

Page replacement algorithms

1 Introduction

The purpose of this lab is to get you familiarized with page replacement algorithms within the memory management unit (MMU) within an operating system (OS).

The virtual memory of a process is divided into pages. These pages are brought into physical memory, called frames, once they are accessed by the process. However, the number of frames is usually much smaller than number of pages so one needs some strategy to decide which page to evict once a new page must be brought into memory.

You will be given a solution for the LRU (Least Recently Used) frame eviction strategy. Familiarize with the program and try out different frame sizes and page reference strings.

2 Assignment

In this lab you will implement 3-4 different page replacement algorithms:

1. **FIFO.** First in first out eviction strategy is as simple as it sounds: Page first brought into memory is the one that is evicted.
2. **LFU.** Least frequently used. This algorithm uses a reference counter and evicts the page that has the smallest value at eviction stage. That is, it evicts the page that has been least frequently used. Two pages having the same value, you can choose page at random.
3. **OPT or Algorithm of your own choice.** The theoretically optimal page replacement algorithm is an algorithm that works as follows: when a page needs to be swapped in, the operating system swaps out the page whose next use will occur farthest in the future. This algorithm cannot be implemented in a general purpose operating system because it is impossible to compute reliably how long it will be before a page is going to be used, except when all software that will run on a system is known beforehand.

Algorithm of your own choice is either some more intricate solution OR if you really want to challenge yourself you come up with an algorithm based on a completely new idea.

If you are implementing an Algorithm of your own choice, discuss this with your lab assistant.