CSCI-UA.0201-001/2

Computer Systems Organization

Midterm Exam Spring 2015

(60 minutes)

Last name: First name:

Notes:

- **If you perceive any ambiguity in any of the questions, state your assumptions clearly**
- **Questions vary in difficulty; it is strongly recommended that you do not spend too much time on any one question.**
- **This exam is open book/notes.**

---

1. (5 points) Circle the correct answer among the choices given. If you circle more than one answer, you will lose the grade of the corresponding question.

(A) If you distribute your program among 10 C files, the compiler will generate:

(1.) 10 assembly files 2. 10 object code files
3. 1 big assembly file 4. 1 big object code file

(B) If there is an array declaration: int x[10], then x[0] will be stored at a memory address lower than x[5] (i.e. if x[0] is in memory address 100 for example, then x[5] must be after 100)

(1.) True 2. False 3. correct only for big endian machines
4.correct only for little endian machine

(C) Presenting -7 in signed integer or in IEEE 754 yields the same bit pattern.

1. The above statement is true
(2.) The above statement is false
3. It depends on whether the machine is 32-bit or 64-bit
4. It depends on whether the machine is big-endian or little endian.

(D) Suppose x = 0x0023 and y = 0x2300, which of the following conditions is evaluated to true if used in an if-statement?

(1.) (x && y) 2. (x & y) 3. both of them 4. none of them

(E) The following number: 00011101 can be interpreted as:

1. unsigned number 2. signed number (3.) both 4. none

2. (5 points) List the bugs in the following code. No need to correct them.

```
#define  NUM   100
void  adjust_data(){

  int * x;

  x = malloc(NUM * sizeof(int));

  x[0] = 40000;
  for(i = 1; i <=  NUM;  i++)
      x[i] = x[i-1]++;

  return x[0];

}
```

- The function returns an integer, so cannot be "void".
- Typecasting of (int *) must be used with malloc.
- "i" must be declared before used.
- The for-loop must end before i reaches NUM, otherwise will be out of range.

3. (2 points) What is the difference between macros and function calls in term of: resulting code size and speed?

- Size: macro will make the code bigger in size, because each occurrence of the macro name will be substituted by the macro code. This is not the case with a function.
- Speed: Calling a function is slower than a macro, because a function requires setting the stack, etc.

4. Assume we have the following array: int x[4];

We want to access x[i]. Assume the array starts at address 0xA0080000.

a. (2 points)Write a single IA-32 assembly instruction to implement: y = x[i]; Assume i is already stored in eax and y is stored in ebx.

movl 0xA0080000(,%eax,4), %ebx

b. (2 points) What is the starting address of x[3] if the machine is big-endian?

x[3] is the last element of the array. It starts at:
0xA0080000 + (4*3) = 0xA0080000 + 0xC = 0xA008000C

5. (2 points) Why don't we have a *movl src, dest* instruction with both src and destination are memory addresses?

To move an element from memory to memory the processor needs to do two things:

1. Bring that element from memory into register.
2. Move that element from the register to the new memory location.

The ISA already has instructions to do each one of the above steps.

6. (2 points) State two advantages for moving from 32-bit machines to 64-bit machines.

- Larger memory address space can be accessed.
- Wider buses, which enables faster data movement to/from memory and processor.