**Operating System Part 1**

**CSC 441**

**Austin Blythe, Matthew Francis**

**6 February 2013**

**Account: cs441102**

**Program problem/statement:**  
In order to develop and explore operating system components, an architecture and platform is necessary. Implement a machine language interpreter (hardware simulator) for the architecture below using the following specification outline. This will provide the basis for the “hardware system”. The O/S design will begin addressing the O/S components. This is O/S version 1. Work in teams of two or three is indicated in class.  
 **Design:**

* 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:**

* Original conception had many of the functions not storing in the proper places and things of that nature, after it was explained, we corrected the functions to store in the proper memory address or registers, depending on the command.
* Had to fix many of the NOT, AND, and IOR commands when we realized what their function was. This was easily solved by C’s bitwise functions.
* After putting the program together and having it run with both the code on the sheet and a test set, we created the command loop. For some reason when this happened, it caused an error with a pointer being passed and caused an infinite loop. It was fixed by making certain variables global, rather than passing them to methods.

**Procedure:**  
 We began the project with the code from a similar project from a previous semester. We used that code as a foundation and modified it to fit the specifications of the operating system project. We initially focused on getting the machine language interpreter functioning properly, and then we moved on to the O/S functionality, such as the interactive user input commands.  
  
**Additional Observations:**  
 We started working early enough to not have to worry about problems too much, and allowed us to have it compile and run correctly. Correcting some of the errors that were present in the former 214/475 program. We’ve done most of work at a consistent rate since the project was assigned in the McGee computer lab.  
  
**Detailed table of work:**  
We met in the computer lab almost daily to work on it together and to brainstorm how certain things function / code.

1/22/2013 - Austin & Matt - 2 Hours

1/23/2013 - Austin & Matt - 2 Hours

1/24/2013 - Austin - 2 Hours

        Matt     - 1 Hour

1/28/2013 - Austin & Matt - 2 Hours

1/29/2013 - Austin & Matt - 2 Hours

1/30/2013 - Austin & Matt - 2 Hours

1/31/2013 - Austin 2 - Hours

        Matt  -  1 - Hour

Each having small amounts of work spread throughout almost daily.