

CST8134: In-lab Exercise 6

StrLen function

You are likely familiar with the `strlen()` function from the standard library, which operates on zero-terminated strings (C type – C++ uses `Strings`). By an amazing coincidence (or perhaps good planning), the Motorola 68000 uses zero-terminated strings natively, as in the `.ASCIZ` pseudo-op.

What you need (trust me on this) is a set of library function in 68000 assembly language, so your task today is to write the one called `StrLen`. It will return in register `d0` the length in bytes (as a 16-bit word) of the string passed to it in `a0`, not including the zero byte that marks its end. You can limit your maximum string length to a signed 16-bit value if you need to (`0x7FFF` or `32,767`).

Therefore, you will write the function (as a function, not as part of your main program) and call it from your test driver program like this:

```
TestLine1:    .ASCIZ    "some data here"
TestLine2:    .ASCII    "more data"
              .BYTE    CR, LF, 0
              ...
              lea      TestLine1, a0 | string to measure
              bsr      StrLen        | call my function
/* register d0 now contains the length of the string */
```

Repeat the call for `TestLine2` in your driver program, and display the lengths on the screen using `OutHex` (presumably with some meaningful text and suitable new lines).

You may use `movea.l` instead of `lea`, and `jsr` instead of `bsr` if you wish.

Show the source code and the correct output of your program to your lab instructor before the end of your lab period.