CSE 3150 — C++ Essentials — Yufeng Wu — Fall 2024
Programming Assignment 7. Due: 11:59 pm @ 10/25/2024.

Extra time for this PA: Due to exam 1, you will have one more week to work on this PA.

In this assignment, you are to implement the page replacement algorithms for managing virtual memory in a modern operating system. In a modern computer, the virtual memory is organized into pages (or block of memory). We use an integer to represent a page. That is, each page has a distinct integer as its ID. The main memory has a limited capacity and can only hold up to K pages. Here, K is the capacity of the main memory. An important problem regarding to virtual memory is the page replacement: when the memory reaches its limit (i.e., having K pages) and a new page (not currently in memory) is to be add, then need to swap out one current page to make room for this new page. Virtual memory is an important subject in computer science. If you haven't learned about it, it is time to get to know it (you can ask ChatGPT about virtual memory...).

There are many different page replacement algorithms. In this programming assignment, you need to implement a class that supports following methods. These methods will be used to implement the Least Recently Used (LRU) algorithm which will be tested in the provided test code. Briefly, LRU policy is to evict the page that is least recently used to accommodate a new page to be added to the main memory.

This class would support mainly the LRU algorithm as follows:

1. Constructor: take the number of pages (capacity of main memory). This would initialize the memory.
2. *AccessPage*: this function accesses a specific page; you should run LRU upon this request to access page.
3. *RunOpt*: run the optimal page replacement on all the pages that have been accessed so far. Return the number of page faults by this OPT algorithm.
4. *GetNumPageFaults*: return the number of page faults so far (by LRU).

The following lists things to note when you implement this class.

1. You should define the data structure of the virtual memory as data member in the class.
2. Use STL set/vector/map when applicable. These STL utilities have useful methods/functions that can make the implementation easier.
3. Read the code given in the test code to understand how the test code works. This can help you understand more deeply what the functions are meant to do. In particular, pay attention to the following two main test functions:
   (a) RunLRU: it takes a list of pages to request (the pointer to the array of pages and the size of array), the size (number of pages) of the memory (memoryCapacity, and the the number of page faults it will be when you run the LRU algorithm. If the number of faults from your implementation doesn't match, it will throw an exception.
   (b) RunOpt: the same inputs. The only difference is, it runs the optimal algorithm and examines whether the numbers of page faults match.