LAB TWO

STACK FRAMES AND THE DEBUGGER

CS2263, Fall 2021

LEARNING OUTCOMES

At the conclusion of the lab, students should be able to

- Write C programs and interrogate them using the DDD debugger
- Within DDD move between and examine the stack frames

RESOURCES

- To use DDD you will have to use vnc. If you have not done so, review the documentation under Services & Support > Returning Students at http://help.cs.unb.ca:
- Compiling a C program and starting the debugger:

```
$ gcc -g -Wall -Wshadow prog.c -o prog
$ ddd
```

- Program p1.c from the textbook:

```
/* p1.c */
#include <stdio.h>
#include <stdlib.h>

int g1(int a, int b){
   int c = (a + b) * b;
   printf("g1: %d %d %d \n", a, b, c);
   return c;
}

int g2(int a, int b){
   int c = g1(a + 3, b - 11);
   printf("g2: %d %d %d \n", a, b, c);
   return c - b;
}

int main (int argc, char** argv){
   int a = 5;
   int b = 17;
   int c = g2(a - 1, b * 2);
   printf("main: %d %d %d \n", a, b, c);
   return EXIT_SUCCESS;
}
```

EXERCISE ONE

Modify the p1.c program from the textbook, page 27, by including the printing of the memory addresses of the variables a, b and c in each function (main, g1 and g2). For example:

```
printf("a's address is %p\n ", &a);
```

QUESTIONS:

- Are the values of the variables printed from your program the same as obtained by your colleagues? Why?
- Are the addresses printed from your program the same as obtained by your colleagues? Why?
- Are the addresses printed for the variables in the function g1 bigger or smaller than the addresses printed from the function g2? Why?

SUBMIT:

- the modified source code
- the screen shot of the output from your program
- The answers to the questions in this section

EXCERCISE TWO

Run the modified p1.c program in the debugger. Set up the breakpoints as described in the Section 2.7, page 28, in the textbook.

```
(gdb) b g1
(gdb) b g2
```

QUESTIONS:

- Are the stack addresses listed in backtrace related to the addresses of the variables a, b and c printed by the functions g1 and g2? Explain.

SUBMIT:

- the debugger screenshot showing the backtrace after reaching the breakpoint g2. How many frames are shown in the trace?
- the debugger screenshot showing the backtrace after reaching the breakpoint g1 (as described in Section 2.7). How many frames are shown?
- The answers to the questions in this section

EXCERCISE THREE

Run your program from your assignment 1 for finding the Fibonacci primes in the debugger. Set the breakpoints to the functions isFib and isPrime.

SUBMIT:

- One debugger screenshot showing the backtrace after reaching the breakpoint isFib.
- One debugger screenshot showing the backtrace after reaching the breakpoint isPrime.
- **Do not submit the entire source code**, we need only the debugger screenshots.

SUBMISSION

Before the due date for this lab, students should submit a single zip or tar file (named *LastName_FirstName_Lab2.zip or LastName_FirstName_Lab2.tar*) online to the lms containing:

- the required material for each question (use the headings indicating the question number) in a single pdf file (named *LastName_FirstName_Lab2.pdf)*
- Your source code directory:
    - This should include all of your source files, including any test programs.
    - This should not include object (.o) files and executables. Nobody needs to see those.