Lab III

C Refresher Module

December 31, 2017

> *"Today's lab is difficult as our lives. First day of lab invokes my nihilism."*
>
> -Anonymous (2016)

1 Pointers & Assembly

This is the final lab for the refresher-module. Hope we lived up to your expectations. If not, sorry, not much can be done in three days anyway.

This is lab is going to about `pointers`, for which its safe to assume, is the most dreaded topic. It's the reason why C is still around. The goal of the lab is to understand how C & Assembly is tightly coupled. Your task will be to write an assembly function, which is to be called from a C program.

The objective of the lab is to understand the C calling convention, how function calls are made and how things really work.

You are given a simple `sum.c` program, which calls an `add` function. The implementation of the add function is not provided, but your task is to write that function in assembly, use all the concepts we tried to explore in the sessions, of-course use google, read more. – whatever works for you.

The end result should be an executable that works.

2 Submission

Let's trust you on this one and expect you would do the right thing. Please don't forget to include the writeup.

3 Example

```
mfrw@kp: cat sum.c
#include<stdio.h>
int main(int argc, char *argv[]) {
    int a = 10;
    int b = 20;
    printf("Sum is = %d \n", add(a, b)); //easy
    /* printf("Sum is = %d \n", add(&a, &b));  maybe tricky */
    return 0;
}
mfrw@kp:
mfrw@kp: cat add.asm
.section .text
.globl add
.type add, @function
add:
    ; your assembly code here
    ;
mfrw@kp:
mfrw@kp: gcc sum.c add.asm -o sum #  Makefile preferred
mfrw@kp: ./sum
Sum is = 30
mfrw@kp:
```

4 For those who want more

`Happy New Year !!!`
It's for ***you*** and us too. Finish the lab and off you go to enjoy.

5 I still want more

If you are still reading and you want more!
We would recommend going through this:
Smashing the stack for fun and profit. [`http://phrack.org/issues/49/14.html`]

> *“ You are ~~not~~ expected to*
> *Understand this.”*
>
> `-Ken Thompson`