Report For Lab 9

Connor McCullough

EEN 312

24 April 2013

**Objective:**

Continue to learn how to debug and practice doing some arithmetic instructions while debugging. Also understand the use of data segments and code segments in assembly programs.

**Equipment:**

Windows Visual Studio 2010, step debugger, breakpoints, registers

**Results:**

*What is the instruction pointer value associated with the instruction MOV EDX, 020:* 00401039

*What are results of EAX after ADD EAX, EBX, executed?*

00000320

*What are results of EAX after SUB EAX, ECX executed?*

000002EE

*What are results of EDX and EAX after MUL EAX executed?*

000002ED

**Conclusion:** The program worked for the step debugging and the proper values were acquired for the results.

**Code:**

TITLE Lab 9 debug-multiplication

INCLUDE Irvine32.inc

.DATA ;DATA SEGMENT

.STACK 100H ;STACK SEGMENT

.CODE ;CODE SEGMENT

MAIN PROC

            CALL Clrscr

            MOV EAX, 1000 ; EAX=1000

            MOV EBX, -200 ; EBX=-200

            MOV ECX, 050 ; ECX=50

            MOV EDX, -020 ; EDX=-20

            ADD EAX, EBX ; EAX=EAX+EBX

            SUB EAX, ECX ; EAX = EAX-ECX

            IMUL EDX ; EDX:EAX=EAX\*EDX

            CALL OUTPUT

            CALL Crlf

            EXIT

MAIN ENDP

output proc

            mov ECX, EAX ;lower 32-bit of result is saved in ECX

            mov EAX, EDX ;upper 32-bit is moved to EAX

cmp EDX, 0ffffffffh ;if upper 32-bit has only one negative sign info, go to neglower

jz neglower

cmp eax, 0h ;if upper 32-bit has only positive sign info, go to poslower

jz poslower

mov esi, edx

and esi, 800000000h

jz pos

;output 64-bit negative result in HEX format

mov al, ‘-’

call WriteChar

not ecx

not edx

add ecx, 1

add edx, 0

move ax, edx

call WriteHex

move eax, ecx

call WriteHex

mov al, ‘h’

call WriteChar

ret

;output positive 64-bit product in HEX format

pos: mov al, ‘+’

call WriteChar

move eax, edx

call WriteHex

move ax,ecx

call WriteHex

mov al, ‘h’

call WriteChar

ret

;output 32-bit result

neglower:

mov esi, ecx

and esi, 800000000h

jz addnegsign

move eax, ecx

call WriteInt

ret

addnegsign:

mov al, ‘-‘

call WriteChar

move eax, ecx

call WriteDec

ret

poslower:

mov al, ‘+’

call WriteChar

move ax, ecx

call WriteDec

ret

output endp

end main