CSE 2312 – Homework #4

Assigned: Friday, September 29, 2017
Due: Friday, October 6, 2017 at the end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. If your submission has more than one page, staple the pages. If I have to staple it for you, the cost is 10 points.

Assignment:

1. Write an ARMv7 assembly program that creates an array of 4-byte integers (i.e., words in assembly terminology) and then finds the minimum and maximum elements of the array.

   (a) On the course website is a file, `hw04a-data.txt`, that contains the data you should hard-code into your program.

   (b) With regard to our data, we know the following and can hard-code it in the program:

   - the array elements
   - the number of array elements

   Nothing else in your logic should be hard-coded to your data.

   (c) The final value in R0 should be the sum of the minimum and maximum values, i.e., `R0 = min_value + max_value`.

   (d) The name of this program will be `hw04a.s`, all in lowercase.

2. Write an ARMv7 assembly program that stores a string and then moves through the characters of the string, reversing the case of the letters. When this is complete, the program will write the string to `stdout` (i.e., print the string).

   (a) On the course website is a file, `hw04b-data.txt`, that contains the data you should hard-code into your program.

   (b) The uppercase characters in the original string should be changed to lowercase and vice versa.

   (c) With regard to our data, we know the following and can hard-code it in the program:

   - the string will be null-terminated (i.e., will include zero at the end)
   - the ASCII values for various character classes

   Nothing else in your logic should be hard-coded to this particular string, including the number of characters in the string.

   (d) The modified string should be printed in its entirety after it has been processed completely. This should be done with a single syscall. Do not print each character individually.

(e) The name of this program will be hw04b.s, all in lowercase.

Note the following:

1. You must comment your code. This means having enough comments that I can easily follow your logic.

2. Your programs should conform to the assemble/link approach that we did in class and should have a _start section. That is, I should be able to do the following, assuming your program is called hw02.s:

   ```
   as  -o hw02.o  hw02.s
   ld  -o hw02  hw02.o
   ```

   The code should assemble and run on a Raspberry Pi with the Raspbian operating system.

3. Note, if you submit code that I think was produced by a compiler, then you will not receive credit.

4. As a comment in your programs, include your name.

5. To submit, create a directory with a name that matches your net ID (for example, abc1234) in lowercase.

6. Place your programs in the directory your created, then tar and compress the directory. If the directory is called abc1234, then you would type

   ```
   tar  cvzf  abc1234.tgz  abc1234
   ```

   These are Unix commands and should be performed on a Unix-like system, such as the Raspberry Pi.

7. Upload the compressed and tar'd file to Blackboard.