

EE 306, Fall 2016

Programming Lab 5

Due: Saturday, December 3, 11:59 PM

You must do every programming assignment by yourself. You are permitted to get help ONLY from the TAs and the instructor. **Absolutely no late assignments will be accepted.**

Setup (note the starter files are different than previous labs)

The starter files for Lab 5 can be found [here](#). Once you download the lab zip file, you can extract it the same way you did for [Lab 0](#). This time the lab directory contains two files called `lab5.asm` and `isr.asm`. You will write all the code for the programming lab in these files. When you are loading files into the simulator, you should first load isr.obj and then lab5.obj. You can find the simulator [here](#).

Overview

In this lab you will write a keyboard interrupt service routine (ISR) that counts the number of keys that have been pressed. Meanwhile, a user program will output the character 'x' constantly.

Assignment

Unlike previous labs, for this lab you will be writing your program in two separate parts: the user program and the interrupt service routine.

User Program

The user program, starting at location x3000, will set up the keyboard ISR then loop forever while printing the 'x' character to the screen (no newlines). The steps to set up an interrupt service routine are as follows:

1. Initialize stack pointer (R6) to x3000.
2. Save the starting position of the ISR in the interrupt vector table (IVT). The entry for the keyboard interrupt is at location x180.
3. Enable keyboard interrupts by setting bit 14 of the KBSR to a 1. You may do so by storing the value x4000 to the KBSR (i.e. no need to read the KBSR, set a single bit, and then write it back).

Once the interrupt has been set up, the user program should repeatedly display the character 'x'. The starter code provides a DELAY subroutine that you must call between each 'x' you output in the user program. The purpose is to slow down the user program so that when the interrupt is triggered, you can see its output (see section below). You may put a HALT at the end of your user program, but the PC should never reach it because of the loop.

Interrupt Service Routine

The ISR, starting at location x1500, will output the character that was inputted and then output the number of times the keyboard has been pressed. Before printing out the character, you

should print a newline. After printing the count, you should also print a newline. **The count should be a 4-digit hexadecimal number.** If the 'a' key is pressed and it is the 2nd key that has been pressed, the output should be: 'a (x0002)'. If the 'b' key is pressed and it is the 31st key that has been pressed, the output should be: 'b (x001F)'. Note that the letters in hexadecimal are capitalized.

Sample run

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
a (x0001)
xxxxxxxxxxxxxxxxxxxxx
z (x0002)
xxxxxxxxxxxxxxxxxxxxx
^ (x0003)
xx
1 (x0004)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Your output may vary depending on when you press the keys and what keys you press. In particular, you may have a different number of 'x's before the ISR is called. Also, you should never see the processor halt.

Hints and Suggestions

- Outputting a binary number in ASCII hex is probably the hardest part of this assignment. You may find it useful to use your lab 2 (left rotate) as a subroutine during the conversion.
- You should use subroutines where appropriate.
- In each subroutine and ISR you write, you should save and restore any registers that you use. This will avoid a major headache during debugging.
- The number of times the keyboard is pressed will be less than 2^{16} .

Submitting

Please follow the same submission instructions as [Lab 0](#) *carefully* (the submission part starts at step 4). Make sure you are using lab5 every place that lab0 was used. One of the common mistakes we saw with lab 0 was renaming the lab0 folder to eid_lab0. You should only rename the zip file **after** it is created, and not the directory inside. Make sure you delete all files that the assembler generates (lab5.bin, lab5.obj, etc.) before submitting. Also, make sure you do not just compress the lab5.asm and isr.asm files, and instead compress the entire lab3 directory.