CPSC 240: Computer Organization and Assembly Language
Assignment 03, Fall Semester 2024

CWID: 885024539   Name: Riley Blacklock

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.

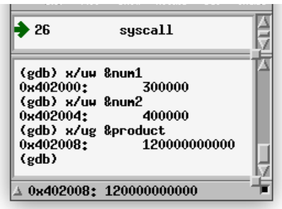
```
unsigned long num1 = 50,000,000,000;        // use dq to declare 64-bit variable
unsigned int num2 = 3,333,333;              // use dd to declare 32-bit variable
unsigned int quotient = 0, remainder = 0;   // use dd to declare 32-bit variable
quotient = num1 / num2;
remainder = num1 % num2;
```

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]

```
; multiplication.asm;
; unsigned int num1 = 300000;
; unsigned int num2 = 400000;
; unsigned long product = 0;
; product = long(num1 * num2);

section .data
SYS_exit   equ 60
EXIT_SUCCESS      equ 0
num1          dw 300000
num2          dw 400000
product             dq 0

section .text
  global _start
_start:
  mov eax, dword[num1]
  mov ebx, dword[num2]
  mul  ebx

  mov [product], eax
  mov [product+4], edx

  mov rax, SYS_exit
  mov rdi, EXIT_SUCCESS
  syscall
```

[Insert multiplication simulation results (GDB panel) here]



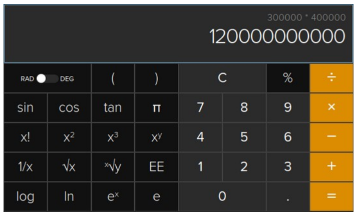
```
26          syscall

(gdb) x/uw &num1
0x402000:       300000
(gdb) x/uw &num2
0x402004:       400000
(gdb) x/ug &product
0x402008:       120000000000
(gdb)

0x402008: 120000000000
```

num1 = 300,000

num2 = 400,000

product = 120,000,000,000

[Insert the multiplication result verification here]



```
300000 * 400000
120000000000
```

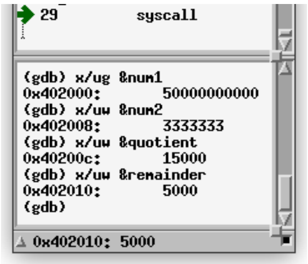
num1   x num2   = product

300,000 x 400,000 = 120,000,000,000

[Insert division assembly source code here]

```
; division.asm;
; unsigned long num1 = 50,000,000,000;
; unsigned int num2 = 3,333,333;
; unsigned int quotient = 0, remainder = 0;
; quotient = num1 / num2;
; remainder = num1 % num 2;

section .data
SYS_exit   equ 60
EXIT_SUCCESS      equ 0
num1         dq 50000000000
num2         dd 3333333
quotient     dd 0
remainder  dd 0

section .text
  global _start
_start:
  mov eax, dword[num1]
  mov edx, dword[num1+4]
  mov ebx, dword[num2]
  div   ebx

  mov [quotient], eax
  mov [remainder], edx

  mov rax, SYS_exit
  mov rdi, EXIT_SUCCESS
  syscall
```

[Insert division simulation results (GDB panel) here]



```
29          syscall

(gdb) x/ug &num1
0x402000:       50000000000
(gdb) x/uw &num2
0x402008:       3333333
(gdb) x/uw &quotient
0x40200c:       15000
(gdb) x/uw &remainder
0x402010:       5000
(gdb)
0x402010: 5000
```

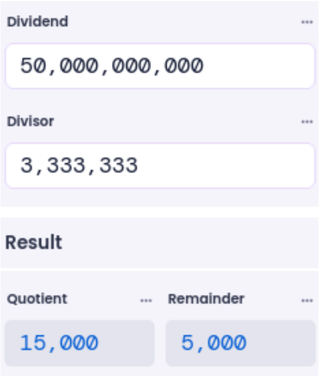
num1 = 50,000,000,000

num2 = 3,333,333

quotient = 15,000

remainder = 5,000

[Insert the division result verification here]



Dividend

50,000,000,000

Divisor

3,333,333

Result

Quotient

15,000

Remainder

5,000

num1 / num2 = quotient

num1 % num2 = remainder

50,000,000,000 / 3,333,333 = 15,000

50,000,000,000 % 3,333,333 = 5,000