Exercise Assignment Week 04: Conditionals, Functions, Parameters, Variables, and Stacks…

You are Sherlock Holmes, and have 2 weeks to solve some mysteries.

Due Monday October 17, midnight.

(ps. Happy Thanksgiving Long Weekend! ☺)

**Learning Objections:**

* Perform reverse engineering to test your understanding
* Introduction to branching and conditional statements

**Notes:**

1. To perform conditional checks, you do a comparison, followed by an action based on the result (each jump command checks appropriate flags after the compare operation, such as carry flag, or zero flag).

Example snippet:

CMP ebx,10 ;compares contents of ebx with 10

JLE there ;jumps (to label “there”) if ebx less than or equal 10

Other jumps (for signed data):

|  |  |
| --- | --- |
| JC | Jump if carry |
| JO | Jump if overflow |
| JL | Jump if lower |
| JG | Jump if greater |
| JE/JZ | Jump equal or jump zero |
| JNE/JNZ | Jump not equal, or jump not zero |
| JL | Jump if lower |
| JG | Jump if greater |
| JMP | No condition! Just do it! |

1. GCC has a flag to give more verbose information “gcc –fverbose-asm –S foo.c”

**Tasks:**

Create ***lastname\_firstname\_A4*** folder, and put the following exercises inside this folder. Start a **makefile**, where you can add each exercise as you work along.

Warmup: Review the code samples from “04 Tutorial Stack” document (D2L🡪 Contents🡪 Lecture). Some of them we did on Monday, but one of them is new. (You don’t need to hand these in).

1. Consider the file mystery1.s (generated by gcc –S –masm=intel) that contains MASM format assembly code, which is very close to NASM.

* After examination, provide comments for the file.
* Write the corresponding C algorithm/program (eg. mystery1.c)
* It should produce same output, and “similar” assembly (via gcc).
* Test and document

1. Consider the file mystery2.asm

* Same checklist as above

1. Consider the file mystery3.asm

* Same checklist as above

1. Write an assembly function that takes in 3 integer parameters, and returns the smallest of the three. Name it smallest.asm. Create a C driver to test it (print numbers, and results). Hint: Use “JL” or “JG” jump command.
2. Create YOUR OWN mystery. Anything. (Write some algorithm in C. Use gcc to generate assembly.) Post it onto D2L🡪 Discussions 🡪 Thread “Mysteries” Hone your assembly detective skills by solving each other’s mysteries ☺ I look forward to seeing the activity here.

**Marking:**

|  |  |
| --- | --- |
|  | Max |
| Completeness | 2 |
| Correctness | 2 |
| Files are well organized | 2 |
| Code is well commented and documented | 2 |
| Proper file naming conventions | 2 |