=max;
return repptr;

}
int Pagerep(int ele)
{
 int temp;
 repptr=Longestopt();
 temp=frame[repptr];
 frame[repptr]=ele;
 ctr[repptr]=1;


 return temp;
}
int Pagefault(int ele)
{if(full!=SIZE)
{ctr[full]++;
               frame[full++]=ele;
}
else
```

```
printf("The page replaced is %d",Pagerep(ele));
}
int Search(int ele)
{int i,flag;
    flag=0;
    if(full!=0)
    {
    for(i=0;i<full;i++)
    if(ele==frame[i])
 {   flag=1;ctr[i]++;
    break;
}}
 return flag;
}
int main()
{int i;
    FILE *fp;
    fp=fopen("Input.txt","r");
    printf("The number of elements in the reference string are :");
    fscanf(fp,"%d",&n);
    printf("%d",n);
    for(i=0;i<n;i++)
    fscanf(fp,"%d",&a[i]);
    printf("\nThe elements present in the string are\n");
    for(i=0;i<n;i++)
    printf("%d  ",a[i]);
    printf("\n\n");
    for(i=0;i<n;i++)
    {f=i;
                    if(Search(a[i])!=1)
                    {Pagefault(a[i]);
                    display();
                    count++;
                    }

                    }
                    printf("\nThe number of page faults are %d\n",count);
                    getche();
return 0;
}
```

Output:

No. of page frames = 3

```
The number of elements in the reference string are :13
The elements present in the string are
7  0  1  2  0  3  0  4  2  3  0  3  2


The elements in the frame are
7

The elements in the frame are
7
0

The elements in the frame are
7
0
1
The page replaced is 7
The elements in the frame are
2
0
1
The page replaced is 0
The elements in the frame are
2
3
1
The page replaced is 2
The elements in the frame are
0
3
1
The page replaced is 0
The elements in the frame are
4
3
1
The page replaced is 4
The elements in the frame are
2
3
1
The page replaced is 3
The elements in the frame are
2
0
1
The page replaced is 2
The elements in the frame are
3
0
1
The page replaced is 3
The elements in the frame are
2
0
1

The number of page faults are 11
```

No. of page frames = 5

```
The number of elements in the reference string are :13
The elements present in the string are
7  0  1  2  0  3  0  4  2  3  0  3  2


The elements in the frame are
7

The elements in the frame are
7
0

The elements in the frame are
7
0
1

The elements in the frame are
7
0
1
2

The elements in the frame are
7
0
1
2
3
The page replaced is 0
The elements in the frame are
7
4
1
2
3
The page replaced is 2
The elements in the frame are
7
4
1
0
3
The page replaced is 3
The elements in the frame are
7
4
1
0
2

The number of page faults are 8
```

No. of page frames = 7

```
The number of elements in the reference string are :13
The elements present in the string are
7  0  1  2  0  3  0  4  2  3  0  3  2


The elements in the frame are
7

The elements in the frame are
7
0

The elements in the frame are
7
0
1

The elements in the frame are
7
0
1
2

The elements in the frame are
7
0
1
2
3

The elements in the frame are
7
0
1
2
3
4

The number of page faults are 6
```