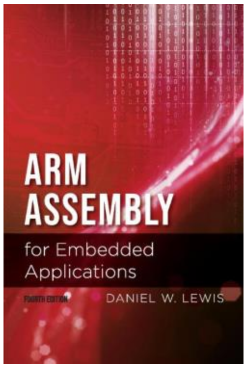


Programming Lab #2

Functions and Parameters

Prerequisite Reading: Chapters 1-3
Revised: January 30, 2019

Create an assembly language source code file containing six functions. If the functions were written in C, they would look like the following:

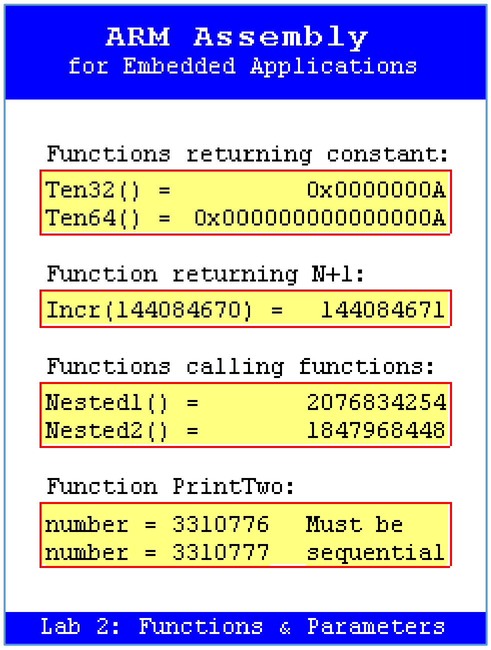
`uint32_t Ten32(void)`
`{`
`return 10 ;`
`}`
	`uint64_t Ten64(void)`
	`{`
	`return (uint64_t) 10 ;`
	`}`
		`uint32_t Incr(uint32_t n)`
		`{`
		`return n + 1 ;`
		`}`

`uint32_t Nested1(void)`
`{`
`return rand() + 1 ;`
`}`
	`uint32_t Nested2(void)`
	`{`
	`return rand() + rand() ;`
	`}`
		`void PrintTwo(char *format, uint32_t n)`
		`{`
		`printf(format, n) ;`
		`printf(format, n + 1) ;`
		`}`

Test your functions using the main program downloaded from here. If you code works correctly, the display should look like this image (although the numbers are likely to be different). Incorrect values will be displayed using white text on a red background.

IMPORTANT – The `.thumb_func` directive:

The "`.thumb_func`" assembler directive specifies that the next label is the entry point of a function that contains instructions from the Thumb subset of the ARM processor and causes the binary representation of instructions that branch to that label to be generated somewhat differently. Thus in a source code file that contains more than one function, it is imperative that you place a `.thumb_func` directive immediately before the entry point label of every function.



```
          ARM Assembly
    for Embedded Applications

 Functions returning constant:
 Ten32() =           0x0000000A
 Ten64() =   0x000000000000000A

 Function returning N+1:
 Incr(144084670) =    144084671

 Functions calling functions:
 Nested1() =         2076834254
 Nested2() =         1847968448

 Function PrintTwo:
 number = 3310776     Must be
 number = 3310777     sequential

 Lab 2: Functions & Parameters
```