**SOME SAMPLE QUESTIONS**

**Note:** The purpose of this document is to give you an idea of the questions on the quiz.

**Assembly Programming Introduction**

**1.** Assume the following values are stored at the indicated memory addresses and registers:

Address Value Register Value

0x100 0xFF %rax 0x100

0x104 0xAB %rcx 0x1

0x108 0x13 %rdx 0x3

0x10C 0x11

Fill in the following table showing the values for the indicated operands:

Operand Value Answer Comment

%rax \_\_\_\_\_\_\_\_\_\_ 0x100 Register

0x104 \_\_\_\_\_\_\_\_\_\_ 0x AB Absolute address

$0x108 \_\_\_\_\_\_\_\_\_\_ 0x108 Immediate

(%rax) \_\_\_\_\_\_\_\_\_\_ 0xFF Address 0x100

4(%rax) \_\_\_\_\_\_\_\_\_\_ 0xAB Address 0x104

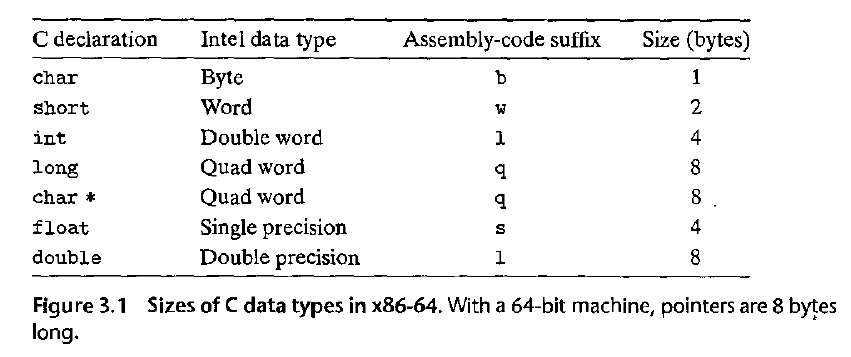
9(%rax,%rdx) \_\_\_\_\_\_\_\_\_\_ 0x11 Address 0x10C

260(%rcx,%rdx) \_\_\_\_\_\_\_\_\_\_ 0x13 Address 0x108

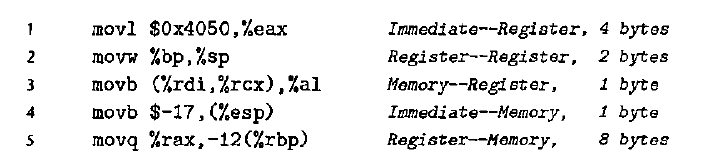
0xFC( ,%rcx,4) \_\_\_\_\_\_\_\_\_\_ 0xFF Address 0x100

(%rax,%rdx,4) \_\_\_\_\_\_\_\_\_\_ 0x11 Address 0x10C

Command Suffix (Optional feature):



Example MOV instruction:



**2.** For each of the following lines of assembly language, determine the appropriate instruction suffix based on the operands. (For example, **mov** can be rewritten as **movb, movw, movl, or movq**.)

mov\_ %eax, (%rsp) l

mov\_ (%rax), %dx w

mov\_ $0xFF, %bl b

mov\_ (%rsp,%rdx,4), %dl b

mov\_ (%rdx), %rax q

mov\_ %dx, (%rax) w

**3.** Each of the following lines of code generates an error message when we invoke the assembler. Explain what is wrong with each line. (Assume 64 bit machine)

movb $0xF, (%ebx) Cannot use %ebx as address register (addresses are 8 bytes long)

movl %rax, (%rsp) Mismatch between instruction suffix and register ID

movw (%rax),4(%rsp) Cannot have both source and destination be memory references

movb %al, %sl No register named %sl

movq %rax, $0x123 Cannot have immediate as destination

movl %eax, %rdx Destination operand incorrect size

movb %si, 8(%rbp) Mismatch between instruction suffix and register ID

**4.** LEA Command Examples:

Suppose register %rax holds value x and %rcx holds value y. Fill in the table below with formulas indicating the value that will be stored in register %rdx for each of the given assembly-code instructions:

leaq 6 (%rax) , %rdx 6+x

leaq (%rax, %rcx) , %rdx x + y

leaq (%rax, %rcx,4), %rdx x +4y

leaq 7 (%rax, %rax, 8) , %rdx 7 + 9x

leaq 0xA(,%rcx,4), %rdx 10 + 4y

leaq 9(%rax,%rcx,2), %rdx 9 + x + 2y

**5.** *Find the C expression t:*

long scale(long x, long y, long z) {

long t = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

return;

*When*

long scale(long x, long y, long z)

x in %rdi, y in %rsi, z in %rdx

scale:

leaq (%rdi,%rsi,4), %rax x + 4\*y

leaq (%rdx,%rdx,2), %rdx z + 2\*z = 3\*z

leaq (%rax,%rdx,4), %rdx (x+4\*y) + 4\*(3\*z) = x + 4\*Y + 12\*Z

ret

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*