Contents

[**USCSP30) USCS303-Operating System (OS)** 2](#_Toc81249758)

[**Practical-08:Page Replacement Algorithm First In First Out (FIFO)** 2](#_Toc81249759)

[**Practical Date**:31st August 2021 2](#_Toc81249760)

[**Pratical Aim:** Page Replacement Algorithm First(FIFO) 2](#_Toc81249761)

[**Algorithm** 2](#_Toc81249762)

[ **Page Replacement Algorithm** 2](#_Toc81249763)

[ **First In First Out (FIFO)** 2](#_Toc81249764)

[**Solved Example** 3](#_Toc81249765)

[**Example:01** 3](#_Toc81249766)

[**Example:02** 4](#_Toc81249767)

[**Example :03** 5](#_Toc81249768)

[**Question** 6](#_Toc81249769)

[**Implementation** 6](#_Toc81249770)

[**File Name: P8\_PR\_FIFO\_SJ.java** 6](#_Toc81249771)

[**Input** 9](#_Toc81249772)

[**Ouput** 9](#_Toc81249773)

[**Sample Ouput of Example:01** 10](#_Toc81249774)

[**Sample Ouput of Example:02** 10](#_Toc81249775)

[**Sample Ouput of Example:03** 11](#_Toc81249776)

# **USCSP30) USCS303-Operating System (OS)**

## **Practical-08:Page Replacement Algorithm First In First Out (FIFO)**

### **Practical Date**:31st August 2021

### **Pratical Aim:** Page Replacement Algorithm First(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)**
* 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.

**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**

**-1**

**-1**

**-1**

**0**

**2**

**-1**

**-1**

**0**

**2**

**1**

**-1**

**0**

**2**

1 1

6

**4**

**2**

**1**

**6**

**4**

**0**

**1**

**6**

4

**0**

**1**

**6**

**4**

**0**

**1**

**6**

**4**

**0**

**3**

**6**

**4**

**0**

**3**

**1**

**2**

**0**

**3**

**1**

**2**

**0**

**3**

**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 replacement=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 faultsand hit ratio using FIFO Page Replacment Algorithm

**Solution:**

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

**Demand Paging Or Number of Frames:**7

**1**

**-1**

**-1**

**1**

**3**

**-1**

**1**

**3**

**0**

**1**

**3**

**0**

**5**

**3**

**0**

**5**

**6**

**0**

**5**

**6**

**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:**

**7**

**-1**

**-1**

**7**

**0**

**-11**

**7**

**0**

**1**

**2**

**0**

1

**2**

**0**

**1**

**2**

**3**

**1**

2

**3**

**0**

**4**

**3**

**0**

**4**

**2**

**0**

**4**

**2**

**3**

**0**

**2**

**3**

**0**

**2**

**3**

**0**

**2**

**3**

**0**

**1**

**3**

0

**1**

**2**

**0**

**1**

**2**

**0**

**1**

**2**

**7**

**1**

**2**

**7**

**0**

**2**

**7**

**0**

**1**

**🗸**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **0** | **1** | **2** | **0** | **3** | **0** | **4** | **2** | **3** | **0** | **3** | **2** | **0** | **1** | **1** | **2** | **0** | **1** | **7** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | ✔ |  |  |  |  |  |  | ✔ | ✔ |  | ✔ | ✔ |  |  |  |  |

**Number of Hits**:count of no replacements=5 ✔

**Number of Faults**:count of replacement=15 

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

### **Question**

Write a Java Program that implements the FIFO page-replacements Algorithm.

### **Implementation**

### **File Name: P8\_PR\_FIFO\_SJ.java**

// Name: Sahil Jadhav

// Batch: B2

// PRN: 2020016400783091

// Date: 31 August,2021

// Prac-08: Page Replacement Algorithm(FIFO)

import java.io.\*;

import java.util.\*;

public class P8\_PR\_FIFO\_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 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);

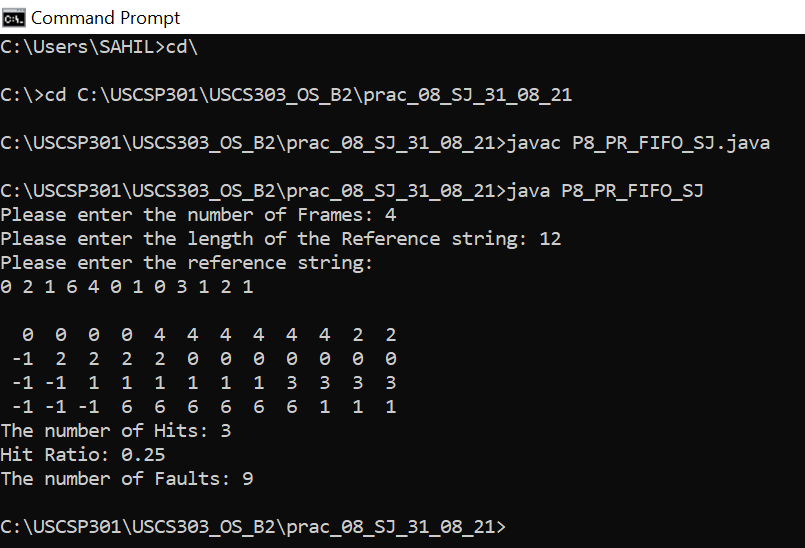
System.out.println("Hit Ratio: " + (float)((float)hit/ref\_len));

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

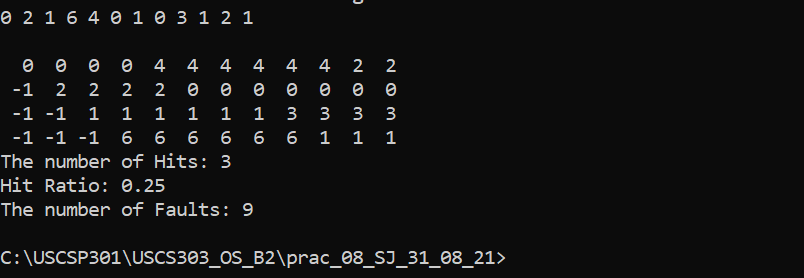
}

}

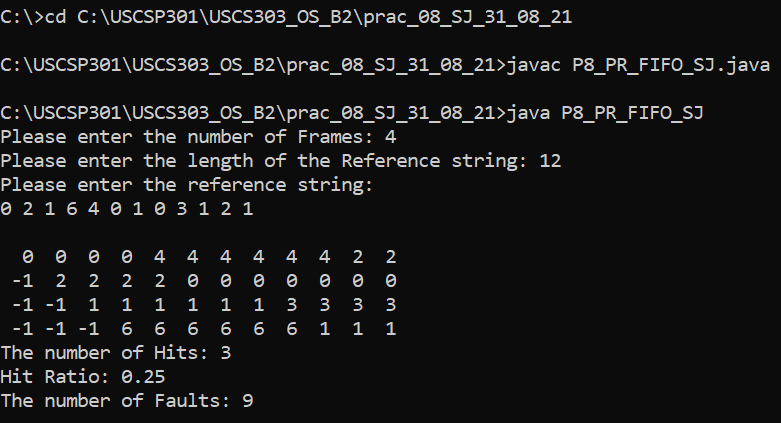
### **Input**



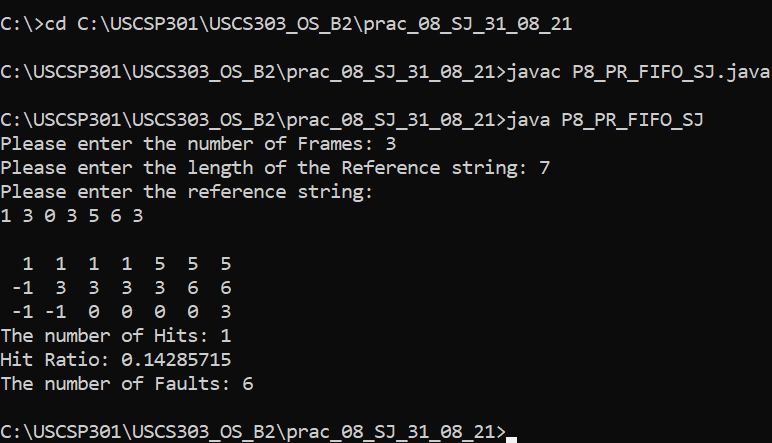
### **Ouput**



### **Sample Ouput of Example:01**



### **Sample Ouput of Example:02**



### **Sample Ouput of Example:03**

