Ain shams university

Faculty of engineering

Computer engineering and software systems programs

Course: Operating Systems (CSE223)

**Project 2**

Submitted by: Shehab Ahmed Hassan Kotb Id:16p6014

Mostafa Ahmed Mohamed Id:17p6040

Hazem hamada Abdellaif Id:16p3100

Submitted to: Dr. Gamal Ibrahim Abd el shafy

Eng. Kareem Darweesh

Table of Contents

[1.0 Algorithms Implementation 2](#_Toc533348733)

[1.1 LRU 2](#_Toc533348734)

[1.2 LFU 2](#_Toc533348735)

[1.3 Second chance 2](#_Toc533348736)

[1.4 Enhanced Second chance 2](#_Toc533348737)

[1.5 Optimal Algorithm 2](#_Toc533348738)

[2.0 input 3](#_Toc533348739)

[3.0 Test cases 3](#_Toc533348740)

# 1.0 Algorithms Implementation

As FIFO is a straight forward algorithm implemented with simple queues we will skip it’s discussion, and discus the other 6 algorithms.

## 1.1 LRU

we use a stack approach to implement LRU, whenever a page is accessed, pull that page from the middle of the stack and place it on the top. The LRU page will always be at the bottom of the stack. Because this requires removing objects from the middle of the stack, we use the LinkedList provided in java.

## 1.2 LFU

## 1.3 Second chance

## 1.4 Enhanced Second chance

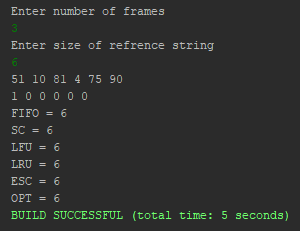
For this algorithm we use the clock algorithm approach, the frames are in an array where a pointer circles this array, at the begging the pointer points to the first element in the array and when a page need to be replaced the code execute the following loops:

* Cycle through the array looking for <0,0>. If one is found, use that page.
* Cycle through the array looking for <0,1>. Set the use bit to zero for all frames bypassed.
* If step 2 failed, all use bits will now be zero and repetition of steps 1 and 2 are guaranteed to find a frame for replacement.

## 1.5 Optimal Algorithm

For the this Algorithm we have a predict function that gets called when a page need to be replaced, the algorithm loops through the frames and finds the page which isn’t used the longest time in the future, if it finds a page which is not used at all in the future it returns it instead, also if all pages in the frames aren’t used in the future it return the first one, the algorithm then proceeds to remove that page.

# 2.0 input



The input is straight forward, the user inputs number of frames followed by size of reference string to generated, the program then outputs the generated string along with the modified bits generated. **Notice that because the reference string numbers are generated from 0-99 the number generated are all unique which makes all algorithms output the same page faults**

# 3.0 Test cases