CS 3843 Spring 2011 Exam 2

Answer the following questions in the space provided. You may use the back of the test as a scratch area. Write your name on this question paper.

True or False
Write T for True or F for False in the spaces provided.

1. The adcl instruction adds three quantities.
2. The %esp register points to the current stack frame.
3. Arguments to compiled C functions are passed in registers.
4. Assigning a value through a pointer can prevent the compiler from performing certain optimizations.
5. The gets() function is awesome and should be used at every opportunity.
6. Under Microsoft Windows, a struct containing a double is aligned on an address divisible by 8.
7. Under x86 Linux, a struct containing a double is aligned on a memory address divisible by 8.
8. Heap-allocated objects can be placed in more than one possible segment of memory.
9. Stack-allocated objects can be placed in more than one possible segment of memory.
10. A union containing an array of 20 ints would be four bytes long.
Addressing Modes

11. Write the final value of `%ebx` for each of the following code snippets. Show your work for partial credit on an incorrect answer. Recall that `shrl` is logical shift right and `sall` is logical shift left.

1. \begin{verbatim}
   movl $3, %ebx
   leal (%ebx,%ebx,2), %ebx
\end{verbatim}

2. \begin{verbatim}
   movl $0x123, %ebx
   shrl $8, %ebx
   leal 1(%ebx,%ebx,4), %ebx
\end{verbatim}

3. \begin{verbatim}
   xorl %ebx,%ebx
   leal 1(%ebx), %ebx
   sall $3, %ebx
   leal 2(%ebx,%ebx,8), %ebx
\end{verbatim}
Reading Comprehension

12. Consider the following assembly language program:

```assembly
f:
    movl 4(%esp),%eax
    andl $1,%eax
    ret

.globl main
main:
    subl $20,%esp
    pushl %esi
    pushl %ebx
    xorl %esi,%esi
    xorl %ebx,%ebx
.L22:
    addl $-12,%esp
    pushl %ebx
    call f
    addl %eax,%esi
    addl $16,%esp
    incl %ebx
    cmpl $999,%ebx
    jle .L22
    addl $-8,%esp
    pushl %esi
    pushl $.LC0
    call printf
    addl $-12,%esp
    pushl $0
    call exit
.LC0:
    .string "%d\n"
```

1. What is the output of this program?
12. (continued)

2. Describe, in 25 words or fewer, what this program does.

3. Write a C program that could plausibly be compiled to result in this assembly language program.
Alignment

13. Consider the following definition of an array A in C:

```c
struct {
    unsigned int a;
    char b;
    double c;
    char d;
    union {
        char x[8];
        unsigned char y[8];
        double z;
    }
    e;
    char f;
} A[3];
```

1. What is the size of A in bytes on x86 Linux?

2. What is the size of A in bytes on Microsoft Windows? (Hint: it’s different.)

3. Rewrite the definition of A so that it contains the same fields with the same types but takes as few bytes as possible under x86 Linux.

4. How many bytes does your new A consume under x86 Linux?