LAB 13

PAGE REPLACEMENT ALGORITHMS

P.PRIYADHARSHINI

2019103562

03-05-2021

FIFO(FIRST IN FIRST OUT):

```
[s2019103562@centos8-linux Mon May 03 09:20 AM lab13]$ ./fifo
Enter the number of frames:     3
Enter the number of pages:      20
Enter the pages
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Ref string               Page frames
7               7       -1      -1
0               7       0       -1
1               7       0       1
2               2       0       1
0
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
2
1               0       1       3
2               0       1       2
0
1
7               7       1       2
0               7       0       2
1               7       0       1
Total number of page faults:15
[s2019103562@centos8-linux Mon May 03 09:21 AM lab13]$
```

```
[s2019103562@centos8-linux Mon May 03 08:55 AM lab13]$ cat fifo.c
#include<stdio.h>
int main(){
        int i,j,n,pages[50],frame[10],n1,k,avail,count=0;
        printf("Enter the number of frames:\t");
        scanf("%d",&n1);
        printf("Enter the number of pages:\t");
        scanf("%d",&n);
        printf("Enter the pages\n");
        for(i=1;i<=n;i++)
                scanf("%d",&pages[i]);
        for(i=0;i<n1;i++)
                frame[i]=-1;
        j=0;
        printf("Ref string\t\t Page frames\n");
        for(i=1;i<=n;i++){
                printf("%d\t\t",pages[i]);
                avail=0;
                for(k=0;k<n1;k++)
                        if(frame[k]==pages[i])
                                avail=1;
                if(avail==0){
                        frame[j]=pages[i];
                        j=(j+1)%n1;
                        count++;
                        for(k=0;k<n1;k++)
                                printf("%d\t",frame[k]);
                }
                printf("\n");
        }
        printf("Total number of page faults:%d\n",count);
        return 0;
}
[s2019103562@centos8-linux Mon May 03 09:00 AM lab13]$
```

OPTIMAL PAGE REPLACEMENT ALGORITHM (OPR):

```
[s2019103562@centos8-linux Mon May 03 09:25 AM lab13]$ ./opt
Enter the number of frames:     3
Enter the number of pages:      20
Enter the pages
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7       -1      -1
7       0       -1
7       0       1
2       0       1
2       0       1
2       0       3
2       0       3
2       4       3
2       4       3
2       4       3
2       0       3
2       0       3
2       0       3
2       0       1
2       0       1
2       0       1
2       0       1
7       0       1
7       0       1
7       0       1
Total number of page faults:9
[s2019103562@centos8-linux Mon May 03 09:27 AM lab13]$
```

```
[s2019103562@centos8-linux Mon May 03 09:21 AM lab13]$ vim opt.c
[s2019103562@centos8-linux Mon May 03 09:23 AM lab13]$ gcc opt.c -o opt
[s2019103562@centos8-linux Mon May 03 09:23 AM lab13]$ ./opt
Enter the number of frames:     3
Enter the number of pages:      12
Enter the pages
1 3 2 4 2 3 1 4 2 4 1 3

1       -1      -1
1       3       -1
1       3       2
4       3       2
4       3       2
4       3       2
4       1       2
4       1       2
4       1       2
4       1       2
4       1       2
3       1       2
Total number of page faults:6
[s2019103562@centos8-linux Mon May 03 09:24 AM lab13]$
```

```
[s2019103562@centos8-linux Mon May 03 09:24 AM lab13]$ cat opt.c
#include<stdio.h>
int main(){
        int no_of_frames,no_of_pages,frames[10],pages[50],temp[10],flag1,flag2,flag3,i,j,k,pos,max,faults=0;
        printf("Enter the number of frames:\t");
        scanf("%d",&no_of_frames);
        printf("Enter the number of pages:\t");
        scanf("%d",&no_of_pages);
        printf("Enter the pages\n");
        for(i=0;i<no_of_pages;i++)
                scanf("%d",&pages[i]);
        for(i=0;i<no_of_frames;i++)
                frames[i]=-1;
        //printf("Ref string\t Page frames\n");
        for(i=0;i<no_of_pages;i++){
                //printf("%d\t",pages[i]);
                flag1=flag2=0;
                for(j=0;j<no_of_frames;j++){
                        if(frames[j]==pages[i]){
                                flag1=flag2=1;
                                break;
                        }
                }
                if(flag1==0){
                        for(j=0;j<no_of_frames;j++){
                                if(frames[j]==-1){
                                        faults++;
                                        frames[j]=pages[i];
                                        flag2=1;
                                        break;
                                }
                        }
                }
```

```
                }
                if(flag2==0){
                        flag3=0;
                        for(j=0;j<no_of_frames;j++){
                                temp[j]=-1;
                                for(k=i+1;k<no_of_pages;k++){
                                        if(frames[j]==pages[k]){
                                                temp[j]=k;
                                                break;
                                        }
                                }
                        }
                for(j=0;j<no_of_frames;j++){
                        if(temp[j]==-1){
                                pos=j;
                                flag3=1;
                                break;
                        }
                }
                if(flag3==0){
                        max=temp[0];
                        pos=0;
                        for(j=1;j<no_of_frames;j++){
                                if(temp[j]>max){
                                        max=temp[j];
                                        pos=j;
                                }
                        }
                }
                frames[pos]=pages[i];
                faults++;
                }
                printf("\n");
                for(j=0;j<no_of_frames;j++)
                        printf("%d\t",frames[j]);
        }
        printf("\nTotal number of page faults:%d\n",faults);
        return 0;
}
[s2019103562@centos8-linux Mon May 03 09:25 AM lab13]$
```

LRU(LEAST RECENTLY USED):

```
[s2019103562@centos8-linux Mon May 03 09:51 AM lab13]$ vim lru.c
[s2019103562@centos8-linux Mon May 03 09:51 AM lab13]$ gcc lru.c -o lru
[s2019103562@centos8-linux Mon May 03 09:52 AM lab13]$ ./lru
Enter the number of frames:     3
Enter the number of pages:      20
Enter the pages:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7       -1      -1
7       0       -1
7       0       1
2       0       1
2       0       1
2       0       3
2       0       3
4       0       3
4       0       2
4       3       2
0       3       2
0       3       2
0       3       2
1       3       2
1       3       2
1       0       2
1       0       2
1       0       7
1       0       7
1       0       7
Total number of Page Faults:12
[s2019103562@centos8-linux Mon May 03 09:52 AM lab13]$
```

```
[s2019103562@centos8-linux Mon May 03 09:52 AM lab13]$ cat lru.c
#include<stdio.h>
int findmin(int time[],int n){
        int i,minimum=time[0],pos=0;
        for(i=1;i<n;i++){
                if(time[i]<minimum){
                        minimum=time[i];
                        pos=i;
                }
        }
        return pos;
}
int main(){
        int i,j,k,flag1,flag2,faults=0,counter=0,time[10],pages[50],frames[10],no_of_frames,no_of_pages,pos;
        printf("Enter the number of frames:\t");
        scanf("%d",&no_of_frames);
        printf("Enter the number of pages:\t");
        scanf("%d",&no_of_pages);
        printf("Enter the pages:\n");
        for(i=0;i<no_of_pages;i++)
                scanf("%d",&pages[i]);
        for(j=0;j<no_of_frames;j++)
                frames[j]=-1;
        for(i=0;i<no_of_pages;i++){
                flag1=flag2=0;
                for(j=0;j<no_of_frames;j++){
                        if(frames[j]==pages[i]){
                                counter++;
                                time[j]=counter;
                                flag1=flag2=1;
                                break;
                        }
                }
```

```
                if(flag1==0){
                        for(j=0;j<no_of_frames;j++){
                                if(frames[j]==-1){
                                        counter++;
                                        faults++;
                                        frames[j]=pages[i];
                                        time[j]=counter;
                                        flag2=1;
                                        break;
                                }
                        }
                }
                if(flag2==0){
                        pos=findmin(time,no_of_frames);
                        counter++;
                        faults++;
                        frames[pos]=pages[i];
                        time[pos]=counter;
                }
                printf("\n");
                for(j=0;j<no_of_frames;j++)
                        printf("%d\t",frames[j]);
        }
        printf("\nTotal number of Page Faults:%d\n",faults);
        return 0;
}


[s2019103562@centos8-linux Mon May 03 09:52 AM lab13]$
```