Assembly – Input and Output of ASCII strings

Pure assembly is only capable of text and binary I/O.

Conversion to numeric encodings is a higher-level action.

Simple-output-program review:

- Create ASCII string in **.data** segment
- In **.text** (instructions) segment:
  - load ASCII string's address into register x1
  - move string's length into x2
  - move destination number into x3
    - 1 → sysout
  - move "sys_write" function number into x8
    - 0x40, *a.k.a.* 64
  - issue supervisor call #0

  - move return code into register w0
  - move "sys_exit" function (0x5d) into x8
  - issue another supervisor call to end the program

Simple Input

- The converse of output
- Needs a *destination* for keyboard's input to go into:
  - create empty storage buffer for the input in **.bss** segment
  - create empty space for a length value
- In **.text** segment:
  - move buffer's address into register x1
  - move buffer's *maximum* length into x2
  - move input source number into x0
    - 0 → sysin
  - move "sys_read" function number (0x3f) into x8
  - issue supervisor call #0
  - svc *returns* the actual, entered length in x0
  - save the text's length in the value space

Minimal "echo" program

```
    .bss                // "Block Started by Symbol" - empty storage
    .balign 4           // align storage on 4-byte boundaries
    .lcomm entlen, 8            // space for input's length
.set MAXSIZE, 32
    .lcomm buffer, MAXSIZE      // space for the input

    .text
// Put these AHEAD of main(), so they're easier for us to find:
bufaddr:        .dword buffer
entaddr:        .dword entlen

    .global main
main:

// get input:
    mov x0, #0x00               // sysin
    ldr x1, bufaddr             // get buffer's address from "bufaddr"
    mov x2, #MAXSIZE
    mov x8, #0x3f               // sys_read
    svc #0
    ldr x1, entaddr             // get entlen's addres from "entaddr"
    str x0, [x1]                // save entered length

// echo it back:
    ldr x1, entaddr             // redundant in this program
    ldr x2, [x1]                // retrieve entered length
    ldr x1, bufaddr
    mov x0, #0x01               // sysout
    mov x8, #0x40               // sys_write
    svc #0

    ldr x1, entaddr             // x1 may have been changed...
    ldr x0, [x1]                // entered length becomes return value
    mov x8, #0x5d               // sys_exit
    svc #0
//-----------------------------------------------------------------------
```

Assemble, Link, Run: