**Operating System Part 3**

**Austin Blythe, Matthew Francis**

**Account: cs441104**

**20 March 2013**

**Program problem/statement:**  
We are changing the memory management to have the use of Page Tables and Frames through the use of MMU. As the user has the program loaded from the disk into main memory, it chooses a location randomly, verifies it is available, and places the pages and updates the frame to locked.

**O/S** **Architecture:**

* 16-bit words
* 256-word memory (16 bits) word addressable
* 3 general purpose registers (1-3), 1 special accumulator (A; 0)
* 8-bit program counter (PC)
* condition code flags: = or zero(010), > or positive(001), < or negative (100)
* machine instruction cycle - fetch, decode, opfetch, execute, writeback

**Difficulties encountered:**

* The most difficult problem we encountered with this iteration was that we could not conceptually understand it for a long period of time.
* We had a few different ways to implement the Page Table, but after trial and error decided on our final version.
* Compiling across different platforms (UNIX, Mac OSX, and Windows) occasionally caused different functionality. We attempted to normalize our compiling be using the same compiler consistently. (Constant problem from previous version).

**Procedure:**  
 We spent a significant amount of time trying to understand how to go about implementing a page table, and attempted to get some work done before spring break. However, upon returning we realized we had made almost no progress, and spent the 17th – 19th programming it.  
**Additional Observations:**  
 Spring break slowed us down significantly for this iteration. We finished most of the programming from the 17th to the 19th. We had a bit of crunch on the final days when we were having an infinite loop bug.

**Assumptions:**

We are assuming that a program is loaded when the user hits run, as opposed to the OS saying run (both were specified in the assignment sheet). The page table, when moving over to main memory, does not choose the frames to be concurrent, so instead they can be randomly chosen and spread out.  
  
**Detailed table of work:**  
The time we spent on it before spring break was only to understand the concept, and was not accounted for in this table (and it was a significant amount of time).

|  |  |  |
| --- | --- | --- |
|  | **Austin** | **Matt** |
| **3/17/2013** | 2 hrs | 2 hrs |
| **3/18/2013** | 3 | 3 |
| **3/19/2013** | 5 | 4 |
| **3/20/2013** | 1 | 2 |