

SYSTEMS I — LAB 6

Programming procedures in *k*-system assembly code

We continue with assembly programming, except this time, we divide our programs into *procedures*—modular chunks of instructions that perform a specific task that can be called upon from any location in your program.

A simple procedure

First, a few simple steps to get started:

1. Login to `remus/romulus`.
2. Open a terminal.
3. Create a directory for this lab and change into it:

```
$ mkdir lab-6
$ cd lab-6
```

4. Copy a source code file into your directory:

```
$ cp ~sfkaplan/public/cs16/lab-6/count.asm .
```

This `count.asm` program contains a fully formed `main()` procedure that begins the program. It creates an activation frame, calls the procedure `count()`, and then leaves the stack pointer referring to the return value from that procedure call. More specifically, as the comments in the code show, `main()` passes an array to `count()`, as well as a value for which `count()` should search. It expects `count()` to return the number of instances of that value in the array.

Your assignment: Write the `count()` procedure. It must rely not on the labels in the `.Numeric` section; instead, it should rely only on the arguments passed into it.

A recursive procedure

Now copy another source code file:

```
$ cp ~sfkaplan/public/cs16/lab-6/fib.asm .
```

Once again, `main()` has already been written to initialize the stack, push a new activation frame for a call to the procedure `fib()`, to perform that call, and then to leave its result at the top of the stack. Specifically, `fib()` is passed a single argument, `n`. The procedure must then calculate and return the n^{th} Fiboanacci number, defined as:

$$F_n = \begin{cases} n & \text{if } 0 \leq n \leq 1 \\ F_{n-1} + F_{n-2} & \text{if } n \geq 2 \end{cases}$$

Your assignment: Write the `fib()` procedure. You **must not** write it *iteratively*—that is, you cannot compute the answer by using a loop structure. Instead, you **must** write it *recursively*—that is, to obtain F_{n-1} and F_{n-2} , you must perform procedure calls to `fib()`. That's right, `fib()` must call itself.

How to submit your work

We will be using the `cs16-submit` command to turn in programming work. Specifically, you should submit your completed `count.asm` and `fib.asm`, like so:

```
cs16-submit lab-6 count.asm fib.asm
```

This assignment is due on Friday, November 20, at 11:59 pm