

Project 4A: Writing Assembly Code

Due

Dec 7, 2021 by 11:59pm

Points

50

Submitting

a file upload

File Types

s

Available

Nov 28, 2021 at 12am - Dec 24, 2021 at 11:59pm

This assignment was locked Dec 24, 2021 at 11:59pm.

Project 4A: Writing Assembly Code

In this project, we will get some practice writing assembly code for a simple function call.

Warning

Do not write your code in C and use the compiler to produce the assembly for you!

Human written / readable assembly code is very different.

Follow best practices for writing correct/readable assembly code and do not optimize for performance.

Grading

50 points – Correct with good programming style

25 points – Correct without good programming style

0 points - Incorrect

We will be running the grading script again one week after the due date. You may submit as many times as you wish.

Programming style will be assessed by hand during finals week

A note on Project Difficulty

Students in the past have reported this project is “The most frustrating assignment they have ever completed” and “This project was really fun.” And, I've heard both comments from the same student. Writing code in assembly is likely something very new, and if you haven't done the recommended practice problems, you will find this project very frustrating. Start with the practice problems in the book before taking on the project.

Finding the maximum value in an array only takes a few minutes to code in C and maybe 20 minutes to code in assembly (if you’ve done the practice problems, attended the lecture, and follow best programming practices). Debugging can take hours – unless you use gdb - then it only takes a few minutes.








Corrections and Additions

None yet.

Specifications

1. Complete the template file find_maximum.s. This file contains exactly one function that returns the index of the maximum value in an array of integers.
2. Write the code in x86-64 assembly.
3. Make your code as human-readable as possible by adding comments for each short block of code as well as each line of assembly.
4. When combined with the provided drivers, your code must write the phrase “The maximum value in the array is found at index ???” where ??? is replaced with the correct index. This text is provided in the driver files.
5. You may print any other debugging code so long as the answer phrase is also printed.
6. Your code may not produce a segmentation fault, infinite loop, or command terminated errors. These errors will result in a grade of 0 on the assignment

Files

- [find_max.s](#)  – Do your work here.
- [driver4.s](#)  – Assembly code for a function that calls prepares a 4-element array and calls Find_Max(). This file also contains the code for the debugging function Print_One_Number(). Your function must work with this code.
- [driver5.s](#)  – Assembly code for a function that calls prepares a 5-element array and calls Find_Max(). This file also contains the code for the debugging function Print_One_Number(). Your function must work with this code.
- [driver4.c](#)  – C code for main with the 4-element array
- [driver5.c](#)  – C code for main with the 5-element array
- [find_max_template.c](#)  – C code provided to show the equivalent functionality of the provided find_max.s
- [find_max.c](#)  – C code that finds the maximum value in an array. (You really shouldn't need this)

Compiling, Running, and Debugging

Compile the code and produce the default executable

```
gcc -g find_max.s driver4.s -Wall
```

Run the default executable

```
./a.out
```

Write a new driver – edit driverX.c and compile to assembly with

```
gcc -S -o driverX.s driverX.c
```

Debug with gdb

```
gdb ./a.out
```

Use the gdb commands:

```
start
layout asm
stepi
nexti
info registers
help
x (to examine memory)
```

Write a new driver

Test your code with extra drivers. Try an array with the maximum value at index 0, at index n-1, and with duplicate values (the code should return the index of the first maximum)

Strategy

Compile and run the provided template. The provided template code prints the length of the array, prints the number 999, and returns -1 (which is printed in main).

1. Write comments first – 15 minutes – see find_max.c for the algorithm in C
2. Write code second – 20 minutes
3. Debug – hours (unless you use gdb)
4. **Do the smallest thing possible in assembly, then test!**

- Test Print_One_Number to make sure you understand how this debugging function works
- Write/complete the prologue and epilogue to set up the stack, push callee saved registers reserve memory, pop the callee saved registers, restore the stack memory and return.
- Make a table to keep track of where in memory you are storing your variables
- Be consistent in which registers you use. (e.g.)
 1. %rax for the array base
 2. %rdx for the index
 3. %rdi and %rsi for conditionals
 4. %rax for the return value
- Did you choose the correct size registers ‘r’ for addresses and ‘e’ for integers.
- Write a loop and print the index – choose good label names (e.g. TOP_OF_LOOP, CONDITIONAL, BOTTOM_OF_LOOP
- Print a[i] in your loop
- Print a[best_index] in the loop
- Write the conditional :: if (a[best_index] < a[i]) – choose good names for lables (e.g. TRUE_BLOCK, FALSE_BLOCK, END_CONDITIONAL)
- Write something in the TRUE_BLOCK
- Write something in the FALSE_BLOCK
- Prepare the return value

Style

- Comment!
 1. Use // for full-line comments
 2. Use /* */ for block-comments
 3. Use # for in line comments
 1. Movl -4(%rbp), %rdi # get max_index and store in rdi
 2. Comment every line!
- Use blank lines between statements that perform one action
- Indent code. Do not indent full-line comments or labels
- Align the Op code, operands, and comments in three columns.
- Choose good label names
 1. See suggestions above.
 2. You may not need all of the suggested labels depending on the style of loop / conditional statement you choose.
- Store all variables on the stack and retrieve them every time you use them.
 1. You may go back and write code before this section that overwrites a register you are assuming contains a variable.
 2. This technique will help guard against that type of error.

Turn in

Upload your completed find_max.s file to canvas


Submission


✓ **Submitted!**


Dec 13, 2021 at 6:40pm (late)
[Submission Details](#)
[Download find_max-1.s](#)


Grade: 50 (50 pts possible)
Graded Anonymously: no


Comments:


 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 9, 2021 at 3:11am

 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 9, 2021 at 3:55am

 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 10, 2021 at 2:34am

 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 11, 2021 at 2:44am

 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 12, 2021 at 1:12am

 See attached files.
[zhiweigian_314915_23435603_find](#)
ASMIT NAYAK, Dec 12, 2021 at 1:12am