Table of Content

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

Practical-09: Page Replacement Algorithm: LRU

Practical Date: 30th Aug, 2021

Practical Aim: Page Replacement Algorithm: LRU

Algorithm.....................................................................................................................2

Solved Example............................................................................................................3

Question........................................................................................................................4

Implementation..............................................................................................................4

Input...............................................................................................................................7

Output............................................................................................................................7

Sample Output................................................................................................................7

**Practical – 09:** 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.
* **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 searched in main memory and allocated.
* If no page frame is free, memory manager has to free a frame by swapping its contents to secondary storage and thus make room for the required page.
* To swap pages, many schemes or 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 long time will probably remain unused for the longest time and are to be replaced.

## Example

* Apply the LRU replacement algorithm for the following page-reference strings: 7,0,1,2,0,3,0,4,2,3,0,3,2.
* Indicate the number 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 | -1 | 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 no replacements = 7

**Number of Faults:** count of replacements = 6

**Hit Ratio:** Number of Hits/Len(Ref String) = 7/13 = 0.53846157

**Question:**

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

**Source Code:**

//NAME: SHRADDHA SAWANT

//BATCH: B1

//PRN: 2020016400773862

//DATE: 30th Aug, 2021

//PRAC-08: PAGE REPLACEMENT ALGORITHM

import java.io.\*;

import java.util.\*;

public class P8\_PR\_FIFO\_SS

{

public static void main(String[] args) throws IOException

{

Scanner scan = new Scanner(System.in);

int frames, pointer = 0, hit =0, fault = 0, ref\_len;

int buffer[];

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 the 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++)

{

int search =-1;

for(int j=0; j<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<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);

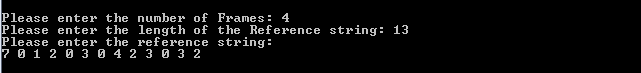
System.out.println("Hit Ratio: " +(float)((float)hit/ref\_len));

System.out.println("The number of Faults: "+fault);

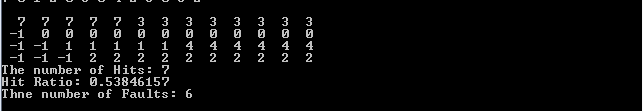
}

}

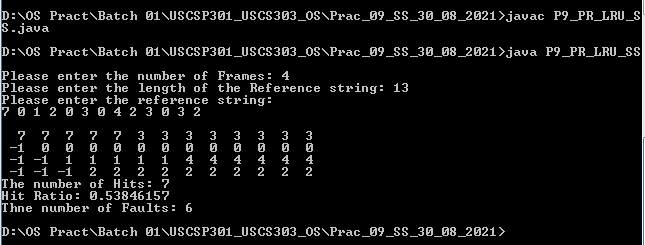
**Input:**



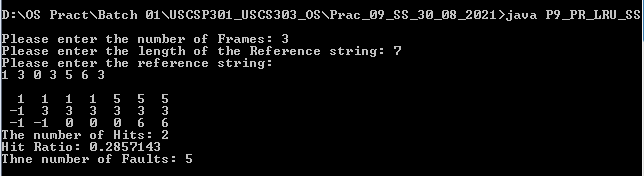
**Output:**



**Sample Output – 01:**



**Sample Output – 02:**



**Sample Output – 03:**

