

Assembly Language (Review Section)

❖ Section 1.1 Review

1. How do assemblers and linkers work together?

An assembler converts source-code programs from assembly language into machine language. A linker combines individual files created by an assembler into a single executable program.

2. How will studying assembly language enhance your understanding of operating systems?

Assembly language is a good tool for learning how application programs communicate with the computer's operating system via interrupt handlers, system calls, and common memory areas. Assembly language programming also helps when learning how the operating system loads and executes application programs.

3. What is meant by a one-to-many relationship when comparing a high-level language to a machine language?

In a one-to-many relationship, a single statement expands into multiple assembly language or machine instructions.

4. Explain the concept of portability as it applies to programming languages.

A language whose source programs can be compiled and run on a wide variety of computer systems is said to be portable.

5. Is the assembly language for x86 processors the same as those for computer systems such as the Vax or Motorola 68x00?

No. Each assembly language is based on either a processor family or a specific computer.

6. Give an example of an embedded systems application.

Some examples of embedded systems applications are automobile fuel and ignition systems, ai-conditioning control systems, security systems, flight control systems, hand-held computers, modems, printers, and other intelligent computer peripherals.

7. What is a device driver?

Device drivers are programs that translate general operating system commands into specific references to hardware details that only the manufacturer knows.

8. Do you suppose type checking on pointer variables is stronger (stricter) in assembly language or in C and C++?

C++ does not allow a pointer of one type to be assigned to a pointer of another type. Assembly language has no such restrictions regarding pointers.

9. Name two types of applications that would be better suited to assembly language than a high-level language.

Applications suited to assembly language: hardware device driver and embedded systems and computer games require direct hardware access.

10. Why would a high-level language not be an ideal tool for writing a program that directly accesses a particular brand of printer?

A high-level language may not provide for direct hardware access. Even if it does, awkward coding techniques must often be used, resulting in possible maintenance problems.

11. Why is assembly language not usually used when writing large application programs?

Assembly language has minimal formal structure so structure must be imposed by programmers who have varying levels of experience. This leads to difficulties maintaining existing code.

12. Challenge: Translate the following C++ expression to assembly language using the example presented earlier in this chapter as a guide: $X = (Y * 4) + 3$

```
mov eax, Y
```

```
mov ebx, 4
```

```
imul, ebx  
add eax, 3  
mov X, eax
```

❖ Section 1.2 Review

1. In your own words, describe the virtual machine concept.

It's a set of layers that allow assembly language to be decoded in machine language.

2. Why do you suppose translated programs often execute more quickly than interpreted ones?

Translated Programs run faster, due to having code directly executed on the machine. Interpretation Programs interprets and execute the instructions one by one.

3. (True/False): When an interpreted program written in language L1 runs, each of its instructions is decoded and executed by a program written in language L0.

True

4. Explain the importance of translation when dealing with languages at different virtual machine levels.

L1 program is completely translated into an L0 program, which then runs on the computer hardware.

5. At which level does assembly language appear in the virtual machine example shown in this section?

At level 3 of the specific machine level

6. What software utility permits compiled Java programs to run on almost any computer

Java Virtual Machine (JVM)

7. Name the four virtual machine levels named in this section, from lowest to highest.

Digital Logics, Instruction Set Architecture, Assembly Language, High Level Language

8. Why don't programmers write applications in machine language?

Machine language is hard to understand and to provide code is difficult to read and write.

9. Machine language is used at which level of the virtual machine shown in Figure 1-1?

Level 2 - Instruction Set Architecture

10. Statements at the assembly language level of a virtual machine are translated into statements at which other level?

Level 2 - Instruction Set Architecture

❖ Section 1.3 Review

1. Explain the term Least Significant Bit (LSB).

Least significant bit (bit 0). Bit to the far right.

2. Explain the term Most Significant Bit (MSB).

Most significant bit (the highest numbered bit). Bit to the far left.

3. What is the decimal representation of each of the following unsigned binary integers?

- a. 11111000
 - b. 11001010
 - c. 11110000
- (a) 248 (b) 202 (c) 240

4. What is the sum of each pair of binary integers?

- a. 00001111 " 00000010
 - b. 11010101 " 01101011
 - c. 00001111 " 00001111
- (a) 00010001 (b) 101000000 (c) 00011110

5. How many bytes are contained in each of the following data types?

- a. word
 - b. doubleword
 - c. quadword
- (a) 2 (b) 4 (c) 8

6. How many bits are contained in each of the following data types?

- a. word
 - b. doubleword
 - c. quadword
- (a) 16 (b) 32 (c) 64

❖ Section 2.1 Review

1-The central processor unit (CPU) contains registers and what other basic elements?

Control Unit, Arithmetic Logic Unit, and the clock.

2-The central processor unit is connected to the rest of the computer system using what three buses?

Data Bus, Control Bus, Address Bus.

3-Why does memory access take more machine cycles than register access?

Conventional memory is outside the CPU, and it responds more slowly to access requests. Registers are hard-wired inside the CPU.

4-What are the three basic steps in the instruction execution cycle?

Fetch, Decode, Execute.

5-Which two additional steps are required in the instruction execution cycle when a memory operand is used?

Fetch memory operands, store memory operands.

❖ Section 2.2 Review

1-What are the x86 processor's three basic modes of operation?
Protected Mode ,Real-address mode ,System management mode.

2- Name all eight 32-bit general-purpose registers.

EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP

3- Name all six segment registers

CS, DS, SS, ES, FS, GS

4- What special purpose does the ECX register serve?

Loop counter

❖ **Section 2.4 Review**

1. Describe SRAM and its most common use.

SRAM stands for Static Random Access Memory, used primarily for expensive, high speed cache memory.

It does not have to be refreshed.

2. Describe VRAM.

VRAM stands for Video Random Access Memory

it holds video data,

it is dual ported

> One port allow to continuously refresh the Display

> Other port writes data to display

3. List at least two features found in the Intel P965 Express chipset.

- Support for multiple USB ports, multiple PCI express slots, networking, and Intel Quiet System technology
- A high definition audio chip provides digital sound capabilities

4. Name four types of RAM

DRAM (Dynamic Read Only Memory), SRAM (Static Read Only Memory), VRAM (Video Read Only Memory), and CMOS RAM (Read Only Memory)

6. What is the purpose of the 8259A PIC controller?

The 8259A Programmable Interrupt Controller (PIC) handles external interrupts from hardware devices, such as the keyboard, system clock, and disk drives.

❖ Section 2.5 Review

1. Of the four levels of input/output in a computer system, which is the most universal and portable?

High level language functions or application program such as C++ or Java are the most universal and portable.

2. What characteristics distinguish BIOS-level input/output?

BIOS: The basic input–output system is a collection of low-level subroutines that communicate directly with hardware devices. The BIOS is installed by the computer’s manufacturer and is tailored to fit the computer’s hardware. Operating systems typically communicate with the BIOS

3. Why are device drivers necessary, given that the BIOS already has code that communicates with the computer’s hardware?

Small computer software that controls a particular hardware device connected to a computers operating system. Connection between operating system and hardware device communication.

Without the device driver, devices such as printers, webcam, CD-ROM cannot perform their respective functions.

4. In the example regarding displaying a string of chapters, which level exists between the operating system and the video controller card?

The BIOS level. It converts characters or maps the character into some particular type of font. Provides display onto the screen of computer.

5. Is it likely that the BIOS for a computer running MS-Windows would be different from that used by a computer running Linux?

No. The same BIOS would work for both operating systems. Many computer owners install two or three operating systems on

❖ Section 3.1 Review

1. Using the value -35, write it as an integer literal in decimal, hexadecimal, octal, and binary formats that are consistent with MASM syntax.

decimal: -35, -35d

hexadecimal: 0DDh

octal: 335o

binary: 11011101b

2. (Yes/No): Is A5h a valid hexadecimal literal?

NO.

3. (Yes/No): Does the multiplication operator (*) have a higher precedence than the division operator (/) in integer expressions?

No. Same precedence

4. Create a single integer expression that uses all the operators from Section 3.1.2. Calculate the value of the expression.

The following integer expression uses all the defined operators:

- 45 MOD 8 + (12 / 3) * 4

evaluation of the expression:

-45 MOD 8 + (12 / 3) * 4

-45 MOD 8 + (4) * 4

-45 MOD 8 + 4 * 4

-45 MOD 8 + 16

-5 + 16

Answer: 11

5. Write the real number -6.2×10^4 as a real number literal using MASM syntax.

-6.3E+04

6. (Yes/No): Must string literals be enclosed in single quotes?

NO

7. Reserved words can be instruction mnemonics, attributes, operators, predefined symbols, and _____.

Directives

8. What is the maximum length of an identifier?

2047

❖ Section 3.2 Review

1. In the AddTwo program, what is the meaning of the INCLUDE directive?

The INCLUDE directive copies necessary definitions and setup information from the text file. The data from this file is inserted into the data stream read by the assembler.

2. In the AddTwo program, what does the .CODE directive identify?

Marks the beginning of the code segment.

3. What are the names of the two segments in the AddTwo program?

.code, .data, .stack

4. In the AddTwo program, which register holds the sum?

Eax

5. In the AddTwo program, which statement halts the program?

INVOKE ExitProcess,0

❖ Section 3.3 Review

1. What types of files are produced by the assembler?

object file, or listing file

2. (True/False): The linker extracts assembled procedures from the link library and inserts them in the executable program.

True

3. (True/False): When a program's source code is modified, it must be assembled and linked again before it can be executed with the changes.

True

4. Which operating system component reads and executes programs?

The loader

5. What types of files are produced by the linker?

executable files

❖ Section 3.4 Review

1. Create an uninitialized data declaration for a 16-bit signed integer.

var1 SWORD ?

2. Create an uninitialized data declaration for an 8-bit unsigned integer.

var2 BYTE ?

3. Create an uninitialized data declaration for an 8-bit signed integer.

var3 SBYTE ?

4. Create an uninitialized data declaration for a 64-bit integer.

var4 QWORD ?

5. Which data type can hold a 32-bit signed integer?

SDWORD

❖ Section 3.5 Review

1. Declare a symbolic constant using the equal-sign directive that contains the ASCII code (08h) for the Backspace key.

Ans.: BACKSPACE = 08h

2. Declare a symbolic constant named SecondsInDay using the equal-sign directive and assign it an arithmetic expression that calculates the number of seconds in a 24-hour period.

Ans.: SecondsInDay = 24 * 60 * 60

3. Write a statement that causes the assembler to calculate the number of bytes in the following array, and assign the value to a symbolic constant named ArraySize:

myArray WORD 20 DUP(?)

Ans.: ArraySize = (\$ - myArray) 6

4. Show how to calculate the number of elements in the following array, and assign the value to a symbolic constant named ArraySize:

myArray DWORD 30 DUP(?)

Ans.: ArraySize = (\$ - myArray) / 4 OR ArraySize = (\$ - myArray) / TYPE DWORD
EQU Directive

5. Use TEXTEQU to create a symbol named Sample for a string constant, and then use the symbol when defining a string variable named MyString.

Ans.: Sample TEXTEQU <"This is a string">

❖ Section 4.1 Review

1. *What are the three basic types of operands?*

Register, immediate, and memory

2. (*False*): *The destination operand of a MOV instruction cannot be a segment register.*

3. (*False*): *In a MOV instruction, the second operand is known as the destination operand.*

4. (*True*): *The EIP register cannot be the destination operand of a MOV instruction.*

5. *In the operand notation used by Intel, what does reg/mem32 indicate?*

A 32-bit register or memory operand

6. *In the operand notation used by Intel, what does imm16 indicate?*

A 16-bit immediate (constant) operand

❖ Section 4.2 Review

Use the following data for the next several questions:

```
.data  
val1 BYTE 10h  
val2 WORD 8000h  
val3 DWORD 0FFFFh  
val4 WORD 7FFFh
```

1. Write an instruction that increments val2.

inc val2

2. Write an instruction that subtracts val3 from EAX.

sub eax,val3

3. Write instructions that subtract val4 from val2.

```
mov ax,val4  
sub val2,ax
```

4. If val2 is incremented by 1 using the ADD instruction, what will be the values of the Carry and Sign flags?

CF = 0, SF = 1

5. If val4 is incremented by 1 using the ADD instruction, what will be the values of the Overflow and Sign flags?

OF = 1, SF = 1

6. Where indicated, write down the values of the Carry, Sign, Zero, and Overflow flags after each instruction has executed:

```
mov ax,7FF0h  
add al,10h ; a. CF = 1 SF = 0 ZF = 1 OF = 0  
add ah,1 ; b. CF = 0 SF = 1 ZF = 0 OF = 1  
add ax,2 ; c. CF = 0 SF = 0 ZF = 0 OF = 0
```

7. Implement the following expression in assembly language: $AX = (-val2 + BX) - val4$.

```
mov ax, val2  
neg ax  
add ax, bx  
sub ax, val4
```

8. (No): Is it possible to set the Overflow flag if you add a positive integer to a negative integer?

9. (Yes): Will the Overflow flag be set if you add a negative integer to a negative integer and produce a positive result?

10. (Yes): Is it possible for the NEG instruction to set the Overflow flag?

11. (No): Is it possible for both the Sign and Zero flags to be set at the same time?

12. Write a sequence of two instructions that set both the Carry and Overflow flags at the same time.

```
mov al, 80h  
add al, 80h
```

❖ Section 4.3 Review

1. (True/False): *The OFFSET operator always returns a 16-bit value.*

False

2. (True/False): *The PTR operator returns the 32-bit address of a variable.*

False

3. (True/False): *The TYPE operator returns a value of 4 for doubleword operands.*

True

4. (True/False): *The LENGTHOF operator returns the number of bytes in an operand.*

False

5. (True/False): *The SIZEOF operator returns the number of bytes in an operand.*

True

❖ Section 4.4 Review

1. (*False*): *Any 16-bit general-purpose register can be used as an indirect operand.*
2. (*True*): *Any 32-bit general-purpose register can be used as an indirect operand.*
3. (*False*): *The BX register is usually reserved for addressing the stack.*
5. (*True*): *The following instruction is invalid: inc [esi]*
6. (*True*): *The following is an indexed operand: array[esi]*

❖ Section 4.5 Review

1. (*True*): A JMP instruction can only jump to a label inside the current procedure.
2. (*False*): JMP is a conditional transfer instruction.
3. If ECX is initialized to zero before beginning a loop, how many times will the LOOP instruction repeat? (Assume ECX is not modified by any other instructions inside the loop.)
4,294,967,296 times
4. (*False*): The LOOP instruction first checks to see whether ECX is not equal to zero; then LOOP decrements ECX and jumps to the destination label.
5. (*True*): The LOOP instruction does the following: It decrements ECX; then, if ECX is not equal to zero, LOOP jumps to the destination label.
6. In real-address mode, which register is used as the counter by the LOOP instruction?
CX
8. (*False*): The target of a LOOP instruction must be within 256 bytes of the current location.