**CASE STUDY**

**Page Replacement Algorithm**

**Simulator**

**Prepared by:**

**Espinosa, Lazhar Angelo**

**Submitted To:**

**Jo Anne Cura**

**Instructor**

## 

Contents

[**I. Introduction** 3](#_Toc198678741)

[**1.1 Page Replacement Algorithms** 3](#_Toc198678742)

[**1.1.1 First-In-First-Out (FIFO) Algorithm** 3](#_Toc198678743)

[**1.1.2 Least Recently Used (LRU) Algorithm** 4](#_Toc198678744)

[**1.1.3 Optimal Page Replacement (OPT)** 4](#_Toc198678745)

[**1.2 Analysis** 4](#_Toc198678746)

[**II. Documentation** 4](#_Toc198678747)

[**2.1 User Interface** 4](#_Toc198678748)

[**2.1.1 GUI Interface** 5](#_Toc198678749)

[**2.2 Sample Inputs and Outputs** 5](#_Toc198678750)

[**2.3 Source Code** 9](#_Toc198678751)

[**2.3.1 FIFO Implementation** 9](#_Toc198678752)

[**2.3.2 LRU Implementation** 10](#_Toc198678753)

[**2.3.3 OPT Implementation** 11](#_Toc198678754)

## **I. Introduction**

Page replacement algorithms are critical in operating systems to manage memory efficiently when the number of page frames is limited. This project implements and evaluates three widely studied algorithms:

1. **FIFO**: Replaces the oldest page in memory.
2. **LRU**: Replaces the least recently used page.
3. **OPT**: Replaces the page not used for the longest future time (theoretical optimal).

A GUI is integrated to facilitate user interaction, enabling dynamic testing of these algorithms under configurable parameters. The simulation highlights the trade-offs between implementation complexity and fault reduction.

### **1.1 Page Replacement Algorithms**

Page replacement algorithms manage memory efficiently when physical memory is full. They determine which existing page to replace when a new page is loaded.

### **1.1.1 First-In-First-Out (FIFO) Algorithm**

FIFO replaces the oldest page in memory. The operating system maintains a queue of pages, and the page at the front (oldest) is removed when a page fault occurs.

### **1.1.2 Least Recently Used (LRU) Algorithm**

LRU replaces the page that has not been used for the longest time. It assumes that recently used pages are more likely to be accessed again.

### **1.1.3 Optimal Page Replacement (OPT)**

OPT replaces the page that will not be used for the longest time in the future. It is theoretically optimal but impractical due to its reliance on future knowledge.

### **1.2 Analysis**

* **FIFO** is simple but suffers from Belady’s anomaly.
* **LRU** balances efficiency and practicality but requires tracking usage.
* **OPT** is theoretical and unimplementable in real systems.

## **II. Documentation**

### **2.1 User Interface**

The GUI simulator has input and results:

### **2.1.1 GUI Interface**

A screenshot of a computer

AI-generated content may be incorrect.

**Input Fields**:

* + **Number of Frames**: Accepts integer values (e.g., 3).
  + **Reference String**: Accepts space-separated integers (e.g., 7 0 1 2 0 3).
  + **Generate Reference String**: Auto-generates a random string based on the specified length.
* **Run Algorithms**: Executes FIFO, LRU, and OPT simulations.
* **Results Panel**: Displays step-by-step memory states and total page fault

### **2.2 Sample Inputs and Outputs**

**Test Case 1**

* **Input**:
  + Frames: 3
  + Reference Length: 20
  + Reference String: 1 4 4 3 3 1 9 6 9 7 5 0 0 8 4 0 5 6 7 4
* **Output**:

A screenshot of a computer

AI-generated content may be incorrect.

**Test Case 2**

* **Input**:
  + Frames: 4
  + Reference Length: 10
  + Reference String: 9 9 0 9 5 8 4 6 7 3
* **Output**:

A screenshot of a computer

AI-generated content may be incorrect.

**Test Case 3**

* **Input**:
  + Frames: 2
  + Reference String: 4 9 2 0 5 7 3 8 1 6 4 2 7 5 0
* **Output**:

A screenshot of a computer

AI-generated content may be incorrect.

### **2.3 Source Code**

### **2.3.1 FIFO Implementation**

A computer screen shot of a program code

AI-generated content may be incorrect.

### **2.3.2 LRU Implementation**

A computer screen shot of a program code

AI-generated content may be incorrect.

### **2.3.3 OPT Implementation**

A computer screen shot of a program code

AI-generated content may be incorrect.