# EE306 Introduction to Computing

**Lab 7 (due on 4/30, 9pm, on GitHub)**
**Course Instructor:** Dr. Nina Telang

**All Lab assignments must be completed individually. You are not permitted to seek help or clarification from anyone other than the instructor or the TAs.**

**Your file should be named exactly after your EID, for example, xy1234.asm. Your program will not be graded if you fail to follow the naming convention of the file name.**

## Problem statement

The purpose of this assignment is to show how interrupt-driven I/O can interrupt a program, execute the interrupt service routine, and return to the interrupted program, picking up exactly where it left off (just as if nothing had happened). In this assignment, we will use the Keyboard as the input device for interrupting the running program.

To demonstrate this, we will need three things:

1. The interrupt vector table must be generated; interrupts will need to be enabled. The operating system will set up the interrupt vector table and enable the interrupts and then start execution of the user program.

2. A program that will run at priority level 0.

3. An interrupt service routine that will be invoked as a result of someone hitting a key on the keyboard.

## The operating system enabling code.

We are going to require you to do the enabling actions that the operating system would do before the execution of the User program.

1. Normally, the operating system establishes the interrupt vector table to contain the starting addresses of the corresponding interrupt services routines. You will have to do that for the keyboard interrupt. The starting address of the interrupt vector table is x0100 and the interrupt vector for the keyboard is x80.

2. Normally, the operating system would set KBSR[14] to enable the keyboard to interrupt user programs. You will have to do that.

3. Push Appropriate PSR and PC values to the stack. Then, an RTI instruction in the OS code will take you to the user program starting at 0x3000.

You must perform these tasks as a part of the operating system code.

# EE306 Introduction to Computing

**The user program**.

The user program will print the prompt "WHAT STARTS HERE CHANGES THE WORLD" followed by a newline character (x0A). The user program is to do this in an infinite loop.

In order to make sure the prompts don't get written to the monitor too fast to be seen by the naked eye, the user program should include a piece of code that will count down from say x5000 each time a character is output to the screen. **You may want to try out other numbers if your processor is too fast or slow.**

**The keyboard interrupt service routine**.

The keyboard interrupt service routine will read the character typed, prints a newline and performs one of the following tasks based on the type of character:

1. Upper Case Alphabet, then print the string "Practice Social Distancing" and then a newline.

2. Lower Case Alphabet, then print "Wash your Hands frequently" and then a newline.

3. Number between 0 and 9, then print "Stay Home, Stay Safe" and then a newline

4. For any other character, print "----------- End of EE306 Labs ^^ -----------", a newline and Halt the processor.

**VERY IMPORTANT**: **You are not allowed to use any TRAP instruction other than HALT in your interrupt service routine.** To display a character to the screen, you must poll the DSR and then write to the DDR, you may not call TRAP x21 (OUT), or use any of the other TRAP routines. If you use TRAP in the interrupt service routine or if you do not properly poll the DSR before writing to the DDR, your program is not correct and will not be graded *even though it may appear to work.*

**Hint**: Don't forget to save and restore any registers that you use in the interrupt service routine.
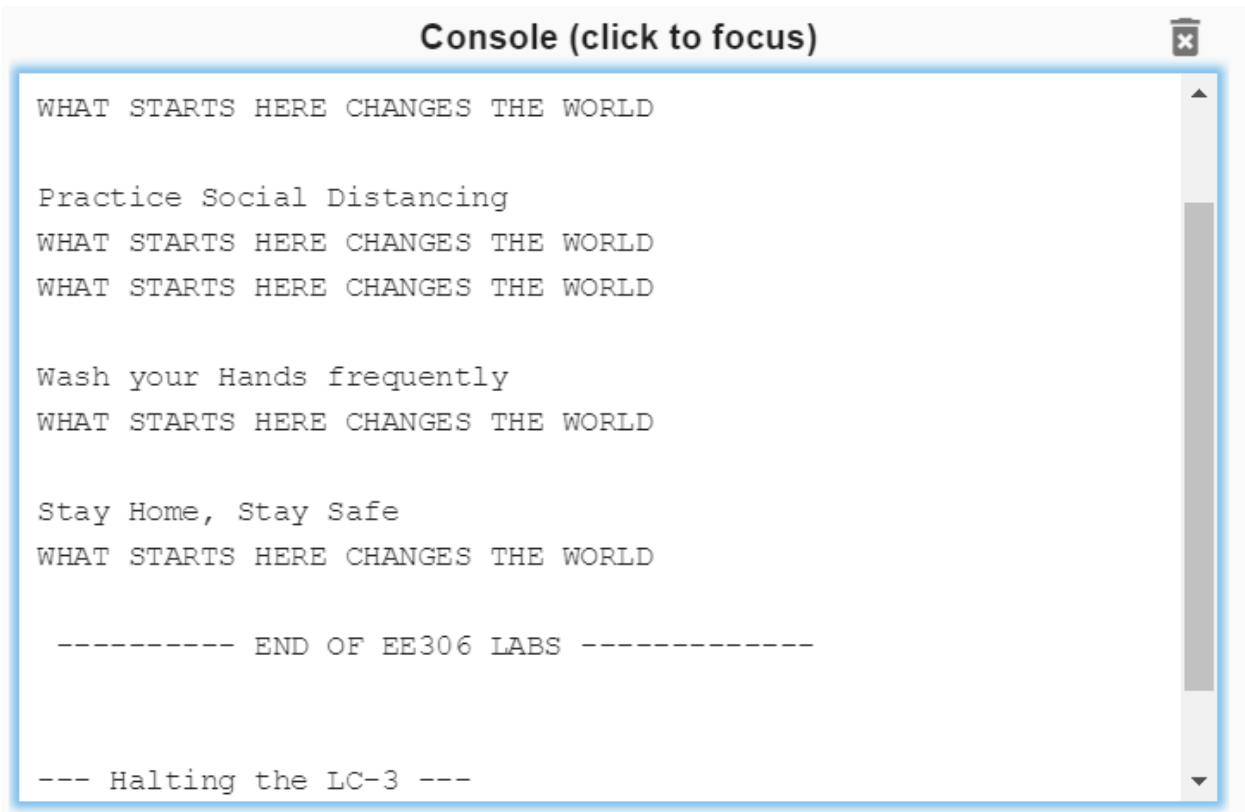
Your job will be to write the operating system code, user program and the keyboard interrupt service routine describe above.

**A template for each of these files is provided to you. Do not change the .ORIG address in these files and use them to finish the requirements of this lab.**

Turn-in a single file named exactly after your eid. For example, xy123.asm.

# EE306 Introduction to Computing

An example of how the console should look when you run the program:



Figure 1: Sample Output - Key Sequence: A, z, 9, ]