

CMPS-2240 Assembly Language Programming

Homework Set 1

You are free to use any resources at your disposal to answer the homework. However, directly copying answers from online sources is plagiarism. You must show your work for problems 6-9. Unsupported answers will be given a zero grade.

1. What is the difference between a high level language and a low level language?

High Level languages are easier to write, read, edit and understand.
Low level languages are more dependent on the platform they target.

2. What is a mnemonic?

It's a learning technique. It's an aid to help us remember low level languages.

3. A program is translated to machine code for a computer with an Intel x86 microprocessor. Will this code run on a machine with a MIPS microprocessor?

Explain your answer.

No, machine language is platform dependent.

4. What is the difference between reduced instruction set computing and complex instruction set computing?

Reduced Instruction Set typically use commands that can be completed within one clock cycle. Complex Instructions Sets try to accomplish as much as possible within a single instruction.

5. A microprocessor is designed to receive specific instructions, and the set of these instructions are referred to as an Instruction Set Architecture (ISA). One such architecture is MIPS. Find 3 other ISAs and describe them.

ARM, x86, x86-64, PowerPC

6. Convert the following numbers to binary:

- A. 128: 10000000
- B. 399: 110001111
- C. 271: 100001111

7. Convert the following numbers to hexadecimal:

- A. 10: A
- B. 233: E9
- C. 199: C7

8. Convert the following numbers to decimal:

- A. 1001 1001: 153
- B. 1010 0000: 160
- C. 0xFF: 255

9. Perform these binary operations. Then convert the binary results into decimal.

1011	1110	0111 0111	1100
+ 1110	- 1011	+ 0000 1011	- 0101
----	----	-----	----

1011	1110	0111 0111	1100
+ 1110	- 1011	+ 0000 1011	- 0101
-----	-----	-----	-----
11001	0011	1000 0010	111
check: 11+14=25	14-11=3	119+11=130	12-5=7

10. Compute these bitwise logical operations.

This is C syntax: '&' is bitwise AND; '|' is bitwise OR; '^' is bitwise XOR; '~' is NOT.

11100	11010	11001	~1	~0
& 10101	10001	^ 10010	--	--
-----	-----	-----		

11100	11010	11001	~1	~0
& 10101	10001	^ 10010	--	--
-----	-----	-----	0	1
10100	11011	01011		