Smt. Chandibai Himathmal Mansukhani College

USCSP301 : USCS303 - Operating System (OS)

Contents

Smt. Chandibai Himathmal Mansukhani College

USCS3P01:USCS303 – Operating System (OS) Practical_09

Practical Date: 31st August 2021

Practical Aim: Page Replacement Algorithm : LRU

Page Replacement Algorithm : LRU

- **Content**: In page replacement algorithm the page that has not been used for the longest Period of the time is chosen and replaced.
- **Process**: Implement LRU algorithm and find out page hits and page faults.
- **Prior Knowledge**: Page Replacement Algorithm.

Page Replacement Algorithm

- In demand paging memory management technique, if a page demanded for execution is not present in main memory, then a page fault occurs.
- To load the page in demand into main memory, a free page frame is search in main memory and allocated.
- If no page frame is free, Memory Manager has to free a frame by swapping its content to secondary storage and thus make room for the required page.
- To swap pages many screens of strategies are used.

Least Recently Used (LRU)

- The Least Recently used (LRU) algorithm replaces the page that has not been used for the longest period of time.
- It is based on the observation that pages that have not been used for the long time with probably remain unused for the longest time and are to be replaced.

Smt. Chandibai Himathmal Mansukhani College

Solved Example:

- Apply the LRU replacement algorithms for the following page-reference strings :7,0 , 1 , 2 ,0 , 3 ,0 , 4 ,2 ,3 ,0 ,3 ,2 .
- Indicate the number of page faults for LRU you algorithm assuming demand paging with four frames.
- Find the number of hits, number of faults and hit ratio.

Page-Reference String :7,0 , 1 , 2 ,0 , 3 ,0 , 4 ,2 ,3 ,0 ,3 ,2

Demand Paging or Number of Frames: 4

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

7	0	1	2	0	3	0	4	2	3	0	3	2
W	W	W	W	R	W	R	W	R	R	R	R	R

Number of Hits: count of number replacements = 7

Number of faults: count of replacements = 6 [R]

Hit Ratio: Number of hits / len(Ref string)= 7/13 = 0.53846157 [W]

Do it Yourself 1 :

Consider the following example 3 frames with 1 ,3 ,0 ,3 ,5 ,6 ,3 page-reference strings.

Find the number of hits, number of faults and hit ratio using page using LRU Page Replacement Algorithm.

Do it Yourself 2 :

Consider the following example 3 frames with 7 ,0 ,1 ,2 ,0 ,3 ,0 ,4 ,2 ,3 ,0 ,3 ,2 ,1 ,2 ,0 ,1 ,7 ,0 ,1 page-reference strings.

Find the number of hits, number of faults and hit ratio using page using LRU Page Replacement Algorithm

Question:

Write a Java Program that implements the LRU page-replacement algorithm.

Implementation

```
// Name: Gaurang Sanyasi
// Batch: B2
// PRN: 2020016400785461
// Date: 31st August, 2021
// Prac-08: Page Replacement Algorithm LRU

import java.io.*;
import java.util.*;
public class P9_PR_LRU_GS
{
public static void main(String[] args) throws IOException
{
Scanner scan = new Scanner(System.in);
int frames,pointer = 0,hit = 0,fault = 0,ref_len;
Boolean isFull = false;
int buffer[];
```

```
ArrayList<Integer> stack = new ArrayList<Integer>();
int reference[];
int mem_layout[][];
System.out.print("Please enter the number of frames:");
frames = scan.nextInt();
System.out.print("Please enter the length ofReference string: ");
ref_len = scan.nextInt();
reference = new int[ref_len];
mem_layout = new int [ref_len][frames];
buffer = new int[frames];
for(int j=0;j<frames;j++)
buffer[j] = -1;
System.out.println("Please enter the reference string:");
for(int i = 0;i<ref_len;i++)
{
reference[i] = scan.nextInt();
}
System.out.println();
for (int i=0;i<ref_len;i++)
{
if(stack.contains(reference[i]))
{
stack.remove(stack.indexOf(reference[i]));
}
stack.add(reference[i]);
int search = -1;
for(int j =0;j<frames;j++)
{
if(buffer[j]==reference[i])
{
search = j;
hit++;
break;
}
}
if(search==-1)
{
if(isFull)
{
int min_loc = ref_len;
for(int j = 0;j<frames;j++)
{
if(stack.contains(buffer[j]))
```

```
{
int temp=stack.indexOf(buffer[j]);
if(temp<min_loc)
{
min_loc=temp;
pointer=j;
}
}
}
}
buffer[pointer]=reference[i];
fault++;
pointer++;
if(pointer==frames)
{
pointer=0;
isFull=true;
}
}
for(int j=0;j<frames;j++)
mem_layout[i][j] = buffer[j];
}
for(int i=0;i<frames;i++)
{
for(int j=0;j<ref_len;j++)
System.out.printf("%3d",mem_layout[j][i]);
System.out.println();
}
System.out.println("The number of Hits:" +hit);
System.out.println("HitRatio:"+(float)((float)hit/ref_len));
System.out.println("The number of faults:"+fault);
}
}
```

Input:

```
E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>javac P9_PR_LRU_GS.java

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>java P9_PR_LRU_GS
Please enter the number of frames:4
Please enter the length ofReference string: 13
Please enter the reference string:
7 0 1 2 0 3 0 4 2 3 0 3 2
```

Output:

```
  7  7  7  7  7  3  3  3  3  3  3  3  3
 -1  0  0  0  0  0  0  0  0  0  0  0  0
 -1 -1  1  1  1  1  1  4  4  4  4  4  4
 -1 -1 -1  2  2  2  2  2  2  2  2  2  2
The number of Hits:7
HitRatio:0.53846157
The number of faults:6
```

Sample Output01:

```
E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>javac P9_PR_LRU_GS.java

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>java P9_PR_LRU_GS
Please enter the number of frames:4
Please enter the length ofReference string: 13
Please enter the reference string:
7 0 1 2 0 3 0 4 2 3 0 3 2

  7  7  7  7  7  3  3  3  3  3  3  3  3
 -1  0  0  0  0  0  0  0  0  0  0  0  0
 -1 -1  1  1  1  1  1  4  4  4  4  4  4
 -1 -1 -1  2  2  2  2  2  2  2  2  2  2
The number of Hits:7
HitRatio:0.53846157
The number of faults:6

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>
```

Input2:

```
E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>javac P9_PR_LRU_GS.java

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>java P9_PR_LRU_GS
Please enter the number of frames:3
Please enter the length ofReference string: 7
Please enter the reference string:
1 3 0 3 5 6 3
```

Output2:

```
  1  1  1  1  5  5  5
 -1  3  3  3  3  3  3
 -1 -1  0  0  0  6  6
The number of Hits:2
HitRatio:0.2857143
The number of faults:5
```

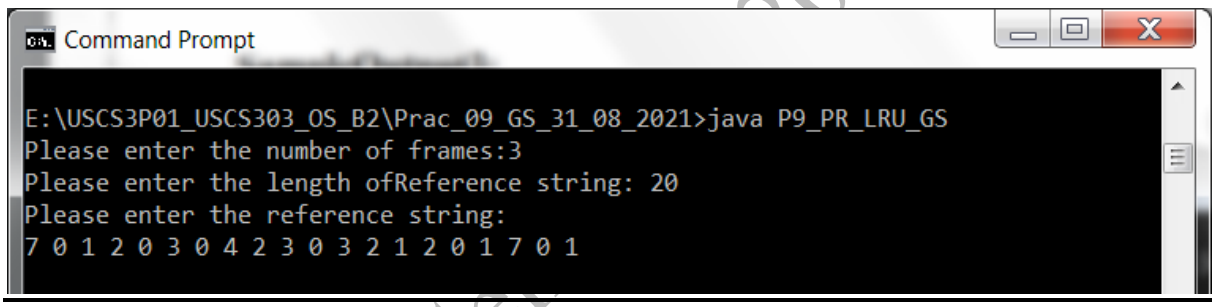
Sample Output02:

```
E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>javac P9_PR_LRU_GS.java

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>java P9_PR_LRU_GS
Please enter the number of frames:3
Please enter the length ofReference string: 7
Please enter the reference string:
1 3 0 3 5 6 3

  1  1  1  1  5  5  5
 -1  3  3  3  3  3  3
 -1 -1  0  0  0  6  6
The number of Hits:2
HitRatio:0.2857143
The number of faults:5
```

Input:



Output:

```
 7  7  7  2  2  2  2  4  4  4  0  0  0  1  1  1  1  1  1  1
-1  0  0  0  0  0  0  0  0  3  3  3  3  3  3  0  0  0  0  0
-1 -1  1  1  1  3  3  3  2  2  2  2  2  2  2  2  2  7  7  7
The number of Hits:8
HitRatio:0.4
The number of faults:12
```

SampleOutput3:

```
E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>java P9_PR_LRU_GS
Please enter the number of frames:3
Please enter the length ofReference 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  1  1  1  1  1  1  1
 -1  0  0  0  0  0  0  0  0  3  3  3  3  3  3  0  0  0  0  0
 -1 -1  1  1  1  3  3  3  2  2  2  2  2  2  2  2  7  7  7
The number of Hits:8
HitRatio:0.4
The number of faults:12

E:\USCS3P01_USCS303_OS_B2\Prac_09_GS_31_08_2021>
```