Smt. Chandibai Himathmal Mansukhani College

USCSP301-USCS303: Operating System (OS) Practical-09

Practical-09 : Page Replacement Algorithm LRU

Contents

Name: Sahil Jadhav **Date:31-08-2021** **Batch: B2**

Smt. Chandibai Himathmal Mansukhani College

Practical Date : 31-08-2021 (Tuesday)
Practical Aim : Page Replacement Algorithm LRU

- **Content:**

- In LRU 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.

Algorithm :

- **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 searching 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.

- The Least Recently used (LRU) algorithm replaces the page that has not been used for the longest period of time.

Solved Example :

Question – 01

- 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	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	2	2	2	2	2	2	2	2	2	2	2

7	0	1	2	0	3	0	4	2	3	0	3	2
×	×	×	×	✓	×	✓	×	✓	✓	✓	✓	✓

Number of Hits: count of number replacements = 7

Number of faults: count of replacements = 6

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

Question – 02

- 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.

Page-Reference String : 1 , 3 , 0 , 3 , 5 , 6 , 3

Demand Paging or Number of Frames : 3

Number of Hits: 2

Number of faults: 5

Hit Ratio : 2/7 = 0.2857

Question – 03

- 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

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

Demand Paging or Number of Frames : 3

Number of Hits: 8

Number of faults: 12

Hit Ratio : 8/20 = 0.4

Question:

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

Implementation:

```
//Name:sahil jadhav
//Batch No:B2
//PRN:2020016400783091
//Date:31-08-2021
//Prac-09:Page Replacement Algorithm LRU

import java.io.*;
import java.util.*;
public class P9_PR_LRU_SJ
{
	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 of Reference 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("Hit Ratio:"+(float)((float)hit/ref_len));
        System.out.println("The number of faults:"+fault);
 }
}
```

Input:

Please enter the number of frames : 4

Please enter the length of Reference 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**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**

Number of Hits: 7

Number of faults: 6

Hit Ratio : 7/13 = 0.53846157

Sample Output:

Sample Output – 01

Command Prompt

```
C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>javac P9_PR_LRU_SJ.java

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>java P9_PR_LRU_SJ
Please enter the number of Frames: 4
Please enter the length of the References strings: 13
Please enter the references strings: 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
Hit Ratio: 0.53846157
The number of Faults: 6

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>
```

Sample Output – 02

Command Prompt

```
C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>javac P9_PR_LRU_SJ.java

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>java P9_PR_LRU_SJ
Please enter the number of Frames: 3
Please enter the length of the References strings: 7
Please enter the references strings: 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
Hit Ratio: 0.2857143
The number of Faults: 5

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>
```

Sample Output – 03

Command Prompt

```
        at P9_PR_LRU_SJ.main(P9_PR_LRU_SJ.java:33)

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>javac P9_PR_LRU_SJ.java

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>java P9_PR_LRU_SJ
Please enter the number of Frames: 3
Please enter the length of the References strings: 20
Please enter the references strings: 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  2  7  7  7
The number of Hits: 8
Hit Ratio: 0.4
The number of Faults: 12

C:\USCSP301\USCS303_OS_B2\prac_09_SJ_LRU>
```