

CSCE A320: Operating Systems
Fall 2017. Assignment 6

MARK QUIMPO
(31113576)

VIRTUAL MEMORY SIMULATOR
MMU AND TLB

Source files:

```
makefile
vm.h
vm.c
pagetable.h
pagetable.c
inputgen.c
replacement.c
disk.h
disk.c
```

In this last assignment, I implement a virtual memory simulator in order to understand the behavior of a page table and the page replacement algorithms. Virtual memory allows the execution of processes that are not completely in memory. This is achieved by page table and page replacement techniques. A page table stores the mapping between virtual addresses and physical addresses. A page fault mechanism by which the MMU can ask the OS to bring in a page from the disk.

Page Table

The structure of this page table will simulate a linear page table. Each page table entry should contain a valid bit, a physical page number, and a dirty bit. The size of main memory on the machine you are simulating is 256 pages.

Initially, physical memory is empty and a free-frame list contains every physical page. The page allocation policy will simply hand out the physical pages in order of increasing page number, until the free-frame list is empty. From that point on, all physical pages will be obtained by page replacements algorithm. This simulation, pages will never be added back to the free-frame list but processes never terminate.

No actual placement of virtual pages on the disk. Instead, simply keep track of the number of disk read and write operations that are needed to handle the page faults.

- Data structure for the page table entry and statistics (hit and miss count) are defined in `pagetable.h`
- You will need implement `"hittest()"` and `"pagefaulthandler()"` functions in `pagetable.c` file.
- You will need to call `"diskread"` to read a page from the disk into the main memory, and `"diskwrite"` to write out a dirty page to the disk.
- If the physical memory is full, you will need to call `"pagereplacement()"` to find a victim page.

This simulator only supports random page replacement algorithm that randomly choose a victim page. The simulator uses the command-line argument to specify a random replacement policy.

```
$ ./vm 0
```

Run :

```
$ make  
$ ./vm 0 < input.txt
```