

# CPSC 240: Computer Organization and Assembly Language

## Assignment 02, Fall Semester 2024

CWID: 885024539 Name: Riley Blacklock

### Quiz Questions:

From the textbook "X86-64 Assembly Language Programming with Ubuntu," study quiz questions 8, 9, 10, and 11 on page 120. 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 Assignment02.docx" document.
2. Design a 16-bit addition program "addition.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 short num1 = 0xFEDC;           // use dw to declare 16-bit variable
unsigned short num2 = 0x1234;           // use dw to declare 16-bit variable
unsigned int sum = 0;                   // use dd to declare 32-bit variable
sum = int(num1 + num2);
```

3. Assemble the "addition.asm" file and link the "addition.o" file to get the "addition" executable file.
4. Run the "addition" file with the GDB debugger to display the simulation results of num1 and num2, as well as the simulation results of sum.
5. Insert source code (addition.asm) and simulation results (GDB debugger window) of the memory (num1, num2, and sum) in the document. Use calculator or hand calculation to verify the simulation results.
6. Design a 16-bit subtraction program "subtraction.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.**

```
signed short num1 = 0x1234;             // use dw to declare 16-bit variable
signed short num2 = 0xFEDC;             // use dw to declare 16-bit variable
signed int dif = 0;                     // use dd to declare 32-bit variable
dif = int(num1 - num2);
```

7. Assemble the "subtraction.asm" file and link the "subtraction.o" file to get the "subtraction" executable file.
8. Run the "subtraction" file with the GDB debugger to display the simulation results of num1 and num2, as well as the simulation results of dif.
9. Insert source code (subtraction.asm) and simulation results (GDB debugger window) of the memory (num1, num2, and dif) in the document. Use calculator or hand calculation to verify the simulation results.
10. Save the file in pdf format and submit the pdf file to Canvas before the deadline.

[Insert addition source code here]

```
; addition.asm;
; unsigned short num1 = 0xFEDC;
; unsigned short num2 = 0x1234;
; unsigned int sum = 0;
; sum = int(num1 + num2);
```

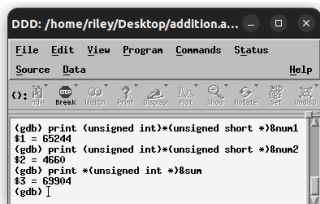
```
section .data
SYS_exit equ 60
EXIT_SUCCESS equ 0
num1      dw 0xFEDC
num2      dw 0x1234
sum       dd 0
```

```
section .text
global _start
_start:
    mov ax, word[num1]
    add ax, word[num2]
    adc dx, 0

    mov [sum], ax
    mov [sum+2], dx

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

[Insert addition simulation result here]

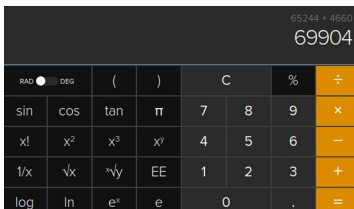


num1 = 65244

num 2 = 4660

sum = 69904

[Insert the addition result verification here]



65244 + 4660 = 69904

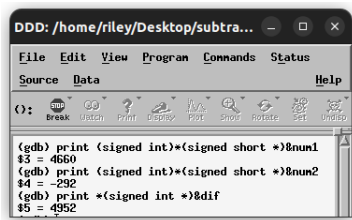
[Insert subtraction source code here]

```
; subtraction.asm;  
; signed short num1 = 0x1234;  
; signed short num2 = 0xFEDC;  
; signed int dif = 0;  
; dif = int(num1 - num2);
```

```
section .data  
SYS_exit equ 60  
EXIT_SUCCESS equ 0  
num1 dw 0x1234  
num2 dw 0xFEDC  
dif dd 0
```

```
section .text  
global _start  
_start:  
    mov eax, dword[num1]  
    sub eax, dword[num2]  
    sbb edx, 0  
  
    mov [dif], eax  
    mov [dif+2], edx  
  
    mov rax, SYS_exit  
    mov rdi, EXIT_SUCCESS  
    syscall
```

[Insert subtraction simulation result here]

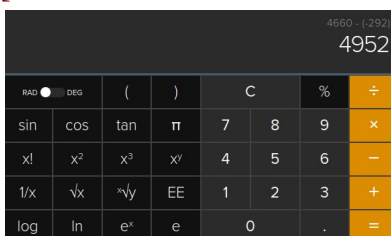


num1 = 4660

num2 = -292

dif = 4952

[Insert the subtraction result verification here]



$4660 - (-292) = 4952$