Smt. Chandibai Himathmal Mansukhani College

Contents

USCS3P01:USCS303-Operating System (OS) Practical-08 2

Practical – 08: Page Replacement Algorithm First In First Out (FIFO) 2

Practical Date: 30-08-2021 2

Practical Aim: Page Replacement Algorithm (FIFO) 2

Algorithm: 2

Page Replacement Algorithm 2

First In First Out (FIFO) Algorithm 2

Solved Example: 3

Example-01: 3

Example-02: 4

Example-03: 5

Question: 6

Implementation: 6

Input 01: ..... **Error! Bookmark not defined.**

Output 01: ..... **Error! Bookmark not defined.**

Sample Output 01: ..... **Error! Bookmark not defined.**

Input 02: ..... **Error! Bookmark not defined.**

Output 02: ..... **Error! Bookmark not defined.**

Sample Output 02: ..... **Error! Bookmark not defined.**

Input 03: ..... **Error! Bookmark not defined.**

Output 03: ..... **Error! Bookmark not defined.**

Sample Output 03: ..... **Error! Bookmark not defined.**

USCS3P01:USCS303-Operating System (OS) Practical-08

Practical – 08: Page Replacement Algorithm First In First Out (FIFO)

Practical Date: 30-08-2021

Practical Aim: Page Replacement Algorithm (FIFO)

Algorithm:

Page Replacement Algorithm

In operating systems that use paging for memory management, **page replacement algorithm** are needed to decide which page needed to be replaced when new page comes in.

Whenever a new page is referred and not present in memory, page fault occurs and Operating System replaces one of the existing pages with newly needed page.

Different page replacement algorithms suggest different ways to decide which page to replace.

The target for all algorithms is to reduce number of page faults.

Page Fault – A page fault happens when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

Step 1: First of all, find the location of the desired page on the disk.

Step 2: Find a free Frame:

Step 2.1: If there is a free frame, then use it.

Step 2.2: If there is no free frame then make use of the page-replacement algorithm in order to select the victim frame.

Step 2.3: Then after that write the victim frame to the disk and then make the changes in the page table and frame table accordingly.

Step 3: After that read the desired page into the newly freed frame and then change the page and frame tables.

Step 4: Restart the process.

First In First Out (FIFO) Algorithm

It is a very simple way of Page replacement and is referred to as **First In First Out (FIFO).**

This algorithm mainly replaces the oldest page that has been present in the main memory for the longest time.

This algorithm is implemented by keeping the track of all the pages in the queue.

As new pages are requested and are swapped in, they are added to the tail of a queue and the page which is at the head becomes the victim.

This is not an effective way of page replacement but it can be used for small systems.

Solved Example:

Example-01:

Apply the FIFO replacement algorithms for the following page-reference strings: 0,2,1,6,4,0,1,0,3,1,2,1.

Indicate the number of page faults for FIFO algorithm assuming demand paging with four frames.

Find the number of hits, number of faults and hit ratio.

Solution:

Page Reference String: 0,2,1,6,4,0,1,0,3,1,2,1

Demand Paging or Number of Frames: 4

0	**0**	**0**	**0**	**4**	**4**	**4**	**4**	**4**	**4**	**2**	**2**
-1	**2**	**2**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
-1	**-1**	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**3**	**3**	**3**
-1	**-1**	**-1**	**6**	**6**	**6**	**6**	**6**	**6**	**1**	**1**	**1**

0	**2**	**1**	**6**	**4**	**0**	**1**	**0**	**3**	**1**	**2**	**1**
✖	✖	✖	✖	✖	✖	✔	✔	✖	✖	✔	✔

Number of Hits: Count of no replacements = 3 ✔

Number of Faults: Count of replacements = 9 ✖

Hit Ratio: Number of Hits/Len (Ref String) = 3/12 = 0.25

Example-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 FIFO Page Replacement Algorithm.

Solution:

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

Demand Paging or Number of Frames: 3

1	**1**	**1**	**1**	**5**	**5**	**5**
-1	**3**	**3**	**3**	**3**	**6**	**6**
-1	**-1**	**0**	**0**	**0**	**0**	**3**

1	**3**	**0**	**3**	**5**	**6**	**3**
✖	✖	✖	✖	✖	✖	✔

Number of hits: Count of no replacements = 1 ✔

Number of Faults: Count of replacements = 6 ✖

Hit Ratio: Number of Hits/Len (Ref String) = 1/7= 0.14

Example-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 FIFO Page Replacement Algorithm.

Solution:

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

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

7	**0**	**1**	**2**	**0**	**3**	**0**	**4**	**2**	**3**	**0**	**3**	**2**	**1**	**2**	**0**	**1**	**7**	**0**	**1**
✖	✖	✖	✖	✔	✖	✖	✖	✖	✖	✖	✔	✔	✖	✔	✔	✖	✖	✖	✖

Number of Hits: Count of no replacements = 5 ✔

Number of Faults: Count of replacements = 15 ✖

Hit Ratio: Number of Hits/Len (Ref String) = 5/20 = 0.25

Question:

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

Implementation:

```
//Name: ABHISHEK NIKAM
// Batch: B2
// PRN: 2020016400805951
// Date: 31 August,2021
// Prac-08: Page Replacement Algorithm FIFO

import java.io.*;
import java.util.*;
public class P8_PR_FIFO_AN
{
	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 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++)
{
        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)
{
        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 01 :

```
Please enter the number of Frames: 4
Please enter the length of the Reference string: 12
Please enter the reference string:
0 2 1 6 4 0 1 0 3 1 2 1
```

Output 01 :

```
  0  0  0  0  4  4  4  4  4  4  2  2
 -1  2  2  2  2  0  0  0  0  0  0  0
 -1 -1  1  1  1  1  1  1  3  3  3  3
 -1 -1 -1  6  6  6  6  6  6  1  1  1
The number of Hits: 3
Hit Ratio: 0.25
The number of Faults: 9
```

Sample output 01 :

```
Please enter the number of Frames: 4
Please enter the length of the Reference string: 12
Please enter the reference string:
0 2 1 6 4 0 1 0 3 1 2 1

  0  0  0  0  4  4  4  4  4  4  2  2
 -1  2  2  2  2  0  0  0  0  0  0  0
 -1 -1  1  1  1  1  1  1  3  3  3  3
 -1 -1 -1  6  6  6  6  6  6  1  1  1
The number of Hits: 3
Hit Ratio: 0.25
The number of Faults: 9
```

Input 02 :

```
Please enter the number of Frames: 3
Please enter the length of the Reference string: 7
Please enter the reference string:
1 3 0 3 5 6 3
```

Output 02 :

```
  1  1  1  1  5  5  5
 -1  3  3  3  3  6  6
 -1 -1  0  0  0  0  3
The number of Hits: 1
Hit Ratio: 0.14285715
The number of Faults: 6
```

Sample output 02 :

```
Please enter the number of Frames: 3
Please enter the length of the Reference 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  6  6
 -1 -1  0  0  0  0  3
The number of Hits: 1
Hit Ratio: 0.14285715
The number of Faults: 6
```

Input 03 :

```
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
```

Output 03 :

```
  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
```

Sample output 03 :

```
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
```