**Operating System Part 5**

**Austin Blythe, Matthew Francis**

**Account: cs441104**

**24 April 2013**

**Program problem/statement:**  
For this iteration of the Operating System, we have implemented a File Allocation Table (FAT). Rather than having the users information locked in the code, it can be loaded from a file at runtime.

**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:**

* We had to undo some of the work of the previous iteration of the program, such as repairing the functionality of the dmp command, and making sure the queue’s are working in the manner specified.
* Reading in from the file caused a bit of headache when trying to parse the information, once we had the proper format for reading it in, there was little difficulty after that.

**Procedure:**  
 We first fixed the issues with the queue’s and commands. From there, we began working on reading in from the file, and having it loaded into the proper places from there.

**Additional Observations:**  
 Other than input issues, this was not a difficult iteration, just tedious.

**Assumptions:**

We are assuming that the file we are reading from is of our own creation, and therefore format as we were not able to find a file in the instr directory in unix.   
  
**Detailed table of work:**

|  |  |  |
| --- | --- | --- |
|  | **Austin** | **Matt** |
| **4/20/2013** | 4 hrs | 4 hrs |
| **4/21/2013** | 6 hrs | 6 hrs |
| **4/22/2013** | 4 hrs | 4 hrs |
| **4/23/2013** | 3 hrs | 3 hrs |