Project
CSE 421/521 – Operating Systems

1. Preparation

Before beginning your work, please read the following carefully:

- Chapter 9 from Silberschatz
- Lecture slides on Virtual Memory and Page Replacement Algorithms

2. Programming Task: Implement a Simple Virtual Memory Manager

The objective of this project is to implement a simple virtual memory manager "**virtualmem**" in C/C++ on a UNIX-based platform.

SYNOPSIS: virtualmem [**–h**] [-**f** *available-frames*] [**–r** *replacement-policy*] [**–i** *input_file*]

DESCRIPTION: virtualmem is a simple virtual memory manager. It takes a sequence of page references as an input, as well as the number of available frames. It performs the placement of these pages to the available frames using the page replacement policy specified by the user.

2.1 OPTIONS:

–h :	**:** Print a usage summary with all options and exit.
–f *available-frames* **:**	**:** Set the number of available frames. By default it should be 5.
–r *replacement-policy*	**:** Set the page replacement policy. It can be either **FIFO** (First-in- firstout), **SC** (second chance/clock), or **LFU** (Least-frequently-used). The default will be **FIFO**.
–i *input file* **:**	**:** Read the page reference sequence from a specified file. If not given, the sequence should be read from STDIN (ended with ENTER).

2.2 OUTPUT:

You should also implement the **Optimal** page replacement algorithm, and compare the performance of the replacement-policy chosen by the user (above) to the Optimal algorithm, in terms of number of page replacements for the given string.

i.e:

$ virtualmem –f 10 –r LFU –i myinpufile

# of page replacements with LFU	: 118
# of page replacements with Optimal	: 85
% penalty using LFU	: 38.8%