CPSC 240: Computer Organization and Assembly Language

Assignment 03, Fall Semester 2024

CWID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Quiz Questions:**

From the textbook "X86-64 Assembly Language Programming with Ubuntu," study quiz questions 13, 14, 15, and 16 on page 120 and page 121. Students do not need to submit answers to the quiz questions as they are found in Appendix D of the textbook.

**Programming:**

1. Download the “CPSC-240 Assignment03.docx” document.
2. Design a 32-bit multiplication program "multiplication.asm", and use assembly language to realize the function of the following C++ instructions. NOTE: variable sizes and program functions should be equivalent to C/C++ instructions.  
    unsigned int num1 = 300,000; // use dd to declare 32-bit variable  
    unsigned int num2 = 400,000; // use dd to declare 32-bit variable  
    unsigned long product = 0; // use dq to declare 64-bit variable  
    product = long(num1 \* num2);
3. Assemble the "multiplication.asm" file and link the "multiplication.o" file to get the "multiplication" executable file.
4. Run the "multiplication" file with the GDB debugger to display the simulation results of num1 and num2, as well as the simulation results of product.
5. Insert source code (multiplication.asm) and simulation results (GDB panel) of the memory (num1, num2, and product) in the document. Use calculator or hand calculation to verify simulation results.
6. Design a 32-bit division program "division.asm", and use assembly language to realize the function of the following C++ instructions. NOTE: variable sizes and program functions should be equivalent to C/C++ instructions.  
    unsigned long num1 = 50,000,000,000; // use dq to declare 64-bit variable  
    unsigned int num2 = 3,333,333; // use dd to declare 32-bit variable  
    unsigned int quotient = 0, remainder = 0; // use dd to declare 32-bit variable  
    quotient = num1 / num2;  
    remainder = num1 % num2;
7. Assemble the "division.asm" file and link the "division.o" file to get the "division" executable file.
8. Run the "division" file with the GDB debugger to display the simulation results of num1 and num2, as well as the simulation results of quotient and remainder.
9. Insert source code (division.asm) and simulation results (GDB panel) of the memory (num1, num2, quotient, and remainder) in the document. Use calculator or hand calculation to verify simulation results.
10. Save the file in pdf format and submit the pdf file to Canvas before the deadline.

[Insert multiplication assembly source code here]

[Insert multiplication simulation results (GDB panel) here]

[Insert the multiplication result verification here]

[Insert division assembly source code here]

[Insert division simulation results (GDB panel) here]

[Insert the division result verification here]