

Computer Architecture and Low-Level Programming

Assignment 1

Basic Assembly Language Programming

Create a SPARC assembly language program that finds the minimum of $y = 2x^3 - 19x^2 + 9x + 45$ in the range $-2 \leq x \leq 10$, by stepping through the range one by one in a loop and testing. Use only integers for x , and do not factor the expression. Use the `printf()` function to display to the screen the values of x , y and the current *minimum* on each iteration of your loop. Also, put the value for the *minimum* into register `%l0` at the end of your program.

You are to create 2 versions of your program:

1. Write the program without macros (i.e. don't use `m4`), and do not fill the delay slots (i.e. use nop's after branch and call instructions).
2. Optimize the above program by filling delay slots (eliminate nop's where possible). Also, add macros to the above program to make it more readable (use `m4`). In particular, provide macros for all constants and heavily used registers.

Running Your Program

To verify that your assembly language program works, run both versions under `gdb`, capturing output from each session using `script`. For version 1, single step through the program (use `ni`), displaying the instruction being executed (use `display/i $pc`). Also print out the contents of particular registers (use `p`) at key points in your program to show that it is working as expected. For version 2, set a breakpoint just after the place where the final result is calculated, and then print out the final result (it should be in register `%l0`). Do not single step through this version.

Other Requirements

Make sure your code is properly formatted into columns, is readable and fully documented, and includes identifying information at the top of each file. You must comment each line of assembly code. Your code should also be well designed: make sure it is well organized, clear, and concise.

New Skills Needed for this Assignment:

- Ability to do basic arithmetic, loops and if-else constructs in assembly
- Ability to optimize assembly code by filling delay slots
- Ability to use macros in assembly code
- Ability to assemble programs using `gcc` and use `m4` to process macros
- Ability to use `gdb` to debug and display assembly language programs

Submit the following:

1. A hard copy of both versions of your assembly language program. Make sure your name, UNIX login name, ID, tutorial number, and TA's name appear on the title page.
2. Your assembly source code files for both programs and 2 scripts via electronic submission. Use the `submit` command to submit electronically. The TA will assemble and run your programs to test them. Name your programs `assign1.s` and `assign1.m`, and your scripts `script1` and `script2`.

Computer Architecture and Low-Level Programming

Assignment 1 Grading

Student: _____

Functionality (Version 1)

Equation calculation	4	_____
Test for minimum	4	_____
Display to screen	2	_____
Loop	4	_____

Optimization (Version 2)	4	_____
--------------------------	---	-------

Use of Macros (Version 2)	4	_____
---------------------------	---	-------

2 Scripts showing use of <i>gdb</i>	4	_____
-------------------------------------	---	-------

Complete documentation and commenting	4	_____
---------------------------------------	---	-------

Design quality	2	_____
----------------	---	-------

Total	32	_____	_____ %
--------------	-----------	-------	---------