CS220 - Computer System II
Project 3

Due: 11/22/2016, 11:59pm

1 Instructions

- You are to work on this project alone. You are allowed to discuss ideas with your friends (e.g., "hey, what do you think the rtl array is? I think it is the reverse lookup table that maps a physical page to the owning process." is acceptable. However, "hey, can I take a look at your code?" IS NOT!).
- You are required to demonstrate your outputs to the TA. Await demo instructions from the TAs. Failure to demo will result in a 0!

2 Introduction

In this project, you will be implementing a virtual memory system. You will be given a simulator that manages memory models, but has some critical parts missing. This project will help you understand C programs that span multiple source files. We will guide you through the implementation of each of the missing parts step-by-step. It is in your best interest to understand underlying concepts before you type up code! The implementation of the virtual memory system is straightforward. If you are struggling to write code, then you should step back and contemplate on the concepts. If you do not understand the concept behind each step, then you will have a difficult time implementing them in code.

3 Project Folder Structure

The cs220_proj3 folder contains 3 sub-folders, the Makefile and this pdf document. The simulator source code is in sim_src folder, you will not be writing any code in this folder, however, you will be examining the code in sim_src to understand the overall control and data flow. Your changes are required in impl_src folder. The source files have been annotated with "TASK"s. You are to complete the tasks. You are not required to add any new files. The input folder contains sample inputs to test your code.

4 Point Distribution

- **TASK 1 (10 points):** Complete the macros (in macros.h).

- **TASK 2 (25 points):** Implement TLB lookup (in tlb-lookup.c).
- **TASK 3 (15 points):** Implement page table lookup (in page-lookup.c).
- **TASK 4 (15 points):** Implement physical page allocation (in page-replacement.c).
- **TASK 5 (25 points):** Implement the page fault handler (in page-fault.c).
- **TASK 6 (10 points):** Implement code to calculate access time (in eval.c).

5 Compiling and Running

To recompile the simulator, we have provided you with a Makefile in the top-level directory. You can type in the command make in the terminal to build the project. When the project is finished building, there will be an executable in the top level directory called vsim. To execute this file, you must provide it with an file. Input files are found in inputs folder, and it provides four sample inputs:

- **sample1** – A basic input file that yields no page faults *due to eviction*.
- **sample2** – A reference file that provides a bit of everything (tlb miss, page fault, etc.).
- **sample3** – A reference file that should cause a large number of TLB hits.
- **sample4** – A reference file that should produce a large number of page faults.

If you wanted to run the simulator with the sample1 file, you would do the following:

```
$ ./vsim inputs/sample1
```

There are several other command line options for the simulator that you can play around with, such as changing the memory size, page size, and the tlb size. You can change these parameters to see how they affect the memory access time. The default settings are memory size = 16 values, page size = 2 values, and tlb size = 4.

6 Submitting the project

When you are done with the project, move to the top level directory of your project and run the following:

```
$ make submit
```

This will generate a file called p3-submit.tar.gz that contains everything you need to submit for this. Remember to submit this file on Blakcboard. Additionally, you are **required** to demo the project to the TA. When the TAs open slots on their calendars, you are to pick slots to demonstrate your project.