

CPSC 240: Computer Organization and Assembly Language

Assignment 02, Spring Semester 2023

CWID: [886880327](#) Name: [Andrew Saldana](#)

1. Download the "CPSC-240 Assignment02.docx" document.
2. Design the "addition.asm" program, and use assembly language to realize the function of the following C++ instructions.

```
short num1 = 30000;  
short num2 = 40000;  
int sum = 0;  
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 DDD 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 (DDD debugger window) of the memory (num1, num2, and sum) in the document. Write an analysis to verify simulation results.
6. Design the "subtraction.asm" program, and use assembly language to realize the function of the following C++ instructions.

```
short num1 = 30000;  
short num2 = 40000;  
int dif = 0;  
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 DDD debugger to display the simulation results of num1 and num2, as well as the simulation results of diff.
9. Insert source code (subtraction.asm) and simulation results (DDD debugger window) of the memory (num1, num2, and dif) in the document. Write an analysis to verify simulation results.
10. Save the file in pdf format and submit the pdf file to Canvas before 23:59 pm on 02/15/2023.

[Insert addition.asm here]

```
1 ; ex2_addition.asm
2 ; short num1 = 30000, num2 = 40000
3 ; int sum = 0;
4 ; sum = int(num1 + num2);
5
6 section .data
7     num1    dw    30000                ;num1 = 7530h
8     num2    dw    40000                ;num2 = 9C40h
9     sum     dd    0                    ;sum = 0000h
10
11 section .text
12     global _start
13
14 _start:
15     mov     bx, 0
16     mov     ax, word[num1]
17     add     ax, word[num2]
18                                     ;ax = num1 = 7430h
19                                     ;ax = ax + num2 = 1170h(4464 in
20                                     ;decimal) which is the remainder after
21                                     ;all 65536 of ax register is used up
22     adc     bx, 0
23     mov     word[sum+0], ax
24     mov     word[sum+2], bx
25                                     ;bx = bx + 0 + CF = 0001h = 1
26                                     ;sum = ax = 1170Ch
27                                     ;sum = ax & bx = 00011170h = 70000
28
29     mov     rax, 60
30     mov     rdi, 0
31     syscall
32                                     ;terminate excuting process
33                                     ;exit status
34                                     ;calling system services
```

[Insert addition simulation result here]

```
1 ; ex2_addition.asm
2 ; short num1 = 30000, num2 = 40000
3 ; int sum = 0;
4 ; sum = int(num1 + num2);
5
6 section .data
7     num1    dw    30000
8     num2    dw    40000
9     sum     dd    0
10
11 section .text
12     global _start
13
14 _start:
15     mov     bx, 0
16     mov     ax, word[num1]
17     add     ax, word[num2]
18
19
20
21     adc     bx, 0
22     mov     word[sum+0], ax
23     mov     word[sum+2], bx
24
25     r
26     mov     rax, 60
27     mov     rdi, 0
28     syscall
```

DDD: Registers		
Registers		
rax	0x1170	4464
rbx	0x1	1
rcx	0x0	0
rdx	0x0	0
rsi	0x0	0
rdi	0x0	0
rbp	0x0	0x0
rsp	0x7fffffff040	0x7fffffff040
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x0	0
r12	0x0	0

Integer registers All registers

Close Help

```
(gdb) x/xh &num1
0x402000: 0x7530
(gdb) x/xh &num2
0x402002: 0x9c40
(gdb) x/xw &sum
0x402004: 0x00011170
(gdb) x/dh &num1
0x402000: 30000
(gdb) x/uh &num2
0x402002: 40000
(gdb) x/dw &sum
0x402004: 70000
(gdb) |
```

0x402004: 70000

[Insert addition simulation result analysis here]

30000 to hex

$$\begin{array}{rcl}
 30000/16 & = & 1875 \quad R \ 0 \\
 1875/16 & = & 117 \quad R \ 3 \\
 117/16 & = & 7 \quad R \ 5 \\
 7/16 & = & 0 \quad R \ 7
 \end{array}
 \left. \vphantom{\begin{array}{rcl} 30000/16 \\ 1875/16 \\ 117/16 \\ 7/16 \end{array}} \right\} \begin{array}{l} \text{Hex: } 7530h \\ \text{binary: } 0111010100110000 \end{array}$$

40000 in hex

$$\begin{array}{rcl}
 40000/16 & = & 2500 \quad R \ 0 \\
 2500/16 & = & 156 \quad R \ 4 \\
 156/16 & = & 9 \quad R \ 12 \\
 9/16 & = & 0 \quad R \ 9
 \end{array}
 \left. \vphantom{\begin{array}{rcl} 40000/16 \\ 2500/16 \\ 156/16 \\ 9/16 \end{array}} \right\} \begin{array}{l} \text{Hex: } 9C40h \\ \text{binary: } 1001110001000000 \end{array}$$

30000 + 40000

$$\begin{array}{r}
 0111010100110000 \leftarrow 30000 \\
 1001110001000000 \leftarrow 40000 \\
 \hline
 1000100010110000 \leftarrow 70000 \text{ signed}
 \end{array}$$

hex = 11170h

70000	hex	binary
	11170h	0001000100010110000

[Insert subtraction.asm here]

```
1 ; ex2_subtraction.asm
2 ; short num1 = 30000, num2 = 40000
3 ; int dif = 0;
4 ; dif = int(num1 - num2);
5
6 section .data
7     num1    dw    30000                ;num1 = 7530h = 30000
8     num2    dw    40000                ;num2 = 9C40h = 40000
9     dif     dw    0                    ;dif = 0000h
10
11 section .text
12     global _start
13
14 _start:
15     mov     ax, word[num1]              ;ax = num1 = 7530h
16     sub     ax, word[num2]              ;ax = ax + -num2 = D8F0h
17     sbb     bx, 0                        ;bx = bx - 0 - CF = ffffh
18     mov     word[dif + 0], ax           ;dif = ax = D8F0h
19     mov     word[dif + 2], bx           ;dif = ax & bx = ffffD8F0h
20
21     mov     rax, 60                      ;terminate executing process
22     mov     rdi, 0                      ;exit status
23     syscall                             ;calling system services
```

[Insert subtraction simulation result here]

The screenshot shows a debugger window with two panes. The left pane displays assembly code for 'ex2_subtraction.asm'. The right pane, titled 'DDD: Registers', shows the state of various registers. Below the registers, there is a command window showing the results of GDB commands.

Assembly Code (Left Pane):

```
1 ; ex2_subtraction.asm
2 ; short num1 = 30000, num2 = 40000
3 ; int dif = 0;
4 ; dif = int(num1 - num2);
5
6 section .data
7     num1    dw    30000
8     num2    dw    40000
9     dif     dd    0
10
11 section .text
12     global _start
13
14 _start:
15     mov     ax, word[num1]
16     sub     ax, word[num2]
17     sbb     bx, 0
18     mov     word[dif + 0], ax
19     mov     word[dif + 2], bx
20
21     mov     rax, 60
22     mov     rdi, 0
23     syscall
```

Registers (Right Pane):

Register	Value (Hex)	Value (Dec)
rax	0xd8f0	55536
rbx	0xffff	65535
rcx	0x0	0
rdx	0x0	0
rsi	0x0	0
rdi	0x0	0
rbp	0x0	0x0
rsp	0x7fffffff030	0x7fffffff030
r8	0x0	0
r9	0x0	0
r10	0x0	0
r11	0x0	0
r12	0x0	0
r13	0x0	0

Command Window (Bottom):

```
(gdb) x/xh 8num1
0x402000: 0x7530
(gdb) x/xh 8num2
0x402002: 0x9c40
(gdb) x/xw 8dif
0x402004: 0xffffd8f0
(gdb) x/dh 8num1
0x402000: 30000
(gdb) x/uh 8num2
0x402002: 40000
(gdb) x/dh 8dif
0x402004: -10000
(gdb) |
```

[Insert subtraction simulation result analysis here]

30000 to hex

$$\begin{aligned} 30000/16 &= 1875 \quad R 0 \\ 1875/16 &= 117 \quad R 3 \\ 117/16 &= 7 \quad R 5 \\ 7/16 &= 0 \quad R 7 \end{aligned}$$

Hex: 7530h

binary: 0111010100110000

40000 in hex

$$\begin{aligned} 40000/16 &= 2500 \quad R 0 \\ 2500/16 &= 156 \quad R 4 \\ 156/16 &= 9 \quad R 12 \\ 9/16 &= 0 \quad R 9 \end{aligned}$$

Hex: 9C40h

binary: 1001110001000000

30000 + (-40000)

$$\begin{array}{r} 1001110001000000 \quad \leftarrow 40000 \\ 0110001101111111 \quad \leftarrow 1's \text{ complement} \\ + 0000000000000001 \\ \hline 0110001111000000 \quad \leftarrow 2's \text{ complement} = -40000 \\ + 0111010100110000 \quad \leftarrow 30000 \\ \hline 1101100011110000 \quad \leftarrow -10000 \text{ in signed} \\ \text{hex} = D8F0h \end{array}$$

-10000	hex	binary
	D8F0h	1101100011110000