Name : Yash Shastri

TECOC64
Subject : SPOS

Assignment 4

INPUT :

```
import java.io.*;
public class FIFO {
public static void main(String[] args) throws IOException
{
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
int frames, pointer = 0, hit = 0, fault = 0,ref_len;
int buffer[];
int reference[];
int mem_layout[][];
System.out.println(&quot;Please enter the number of Frames: &quot;);
frames = Integer.parseInt(br.readLine());
System.out.println(&quot;Please enter the length of the Reference string:
&quot;);
ref_len = Integer.parseInt(br.readLine());
reference = new int[ref_len];
mem_layout = new int[ref_len][frames];
buffer = new int[frames];
for(int j = 0; j &lt; frames; j++)
buffer[j] = -1;
System.out.println(&quot;Please enter the reference string: &quot;);
for(int i = 0; i &lt; ref_len; i++)
{
reference[i] = Integer.parseInt(br.readLine());
}
System.out.println();
for(int i = 0; i &lt; ref_len; i++)
{
int search = -1;
for(int j = 0; j &lt; frames; j++)
{
if(buffer[j] == reference[i])
{
search = j;
hit++;
break;
}
}
if(search == -1)
{
buffer[pointer] = reference[i];
fault++;
pointer++;
if(pointer == frames)
pointer = 0;
```

```
}
for(int j = 0; j &lt; frames; j++)
mem_layout[i][j] = buffer[j];

}
for(int i = 0; i &lt; frames; i++)
{
for(int j = 0; j &lt; ref_len; j++)
System.out.printf(&quot;%3d &quot;,mem_layout[j][i]);
System.out.println();
}
System.out.println(&quot;The number of Hits: &quot; + hit);
System.out.println(&quot;Hit Ratio: &quot; + (float)((float)hit/ref_len));
System.out.println(&quot;The number of Faults: &quot; + fault);
}
}
```

output:-

```
Please enter the number of Frames:
3
Please enter the length of the Reference string:
20
Please enter the reference string:
7
0
1
2
0
3
0
4
2
3
0
3
2
1
2
0
1
7
0
1
7 7 7 2 2 2 2 4 4 4 0 0 0 0 0 0 0 7 7
7
-1 0 0 0 0 3 3 3 2 2 2 2 2 1 1 1 1 1 0
0
-1 -1 1 1 1 1 0 0 0 3 3 3 3 3 2 2 2 2 2
1
The number of Hits: 5
Hit Ratio: 0.25
The number of Faults: 15
```

```
import java.io.*;
class FIFO
{
 public static void main(String args[]) throws IOException
 {
int n;
 int f;
 float rat;
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println(&amp;quot;Enter the number of FRAMES :&amp;quot;);
f=Integer.parseInt(br.readLine());
int fifo[]=new int[f];
System.out.println(&amp;quot;Enter the number of INPUTS :&amp;quot;);
 n=Integer.parseInt(br.readLine());
 int inp[]=new int[n];
System.out.println(&amp;quot;Enter INPUT:&amp;quot;);
for(int i=0;i&amp;lt;n;i++)
inp[i]=Integer.parseInt(br.readLine());
System.out.println(&amp;quot; ---------- &amp;quot;);
for(int i=0;i&amp;lt;f;i++)
fifo[i]=-1;
int Hit=0;
int Fault=0;
int j=0;
boolean check;
for(int i=0;i&amp;lt;n;i++)
{
 check=false;
for(int k=0;k&amp;lt;f;k++)
 if(fifo[k]==inp[i])
 {
check=true;
Hit=Hit+1;
 }
 if(check==false)
 {
fifo[j]=inp[i];
j++;

 if(j&amp;gt;=f)
j=0;
Fault=Fault+1;
}
}
 rat = (float)Hit/(float)n;
System.out.println(&amp;quot;HIT:&amp;quot;+Hit+&amp;quot; FAULT:&amp;quot;+Fault+&amp;quot; HIT
RATIO:&amp;quot;+rat);
}
}
/*
```

OUTPUT:

First In First Out (FIFO) page replacement algorithm Output:
run:
Enter the number of FRAMES :
3
Enter the number of INPUTS :
12
Enter INPUT:
1
2
3
4
1
2
5
1
2
3
4
5

HIT:3 FAULT:9 HIT RATIO:0.25
BUILD SUCCESSFUL (total time: 37 seconds)

LRU

INPUT :

```
import java.util.*;
class LruAlgo
{
int p[],n,fr[],m,fs[],index,k,l,flag1=0,flag2=0,pf=0,frsize=3,i,j;
Scanner src=new Scanner(System.in);
void read()
{
System.out.println(&quot;Enter page table size&quot;);
n=src.nextInt();
p=new int[n];
System.out.println(&quot;Enter element in page table&quot;);
for(int i=0;i&lt;n;i++)
p[i]=src.nextInt();
System.out.println(&quot;Enter page frame size&quot;);
m=src.nextInt();
fr=new int[m];
fs=new int[m];
}
void display()
{
```

```
System.out.println(&quot;\n&quot;);
for(i=0;i&lt;m;i++)
{
if(fr[i]==-1)
System.out.println(&quot;[ ]&quot;);
else
System.out.println(&quot;[&quot;+fr[i]+&quot;]&quot;);
}
}
void lru()
{
for(i=0;i&lt;m;i++)
 {
fr[i]=-1;
}

for(j=0;j&lt;n;j++)
 {
flag1=0;flag2=0;
for(i=0;i&lt;m;i++)
 {
if(fr[i]==p[j])
{
flag1=1;
flag2=1;
break;
}
}
if(flag1==0)
{
for(i=0;i&lt;m;i++)
 {
if(fr[i]==-1)
{
fr[i]=p[j];
flag2=1;
break;
}
}
}
if(flag2==0)
{
for(i=0;i&lt;3;i++)
fs[i]=0;
for(k=j-1,l=1;l&lt;=frsize-1;l++,k--)
{
for(i=0;i&lt;3;i++)
{
if(fr[i]==p[k])
fs[i]=1;
}
}
```

```
for(i=0;i&lt;3;i++)
{
if(fs[i]==0)
index=i;
}
fr[index]=p[j];
pf++;
}
System.out.print(&quot;Page : &quot;+p[j]);
display();

}
System.out.println(&quot;\n no of page faults :&quot;+pf);
}
public static void main(String args[])
{
LruAlgo a=new LruAlgo();
a.read();
a.lru();
a.display();
}
}
```

OUTPUT:

```
D:\My Documents\college assg\o.s\programs&gt;javac LruAlgo.java
D:\My Documents\college assg\o.s\programs&gt;java LruAlgo
Enter page table size
10
Enter element in page table
1
5
1
2
6
2
7
1
5
1
Enter page frame size
3
Page : 1
[1]
[ ]

[ ]
Page : 5
[1]
[5]
[ ]
```

Page : 1
[1]
[5]
[ ]
Page : 2
[1]
[5]
[2]
Page : 6
[1]
[6]
[2]
Page : 2
[1]
[6]
[2]
Page : 7
[7]
[6]
[2]
Page : 1
[7]
[1]
[2]
Page : 5
[7]
[1]

[5]
Page : 1
[7]
[1]
[5]
no of page faults :4

[7]
[1]
[5]

Optimal Page Replacement algorithm

INPUT:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class OptimalReplacement {
public static void main(String[] args) throws IOException
{
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
```

```
int frames, pointer = 0, hit = 0, fault = 0,ref_len;
boolean isFull = false;
int buffer[];
int reference[];
int mem_layout[][];
System.out.println(&quot;Please enter the number of Frames: &quot;);
frames = Integer.parseInt(br.readLine());
System.out.println(&quot;Please enter the length of the Reference string:
&quot;);
ref_len = Integer.parseInt(br.readLine());
reference = new int[ref_len];
mem_layout = new int[ref_len][frames];
buffer = new int[frames];
for(int j = 0; j &lt; frames; j++)
buffer[j] = -1;
System.out.println(&quot;Please enter the reference string: &quot;);
for(int i = 0; i &lt; ref_len; i++)
{
reference[i] = Integer.parseInt(br.readLine());
}
System.out.println();
for(int i = 0; i &lt; ref_len; i++)
{
int search = -1;
for(int j = 0; j &lt; frames; j++)
{
if(buffer[j] == reference[i])
{
search = j;
hit++;
break;
}
}
if(search == -1)
{
if(isFull)
{
int index[] = new int[frames];
boolean index_flag[] = new boolean[frames];
for(int j = i + 1; j &lt; ref_len; j++)

{
for(int k = 0; k &lt; frames; k++)
{
if((reference[j] == buffer[k]) &amp;&amp; (index_flag[k] == false))
{
index[k] = j;
index_flag[k] = true;
break;
}
}
}
}
```

```
int max = index[0];
pointer = 0;
if(max == 0)
max = 200;
for(int j = 0; j &lt; frames; j++)
{
if(index[j] == 0)
index[j] = 200;
if(index[j] &gt; max)
{
max = index[j];
pointer = j;
}
}
}
buffer[pointer] = reference[i];
fault++;
if(!isFull)
{
pointer++;
if(pointer == frames)
{
pointer = 0;
isFull = true;
}
}
}
for(int j = 0; j &lt; frames; j++)
mem_layout[i][j] = buffer[j];
}
for(int i = 0; i &lt; frames; i++)
{
for(int j = 0; j &lt; ref_len; j++)
System.out.printf(&quot;%3d &quot;,mem_layout[j][i]);
System.out.println();
}
System.out.println(&quot;The number of Hits: &quot; + hit);
System.out.println(&quot;Hit Ratio: &quot; + (float)((float)hit/ref_len));
System.out.println(&quot;The number of Faults: &quot; + fault);
}
}
```

OUTPUT :

```
Please enter the number of Frames:

3
Please enter the length of the Reference string:
20
Please enter the reference string:
1
```

2
3
2
1
5
2
1
6
2
5
6
3
1
3
6
1
2
4
3
1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 2 4
4
-1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
1
-1 -1 3 3 3 5 5 5 5 5 5 5 3 3 3 3 3 3 3
3
The number of Hits: 11
Hit Ratio: 0.55
The number of Faults: 9