Laboratory 6

6.1. x86 Processor Architecture

Read and analyze Chapter 2 from Kip Irvine, Assembly language. Answer the proposed questions in detail, even if the question requires a True or False answer – it still needs to be explained.

The question numbers you will answer depend on your list number, and will continue by counting every 5th question, check your list number from LIST Groups.xlsx.
If your number in the list is greater than 26, you have to decrease it by 5; otherwise, add 5. You will answer a total of 5 questions. (If you have answered two questions on the theory lesson, take three more to make a total of five.)

The question**:**

1. In 32-bit mode, aside from the stack pointer (ESP), what other register points to variables on the stack?
2. Name at least four CPU status flags.
3. Which flag is set when the result of an unsigned arithmetic operation is too large to fit into the destination?
4. Which flag is set when the result of a signed arithmetic operation is either too large or too small to fit into the destination?
5. Which flag is set when an arithmetic or logical operation generates a negative result?
6. Which part of the CPU performs floating-point arithmetic?
7. On a 32-bit processor, how many bits are contained in each floating-point data register?
8. At which level(s) can an assembly language program manipulate input/output?
9. Why do game programs often send their sound output directly to the sound card's hardware ports?

10. (True/False): When a register operand size is 32 bits and the REX prefix is used, the R8D register is available for programs to use.
11. (True/False): The x86-64 instruction set is backward-compatible with the x86 instruction set.
12. (True/False): In current 64-bit chip implementations, all 64 bits are used for addressing.
13. (True/False): The Itanium instruction set is completely different from the x86 instruction set.
14. (True/False): Static RAM is usually less expensive than dynamic RAM.
15. (True/False): The 64-bit RDI register is available when the REX prefix is used.
16. (True/False): In native 64-bit mode, you can use 16-bit real mode, but not the virtual-8086 mode.
17. (True/False): The x86-64 processors have 4 more general-purpose registers than the x86 processors.

18.(True/False): The 64-bit version of Microsoft Windows does not support virtual-8086 mode.
19.(True/False): DRAM can only be erased using ultraviolet light.
20.(True/False): In 64-bit mode, you can use up to eight floating-point registers.
21.(True/False): A bus is a plastic cable that is attached to the motherboard at both ends, but does not sit directly on the motherboard.
22.(True/False): CMOS RAM is the same as static RAM, meaning that it holds its value without any extra power or refresh cycles.
23.(True/False): PCI connectors are used for graphics cards and sound cards.
24.(True/False): The 8259A is a controller that handles external interrupts from hardware devices.
25.(True/False): The acronym PCI stands for programmable component interface.
26.(True/False): VRAM stands for virtual random access memory.

6.2 **Assembly language. Fundamentals.**

Read and analyze Chapter 3 from Kip Irvine, Assembly language.
Write the code program, add comments and explain how it works. Please attach all program files on ELSE.

6.2.1. Short program

If your number in the list is greater than 26, you have to decrease it by 3; otherwise, add 3. You will make a total of 4 points from the next 15 question block.

1. Define four symbolic constants that represent integer 25 in decimal, binary, octal, and hexadecimal formats.
2. Find out, by trial and error, if a program can have multiple code and data segments.
3. Create a data definition for a doubleword that stored it in memory in big endian format.
4. Find out if you can declare a variable of type DWORD and assign it a negative value. What does this tell you about the assembler's type checking?
5. Write a program that contains two instructions: (1) add the number 5 to the EAX register, and (2) add 5 to the EDX register. Generate a listing file and examine the machine code generated by the assembler. What differences, if any, did you find between the two instructions?
6. Given the number 456789ABh, list out its byte values in little-endian order.
7. Declare an array of 120 uninitialized unsigned doubleword values.
8. Declare an array of byte and initialize it to the first 5 letters of the alphabet.
9. Declare a 32-bit signed integer variable and initialize it with the smallest possible negative decimal value. (Hint: Refer to integer ranges in Chapter 1.)

10. Declare an unsigned 16-bit integer variable named wArray that uses three initializers.
11. Declare a string variable containing the name of your favorite color. Initialize it as a nullterminated string.
12. Declare an uninitialized array of 50 signed doublewords named dArray.
13. Declare a string variable containing the word "TEST" repeated 500 times.
14. Declare an array of 20 unsigned bytes named bArray and initialize all elements to zero.
15. Show the order of individual bytes in memory (lowest to highest) for the following doubleword variable: val1 DWORD 87654321h.

6.2.2. Programing exercises I. (For everyone)

1. Integer Expression Calculation

Using the AddTwo program from Section 3.2 as a reference, write a program that calculates the following expression, using registers: A = (A + B) − (C + D). Assign integer values to the EAX, EBX, ECX, and EDX registers.

2. AddVariables Program

Modify the AddVariables program so it uses 64-bit variables. Describe the syntax errors generated by the assembler and what steps you took to resolve the errors.

6.2.3. Programing exercises II.

(If your number is even, complete the even-numbered exercises; if it is odd, complete the odd-numbered ones.)

1. Symbolic Integer Constants

Write a program that defines symbolic constants for all seven days of the week. Create an array variable that uses the symbols as initializers.

2. Data Definitions

Write a program that contains a definition of each data type listed in Table 3-2 in Section 3.4. Initialize each variable to a value that is consistent with its data type.

3. Symbolic Text Constants

Write a program that defines symbolic names for several string literals (characters between quotes). Use each symbolic name in a variable definition.

4. Listing File for AddTwoSum

Generate a listing file for the AddTwoSum program and write a description of the machine code bytes generated for each instruction. You might have to guess at some of the meanings of the byte values.