

Computer Project #8 -- SPARC Assembly Language

Assignment Overview:

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

It is worth 40 points (4% of course grade), and must be completed no later than 11:59 PM on Thursday, March 28.

Assignment Specifications:

- 1) The program will classify each character in an input stream, then produce a summary about that input stream.
- 2) 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).
- 3) After processing all characters in the standard input stream, the program will display the following counts (using the "printf" function):
  - a) Total number of characters
  - b) Number of control characters (0x00 to 0x1F, 0x7F)
  - c) Number of whitespace characters (0x09, 0x0A and 0x20)
  - d) Number of newline characters (0x0A)
  - 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})
  - h) Number of upper case letters (in the set {A-Z})
  - i) Number of lower case letters (in the set {a-z})

The program will display one line for each of the nine counts, including a descriptive label.

Assignment Deliverables:

The deliverables for this assignment are:

proj08.makefile -- the makefile which produces "proj08"  
proj08.student.s -- the source code file for your solution

Be sure to submit your files for grading via the "handin" program.

Assignment Notes:

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

To read a character from the standard input stream, your program will call "getchar", which returns a 32-bit value in register %r8 (either an ASCII character or the value -1, representing end-of-file).

To write a character to the standard output stream, your program will call "putchar", which accepts a 32-bit value in register %r8 (representing an ASCII character).

To display the nine counts (and the descriptive label for each), 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.student.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
```

An end-of-file is simulated by a control-d at the beginning of a line.

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

a) The standard library function assume that the calling function will place up to six parameters in registers %r8 through %r13 (with the first argument in register %r8).

b) The standard library functions place their return value in register %r8 before returning to the calling function.

c) Registers %r14, %r15, %r30 and %r31 are used to manage the run-time stack of activation records and their contents must not be modified by your function.

d) Registers %r0 through %r7 are shared between all functions in a program. Since they are global, a function should not assume that their contents are the same before and after a call to another function.

e) Registers %r16 through %r23 are local to a given function. Thus, that function can assume that their contents are the same before and after a call to another function.