**Exp 8**

**1. FIFO**

//MAIN PROGRAM:

// PAGE REPLACEMENT BY FIFO

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void main()

{

int page[20]={0},frame[10]={0},flag=0,hit=0,i=0,j=0,n=0,l=0,k=0,fault=0,nf;

printf("ENTER THE NUMBER OF PAGES:");

scanf("%d",&n);

printf("\n ENTER NO OF FRAMES:");

scanf("%d",&nf);

printf("ENTER THE SEQUENCE OF PAGES");

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

{

scanf("%d",&page[i]);

}

for(i=0;i<nf;i++)

{

frame[i]=-1;//FRAMES ARE SET TO -1

}

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

{

flag=0;//CHECK AVAILABILITY

for(j=0;j<nf;j++)

{

if(frame[j]==page[i])

{

flag=1;//HIT OPERATION

hit++;

printf("\n");

}

}

if(flag==0)//PAGE FAULT WITH NO SPACE AVAILABLE IN FRAME BY FIFO

{

printf("\n");

frame[l]=page[i];

l=(l+1)%nf;

}

for(k=0;k<nf;k++)

printf("\t%d",frame[k]);

}

printf("\n");

fault=n-hit;

printf("\nNO. OF HITS=%d",hit);

printf("\nNO. OF FAULTS=%d",fault);

printf("\n");

}

**Output:**

etc@CETCL04-03:~/Downloads$ gcc fifo1.c

etc@CETCL04-03:~/Downloads$ ./a.out

ENTER THE NUMBER OF PAGES:12

ENTER NO OF FRAMES:3

ENTER THE SEQUENCE OF PAGES2

3

2

1

5

2

4

5

3

2

5

2

2 -1 -1

2 3 -1

2 3 -1

2 3 1

5 3 1

5 2 1

5 2 4

5 2 4

3 2 4

3 2 4

3 5 4

3 5 2

NO. OF HITS=3

NO. OF FAULTS=9;

**2. LRU**

**Code:**

//PAGE REPLACEMENT USING LRU ALGORITHM

#include<stdio.h>

int search(int frames[],int nf, int pages)//to check availability

{

int j;

for(j=0;j<nf;j++)

if(frames[j]==pages)

return(1);

return(0);

}

int findmax(int dist[],int nf)//to find max location

{

int i;

int max=0;

for(i=0;i<nf;i++)

if(dist[i]>max)

max=dist[i];

for(i=0;i<nf;i++)

if(max==dist[i])

return(i);

}

int findempty(int frames[],int nf)//to find empty location of frame

{

int j;

for(j=0;j<nf;j++)

if(frames[j]==-1)

return(j);

return(-1);

}

void main()

{

int frames[20],pages[30],np,nf,max;

int i,j,k,loc,dist[10],hits=0;

printf("\nEnter number of pages");

scanf("%d",&np);

printf("\nEnterthe number of frames\n");

scanf("%d",&nf);

printf("\n Enter page sequence\n");

for(i=0;i<np;i++)

{

scanf("%d",&pages[i]);

}

for(j=0;j<nf;j++)

{

frames[j]=-1;

dist[j]=5;//ANY INTEGER CAN BE USED BUT KEEP AS MAX AS POSSIBLE

}

printf("\n pageno \t LRU allocation");

for(i=0;i<np;i++)

{

if(!search(frames,nf,pages[i]))//to check page exist or not

{

loc=findempty(frames,nf);

if(loc!=-1)

{

for(k=0;k<nf;k++)

{

if(dist[k]!=5)

dist[k]++;

}

frames[loc]=pages[i];

dist[loc]=0;

}

else//works for page fault with no space available in frame

{

for(k=0;k<nf;k++)

//printf("%d",dist[k]);

loc=findmax(dist,nf);

//printf("%d",loc);

frames[loc]=pages[i];

for(j=0;j<nf;j++)

dist[j]++;

dist[loc]=0;

}

}

else//hit operation

{

hits++;

for(k=0;k<nf;k++)

{

if(frames[k]==pages[i])

dist[k]=0;

else if(dist[k]!=5)

dist[k]++;

}

}

printf("\n%d",pages[i]);

for(j=0;j<nf;j++)

printf("\t %d",frames[j]);

}

printf("\nHits = %d",hits);

printf("\nfaults = %d",np-hits);

}

**Output:**

etc@CETCL04-05:~/Desktop$ gcc lrudone.c

etc@CETCL04-05:~/Desktop$ ./a.out

Enter number of pages12

Enterthe number of frames

3

Enter page sequence

2

3

2

1

5

2

4

5

3

2

5

2

pageno LRU allocation

2 2 -1 -1

3 2 3 -1

2 2 3 -1

1 2 3 1

5 2 5 1

2 2 5 1

4 2 5 4

5 2 5 4

3 3 5 4

2 3 5 2

5 3 5 2

2 3 5 2

Hits = 5

faults = 7