

CMSC 216 Quiz 3 Worksheet

The next quiz for the course will be on Tue, Jul 29. The following list provides additional information about the quiz:

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>
- Make sure you know your section number and your TA's name.

We will include the following cheat sheet in the quiz so you don't have to memorize some Assembly constructs.

Assembly Cheat Sheet

Registers → %eax, %ecx, %edx, %ebx, %esi, %edi

Assembler Directives → .align, .long, halt, .pos

Data movement → irmovl, rrmovl, rmmovl, mrmovl

Integer instructions → addl, subl, multl, divl, modl

Branch instructions → jmp, jle, jl, je, jne, jge, jg

Reading/Writing instructions → rdch, rdint, wrch, writ

Ascii code for newline character → 0x0a

Ascii code for space → 0x20

End of Assembly Cheat Sheet

Solutions to the exercises below will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

NOTE: The quiz will not ask you to write code for functions that use parameters or local variables; however you need to know how to draw the stack/values of registers for code that relies on parameters and local variables.

1. What is the difference between Big Endian and Little Endian? Suppose we have the value 0x02143657. How would the value be represented using Big/Little Endian?
2. What is the purpose of the .align directive?
3. What is the purpose of the .pos directive?
4. Is it possible to have an Assembly (Y86) program with an array of 5000 elements? Briefly explain.
5. What is the difference between **irmovl MyData, %eax** and **mrmovl MyData, %eax**, assuming MyData is a label?
6. Write Assembly code that will define a global variable named x that has some initial value (e.g., 100). The program will read an integer and will print the result of dividing that value by x.
7. Write an Assembly program that reads an uppercase letter and prints the corresponding lowercase. To compute the lowercase just add 32 to the ascii value of the uppercase. Make sure a \n is printed after the lowercase letter.
8. Write Assembly code that will read two integer values and print 0 if the first value is divisible by the second, and any other number otherwise.

9. Which two registers define the frame of a function?
10. How can you debug Assembly code?
11. Draw the stack and the values of the registers for the following Assembly program when execution reaches for the second time the instruction that has the comment **TRACE STOP**. You can assume the user provided 3 as input.

```

main:      irmovl $0x1000, %esp # init stack ptr
          rdint %ebx
          pushl %ebx           # pushing parameter
          call fact
          popl %ebx            # remove parameter
          writ %eax            # accessing the return value
          irmovl $10, %ecx     # newline
          wrch %ecx
          halt

fact:      pushl %ebp           # ON_ENTRY saving old frame ptr
          rrmovl %esp, %ebp     # ON_ENTRY set new frame ptr

          mrmovl 8(%ebp), %edx  # Retrieving parameter
          irmovl $1, %ecx
          subl %ecx, %edx       # Checking whether it is one
          je baseCase          # if equal 1 we are done
recStep:   pushl %edx           # pushing n - 1 parameter
          call fact            # recursive call      TRACE STOP
          popl %edx            # getting rid of passed parameter
          mrmovl 8(%ebp), %edx  # accesing parameter n
          multl %edx, %eax      # %eax has result of previous call
          jmp factEnd

baseCase:  irmovl $1, %eax

factEnd:   rrmovl %ebp, %esp    # ON_EXIT reset stack ptr
          popl %ebp            # ON_EXIT restore old frame
          ret                  # ON_EXIT

```