Computer Project #8

Assignment Overview

This assignment develops familiarity with control constructs in assembly language. You will develop an assembly language program for the ARM microprocessor to process ASCII characters.

It is worth 30 points (3% of course grade) and must be completed no later than 11:59 PM on Thursday, 11/3.

Assignment Deliverables

The deliverables for this assignment are the following files:

> **proj08.makefile** – the makefile which produces "proj08"
> **proj08.main.s** – the source code for your program

Be sure to use the specified file names and to submit them for grading via the CSE handin system.

Assignment Specifications

The program will classify each character in an input stream, and then produce a summary about that input stream.

1. The program will repeatedly read one character from the standard input stream (using the "getchar" function), will process that character, and will write that character to the standard output stream (using the "putchar" function).

2. After processing all characters in the standard input stream, the program will display the following counts (with appropriate labels):

   a) Total number of characters
   b) Number of newline characters
   c) Number of whitespace characters (blanks, tabs and newlines)
   d) Number of octal digits (in the set {0-7})
   e) Number of decimal digits (in the set {0-9})
   f) Number of hexadecimal digits (in the set {0-9, A-F, a-f})
   g) Number of letters (in the set {A-Z, a-z})

The program will display a separate line for each of the seven counts.

Assignment Notes

1. To perform input and output operations, you will use three functions which are part of the standard C library.

To read one character from the standard input stream, your program will call "getchar", which returns a 32-bit value in register R0 (either an ASCII character or the value -1, representing end-of-file).

To write one character to the standard output stream, your program will call "putchar", which accepts a 32-bit value in register R0 (representing an ASCII character).

To display the seven counts, your program will call "printf".

2. Your program will be assembled and linked using "gcc". For example, if your data file (containing ASCII characters) is named "proj08.data", the following commands could be used to assemble and link your program, then load and execute it:

```
<prompt> gcc proj08.main.s
<prompt> a.out < proj08.data
```

You will test your program using text files which you create, but your program is expected to execute correctly for any text file.

You may also input text directly from the keyboard by executing the program without input redirection:

```
<prompt> a.out
```

End-of-file is simulated by entering control-d at the beginning of a line.

3. In order to interface ARM assembly language functions with standard library functions (such as "getchar", "putchar" and "printf"), you must follow certain conventions about register usage.

The standard library functions assume that the calling function will place up to four parameters in registers R0 through R3 (with the first argument in register R0).

The standard library functions place their return value in register R0 before returning to the calling function.

Registers R12, R13, R14 and R15 are used by the system and their contents must not be modified by your program.