**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code: CSE 321** | **Credits: 1.5** |
| **Course Name: Operating Systems** | **Semester: Fall 18** |

**Lab 11  
Page Replacement Algorithms**

1. **Overview:**

An operating system 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.

1. **Lesson Fit:**

Programming knowledge and concept of FIFO is required for this lab.

1. **Learning Outcome:**

After this lab, students will know how page fault is handled using First in First out Replacement (FIFO) and Least Recently Used (LRU) algorithm.

1. **Anticipated Challenges and Possible Solutions**
   1. Student may find difficulties while manipulating the variables.

**Solutions:** Students must have the knowledge about Data Structure (Queue).

1. **Acceptance and Evaluation**

Students will show their progress as they complete each task. They will be marked according to their lab performance.

**Activity Detail**

* 1. **Hour: 1  
     Discussion:**

1. Discussion on page fault.
2. Why page fault occurs? How we can handle it using Page Replacement Algorithms.
   1. **Hour: 2**
3. Discuss on programming logic of **First in First out (FIFO**) Page Replacement Algorithm.

FIFO Page Replacement technique is one of the simplest one to implement amongst other page replacement algorithms. It is a conservative algorithm. It is a low-overhead algorithm that maintains a queue to keep a track of all the pages in a memory. When a page needs to be replaced, the page at the FRONT of the Queue will be replaced. The FIFO page replacement technique is not implemented in operating systems nowadays.

1. Implement the above mentioned algorithm.
   1. **Hour: 3**
2. Discuss on **Least recently used (LRU)** replacement algorithms.

In Least Recently Used (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. The idea is based on locality of reference, the least recently used page is not likely.

1. Briefly discuss the programming concept and implement LRU algorithm in java.
2. **Home tasks**
3. If student could not finish the coding for any of the above mentioned algorithm, then it will be given as homework.
4. Implement Optimal Page Replacement Algorithm.

**Lab Activity List**

**Task 1**

Given memory capacity (as number of pages it can hold) and a string representing pages to be referred, write a function to find number of page faults.

**Example -1.** Consider page reference string 1, 3, 0, 3, 5, 6 and 3 page slots.

Initially all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> 3 **Page Faults.**

when 3 comes, it is already in memory so —> 0 **Page Faults**.

Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —>1 **Page Fault.**

Finally 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —>1 **Page Fault.**

So total page faults = **5**.

**Task 2**

Let say the page reference string 7 0 1 2 0 3 0 4 2 3 0 3 2. Initially we have 4 page slots empty.

Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots —>**4 Page faults**  
0 is already their so —> **0 Page fault.**

when 3 came it will take the place of 7 because it is least recently used —>**1 Page fault**  
0 is already in memory so —>**0 Page fault**.

4 will takes place of 1 —>**1 Page Fault**

Now for the further page reference string —>**0 Page fault** because they are already available in the memory

