Undergraduate Lab Report of Jinan University

Course Title Computer Organization Evaluation

Lab Name Lab 1: Manipulating Bits Instructor SUN Heng

Lab Address N116

Student Name TSUI, Hon Fai Student No 2018058445

College International School

Department Major CST

Date 15 / 10 / 2019 ~~Morning~~ / Afternoon

1. Introduction

1) Objective

The purpose of this assignment is to become more familiar with the assembly language by testing a simple assembly program.

2) Logistics

An assembly file is assembled and then executed on the computer, on Windows environment. Register are dumped at the end of the execution, in order to see through the operations.

2. Instructions

1. Installing the Microsoft Macro Assembler version 6.15
2. CD C:\Masm615.
3. Use MASM to assemble the provided soure code
4. Execute the program

Assuming that no errors were generated, the following files would be created in the current directory:

- myprogram.obj Object file
- myprogram.lst Listing file
- myprogram.exe Executable file

The Title directive marks the entire line as a comment. You can put anything you want on this line.

All text to the right of a semicolon is ignored by the assembler, so we use it for comments.

The Include directive copies necessary definitions and setup information for a text file named Irvine32.inc, located in the assembler's Include directory.

The .data directive is used to define the variables.

The .code directive marks the beginning of the program, where all

executable statements in a program are located.

The Proc directive identifies the beginning of a procedure. The name chose is main.

The Call statement calls a procedure that displays the current values of the CPU registers.

The exit statement calls a MS-Windows function that halts the program.

The End directive marks the last line of the program to be assemble.

3. Lab Devices

MASM on Window 7 x86

Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz 3.40 GHz

4.00 GB (3.43 GB 可用)

32 位操作系统

4. Results

```
C:\Masm615>myprogram.exe

  EAX=00030000  EBX=7FFDF000  ECX=00000000  EDX=00401005
  ESI=00000000  EDI=00000000  EBP=0012FF94  ESP=0012FF8C
  EIP=0040102B  EFL=00000206  CF=0  SF=0  ZF=0  OF=0
```

5. Discussion

During the experiment, I had a glance on mechanics of assembly language. The assembler generates object file, list file and executable file in the process, with the list file in plain text and the rest as binary file. Initially there was a problem executing the assembler as I was unfamiliar with Windows console environment although that was easy to solve by learning the command parameters of the 'cd' command. (ie. '/d' is required when the directory is not on the same volume as the current one).

6. Question

Please tell us the functions of four directives Mov, Add, Sub, Mov in the program?

%eax is a register and val* are memory allocated to store a dword variable. Below is a brief explanation to the operations.

A. mov eax,val1 : copy value from val1 to %eax.
   Then %eax = 10000

B. add eax,val2 : add value of val2 to value in %eax.
Then %eax = 50000

C. sub eax,val3 : subtract val3 from value in %eax.
Then %eax = 30000

D. mov finalVal,eax : copy value from register %eax to finalVal in memory.

E. After the calculation, results are shown by DumpRegs.

7. Appendix（Program Code）

```
TITLE Add and Subtratct              (AddSub2.asm)
; This program adds and subtracts 32-bit integers

INCLUDE Irvine32.inc
.data
val1        dword   10000h
val2        dword   40000h
val3        dword   20000h
finalVal    dword   ?
.code
main PROC
    mov eax,val1          ; start with 10000h
    add eax,val2          ; add 40000h
    sub eax,val3          ; subtract 20000h
    mov finalVal,eax      ; store the result (30000h)
    call DumpRegs         ; display the registers
    exit
main ENDP
END main
```