CMP SCI 3780 Project 1

Fall 2019

Due date: 9/17/2019

Total points possible: 65

The goal of this project is to reinforce what is going on in the call stack, as well as refreshing ourselves on the unix environment and using a debugger to examine memory locations. All tasks should be done on the student unix server delmar.umsl.edu. If you have never used a debugger to examine raw memory locations, now is the time!

Task 1

Your task here is to examine the size of the activation record for a function. In particular, I want to know the size in bytes on the stack that is taken up by all the data that is not simply local variables being stored. To do this, write a C++ program that contains several functions that call each other in a sequence. These functions should take in at least one variable by value. It would also be useful to set some local variables to some specific values inside of them. Then using GDB or some other debugger, set a breakpoint at the end of the lowest-level function and examine the memory locations of the stack (probably using x in gdb). You should be able to get an understanding of how much space is taken up by non-variable data. Note that you should keep these functions fairly simple as it will make your life easier. For demonstrating your work, I want you to write up a justification for your answer, including relevant evidence. This should include screen captures of at least some display by the debugger of memory, along with a description of its interpretation.

Task 2:

Write another program (in C++) that will allocate a local static array of integers and then a dynamic array of integers. Are they stored next to each other? You can examine this by examining the memory addresses where they are located. As described in class, on some systems the size of a dynamic array is actually stored in the bytes previous to a dynamically allocated array. Through some experiments on your own, try to see if this is true on delmar. Is this true or not true also for the local array? As in the first part, describe the procedure you used to test for this.

Task 3:

Write a program that prompts the user for two numbers and stores them in signed integers. The program should then add those two numbers together and store the result in a signed integer and display the result. Your program should then multiply them by each other and store the result in another integer and display the result. Then do the same but with dividing the first number by the second. Display an error message to the screen if an operation has happened that does not result in a correct calculation. In other words, make sure to test your code for error cases. You can safely assume I will only give your program integers (I will give your program only decimal digits).

Submission:

For submission I want all source code and a single document describing the experiments and the answers to the questions (and justifications) on the previous tasks. This document should be in either pdf or word. Embed any screenshots into those documents, I do not want pictures, jpeg, etc. Also include in this document any sourcecode at the end (yes, this would mean a separate source file and the source code embedded in this document also).