

# Virtual Memory Lab

This lab project addresses the implementation of page-replacement algorithms in a demand-paging system.

Each process in a demand-paging system has a page table that contains a list of entries. For each logical page of the process, there is an entry in the table that indicates if the page is in memory. If the page is in memory, the memory frame number that page is resident in is indicated. Also, for each page, the time at which the page has arrived in memory, the time at which it has been last referenced, and the number of times the page has been referenced since the page arrived in memory are maintained.

The page table data structure is a simple array of page-table entries (PTEs). Each PTE contains five fields as defined below:

```
struct PTE {  
    int is_valid;  
    int frame_number;  
    int arrival_timestamp;  
    int last_access_timestamp;  
    int reference_count;  
}
```

Each process in the system, has a page table that is simply an array of PTEs. Each process also has a pool of frames that is allocated. The frame pool for a process is represented as an array of integers, where each Integer represents the frame number of a frame that is free and is available for use by the process. Note that in order to get the frame number in the pool, you first need to access the integer in the array.

This lab project aims to exercise the various policies for page replacement. In particular, we study the following three page-replacement policies:

1. First-In-First-Out (FIFO)
2. Least-Recently-Used (LRU)
3. Least-Frequently-Used (LFU)

In order to implement the above policies, we need to develop corresponding functions that process page accesses for a process. That is, for each process, given its page table, a logical page number being referenced and the free frame pool of the process, the functions should determine the memory frame number for the logical page. Also, the functions should modify the page table and the free frame pool appropriately. The details of the functions with respect to the different policies are described below. You need to develop code for these functions































































































