Programming Assignment #4—Virtual Memory

COMP 222 Computer Organization

**Objective:**

To translate a virtual address to a physical address, involving a custom-sized fully associative page table.

**Menu Options:**

The program translates a virtual address to a physical address based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:

***Your Full Name*** (must always appear above the menu)

1) Set parameters

2) Map virtual address

3) Quit program

Upon entering the parameters, the page table is to be dynamically allocated based on the total number of page frames. The page frames will be mapped to virtual pages on demand in the page frame order 0,1,2,3,…

**Inputs:**

* The total size of physical memory (in words)
* The page size (words/page)
* The replacement policy (LRU, FIFO)

**Outputs:**

The corresponding physical address for a virtual address

A message indicating a page fault (if any) in the page table

**Programming Notes:**

* To use the **Math library**, use: “**#include <math.h>”** to access various functions, such as pow(base, exp), log(number), etc.
* If necessary, also include the flag “**-lm**” when you compile. If you develop your program using MinGW, you do not need to include this option in your command line, BUT eventually you will need it for k200.ecs.csun.edu.

A sample command line to compile your program and utilize the math library is **gcc –lm prgassign3\_yourlastname.c –o prgassign3\_lazik**

* To use the **String library**, use: **“#include <string.h>”** to access various functions, such as strlen(string) which returns an integer representing the length of a string of characters.

**What to turn in:**

* **Softcopy** of **source code** submitted to **http://moodle.csun.edu** via the submission instructions. Be sure to name your source code: ***prgassign4\_yourlastname.c***

Please make sure **all characters in the program’s name are lower case**, including the first letter of your last name.

* **Input and Output formats must exactly adhere to those shown below**.

Any deviation from the sample input/output formats for submission will result in an automatic **20%** reduction in your grade.

* You can use any editor and/or compiler of your choice to do the assignment, but, before you turn in your assignment, **make sure** your code compiles and executes under the **gcc** compiler and the Unix environment on CSUN’s **k200.ecs.csun.edu** server; otherwise you will receive 0 points for compilation and execution.

**For the first two assignments, I permitted students to go back and fix their first two assignments when it was determined that they didn’t compile/execute error free on k200.ecs.csun.edu. The grade, and the grades of all late assignments was reduced by 25% which is better than zero. But I will not be doing this for this assignment and the remaining assignments in the course.**

**IF YOUR SUBMISSION DOES NOT COMPILE/EXECUTE ERROR FREE ON K200.ECS.CSUN.EDU THE FIRST TIME I TRY IT, YOU WILL RECEIVE A GRADE OF ZERO!**

**The very last thing you should do before submitting your program is compile and execute it on k200.ecs.csun.edu using the sample command line above.**

* Hardcopy printed listing of your program. Please place this on the Professor’s desk at the beginning of class on day the assignment is due. Make sure your full name appears on each page of the listing and that all pages are stapled together in their correct order **BEFORE** you come to class.

**Testing Your Program**

Shortly, I will provide a text file that you can use to test your program. I will also provide a copy of the expected output.