

Lab 8. Structured programming in assembly

Introduction

In this lab, we practice structured programming in both C and assembly by using the for, while, and do-while loops. Note that the assessment is only on the assembly version of the code—the C version is used for a pseudo code for your assembly code.

Lab tasks

You are provided with some base code for both C and assembly. You are also provided requirements, listed below, for writing C and assembly code for four tasks in the lab. Note that you need to read and understand the code of 190_structured_prog before getting started with your lab.

1. Using for loop in assembly

(25 points) Modify Task 3 of 190_structured_prog to use a for loop to find the index of the minimum value of an array.

2. Using while loop in assembly

(25 points) Modify Task 5 of 190_structured_prog to use a while loop to find the number of binary 1's in a 32-bit unsigned number.

3. Using do-while loop in assembly

(25 points) Modify Task 6 of 190_structured_prog to use a do-while loop to find the number of binary 1's in a 32-bit unsigned number. Note that you may have to refer to Task 9 of 170_conditional_execution for the implementation of the do-while loop in assembly.

4. Using any loop in assembly to perform some task

(25 points) Modify Task 4 of 190_structured_prog to use any loop to count the number of digits in terms of an additional parameter—base b —of an unsigned 32-bit number.

Report

Like Labs 5 and 6, this lab report is special as well. The results of Tasks 1, 2, and 3 will be used to assess an ABET program outcome. For each task to be evaluated by the ABET rubrics, your grading will usually be:

- U for less than 70 %
- S for higher than 70 % but less than 90 %

-
- E for higher than 90 %

You need to add appropriate comments to all your code. Copy all the code snippets you changed/modified together with your comments to your report. Please be reminded that you need to perform your programming as independently from the TA as possible.