**SSN COLLEGE OF ENGINEERING, KALAVAKKAM**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**II Year CSE – A & B Sections (IV Semester)**

**Academic Year 2022-23**

**UCS2412 - OPERATING SYSTEMS LAB**

------------------------------------------------------------------------------------------------------------

**Lab Exercise 9         Implementation of Page Replacement Algorithms**

**Aim:**

Develop a C program to implement the page replacement algorithms (FIFO, Optimal, LRU and LFU) using linked list.

**Algorithm:**

Implement the following modules and its operations using linked list.

**Read module:**

1. Read the number of frames.

2. Read the number of frames required by the process N.

3. Read the reference string for allocation of page frames.

**Page replacement module**:

**FIFO REPLACEMENT**

1. Allocate the first N pages into the frames and increment the page faults accordingly.

2. When next frame in the reference string is not already available in the allocated list do  a. Look for the oldest one in the allocated frames and replace it with the next page     frame.

b. Increment the page fault whenever a frame is replaced.

**OPTIMAL REPLACEMENT**

1. Allocate the first N pages into the frames and increment the page faults accordingly.

2. When next frame in the reference string is not already available in the allocated list do

a. Look for a frame in the reference string will not be used for longest period of time. b. Increment the page fault whenever a frame is replaced. (Hint: Locate the position of each allocated frame in the reference string; identify a frame for replacement with largest index position)

3. Display the allocated frame list after every replacement.

**LRU REPLACEMENT**

1. Allocate the first N pages into the frames and increment the page faults accordingly.

2. When next frame in the reference string is not already available in the allocated list do

a. Look for a frame which is not used recently.

b. Increment the page fault whenever a frame is replaced.

3. Display the allocated frame list after every replacement

**LFU REPLACEMENT**

1. Allocate the first N pages into the frames and increment the page faults accordingly.

2. When next frame in the reference string is not already available in the allocated list do

a. Look for a frame which is least frequently used.

b. Increment the page fault whenever a frame is replaced.

3. Display the allocated frame list after every replacement

**Sample input & output:**

**PAGE REPLACEMENT ALGORITHMS**

1. READ\_INPUT

2. FIFO

3. OPTIMAL

4. LRU

5. LFU

6. EXIT

Enter your option: 1

Enter the number of free frames: 10

Enter the number of frames required by the process: 4

Enter the reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Enter your option: 2

**FIFO Page Replacement Algorithm**

The reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

**Page ref 🡪 memory 🡪 PF**

7 🡪 7 - - - 🡪1

0 🡪 7 0 - - 🡪 2

1 🡪 7 0 1 - 🡪 3

2 🡪 7 0 1 2 🡪 4

0 🡪 7 0 1 2 🡪 -

3 🡪 3 0 1 2 🡪 5

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

Total Number of Page Faults: 10