Name: CS520

Midterm Exam

18 Oct 2016, 80 minutes, 20 questions, 100 points

> The exam is closed book and notes.
>
> Please keep all electronic devices turned off and out of reach.
>
> Note that a question may require *multiple* checked boxes for a correct answer. Checking *some* but not *all* of the required boxes will result in a *partial* answer worth only 2 of the 5 points. Checking any box that shouldn't be checked results in an *incorrect* answer, worth zero.

1. How would the following C string (i.e. null terminated) `"bead"` be represented using ASCII in the memory of a Big Endian machine with a byte-addressable memory? The bytes are shown below in increasing memory address order, left to right. The ASCII code for `'b'` is `0x62`. The ASCII code for `'e'` is `0x65`. The ASCII code for `'a'` is `0x61`. The ASCII code for `'d'` is `0x64`. [5 pts]

   ○ `0x62 0x65 0x61 0x64`.

   ○ `0x64 0x61 0x65 0x62`.

   ○ `0x00 0x64 0x61 0x65 0x62`.

   √ `0x62 0x65 0x61 0x64 0x00`.

   ○ `0x64 0x61 0x65 0x62 0x00`.

   ○ none of the above.

2. The algorithm for adding two's complement integer values is: [5 pts]

   √ add the two values as if unsigned and ignore any carry out of the leftmost bit position.

   ○ add the two values as if unsigned and if there is a carry out of the leftmost bit position then add 1 to the rightmost bit position.

   ○ test the sign bits of the values first to determine whether to do an addition or a subtraction.

   ○ none of the above.

3. Consider the following C function: [5 pts]

```
unsigned int f(void)
{
  int i = 1;
  return *(unsigned char *) &i;
}
```

   On a machine with a byte-addressable memory, the function will:

   ○ always return 0.

   ○ always return 1.

   ○ return 0 if the machine is little-endian and 1 otherwise.

   √ return 1 if the machine is little-endian and 0 otherwise.

   ○ none of the above.

4. Which of the following statements about two's complement integer values are true? [5 pts]
   - ◯ There are two zero values, +0 and -0.
   - √ There is one more negative number than positive number.
   - √ The negation of a value can cause an overflow.
   - √ The leftmost bit indicates the sign of the number.

5. What would -39 look like as a 32-bit two's complement integer in the memory of big-endian machine? The bytes are shown below in increasing memory address order, left to right. [5 pts]
   - ◯ `0xD9 0xFF 0xFF 0xFF`.
   - ◯ `0x80 0x00 0x00 0x27`.
   - √ `0xFF 0xFF 0xFF 0xD9`.
   - ◯ `0xFF 0xFF 0xFF 0xD8`.
   - ◯ none of the above.

6. Which of the following statements about IEEE single-precision floating-point are true? [5 pts]
   - √ There are two zero values, +0.0 and -0.0.
   - √ It stores the exponent in biased form.
   - √ It allows some denormalized values to be stored.
   - ◯ It has one more negative value than positive value.

7. Interpret `0x7F800000` as IEEE single-precision floating-point. What is its decimal value? [5 pts]
   - ◯ `12.75`.
   - ◯ NaN.
   - ◯ `25.5`.
   - ◯ `51.0`.
   - √ positive infinity.
   - ◯ none of the above.

8. Interpret `0x017FFFFF` as IEEE single-precision floating-point. Which of the following are accurate descriptions of the value? [5 pts]
   - √ represents a positive value.
   - ◯ represents a negative value.
   - ◯ represents a zero value.
   - ◯ represents an infinity value.
   - ◯ represents a NaN value.
   - ◯ represents a denormalized value.

9. Interpret `0x40A00000` and `0x40A00000` as IEEE single-precision floating-point. Add the two values together. What is the result? [5 pts]

- ◯ `0x00000000`.
- √ `0x41200000`.
- ◯ `0x31200000`.
- ◯ `0x41300000`.
- ◯ `0x42100000`.
- ◯ none of the above.

10. Interpret `0x03FFFFFD` as a 32-bit two's complement value. Convert it to IEEE single-precision floating point. What is the result? [5 pts]

- √ `0x4C7FFFFF`.
- ◯ `0x4C800000`.
- ◯ `0x4C000000`.
- ◯ `0x7F800000`.
- ◯ none of the above.

11. Represent this UTF-16 value, `0x0011`, in UTF-8. What is the result? [5 pts]

- √ `0x11`.
- ◯ `0xC0 0x91`.
- ◯ `0xE0 0x80 0x91`.
- ◯ `0xF0 0x80 0x80 0x91`.
- ◯ none of the above.

12. Represent this UTF-32 value, `0x00010000`, in UTF-16. What is the result? [5 pts]

- ◯ `0x10000`.
- ◯ `0x0001 0x0000`.
- √ `0xD800 0xDC00`.
- ◯ `0xDC00 0xD800`.
- ◯ none of the above.

13. Which of the following statements are true? [5 pts]

- √ The UTF-8 encoding of an ASCII character is always one byte long.
- ◯ The UTF-8 encoding of a Unicode character is always shorter than the UTF-16 encoding of the character.
- ◯ The UTF-8 encoding of a Unicode character is always shorter than the UTF-32 encoding of the character.
- ◯ The UTF-16 encoding of a Unicode character is always shorter than the UTF-32 encoding of the character.

14. The Byte-Order Mark (BOM) is encoded as 0xFEFF. Which of the following statements are true? [5 pts]

- ◯ A file containing UTF-32 characters that starts with `0x00 0x00 0xFE 0xFF` is in little-endian format.
- √ A file containing UTF-32 characters that starts with `0xFF 0xFE 0x00 0x00` is in little-endian format.
- ◯ A BOM at the beginning of the file is not enough to determine the endian-ness of the file. You also need to know the endian-ness of the machine that is used to read the file.
- ◯ A BOM at the beginning of the file is not enough to determine the endian-ness of the file. You also need to know the endian-ness of the machine that was used to write the file.

15. What is the encoding of this maTe instruction: `newint 11`. The words are shown below in increasing memory address order, left to right. [5 pts]

   ◯ `0x0000000B 0x000000F7`.

   ◯ `0x000000F7 0x00000001 0x00000001`.

   √ `0x000000F7 0x0000000B`.

   ◯ `0x000000F7 0x0000000B 0x00000000`.

   ◯ none of the above.

16. Which of the following statements about maTe instructions are true? [5 pts]

   √ Many instructions operate upon implicit operands (e.g. on top of the stack).

   ◯ All encoded instructions have the same length.

   √ All encoded instructions begin with a word containing the encoding of the opcode.

   ◯ All encoded instructions end with a word containing zero.

17. Which of the following statements about maTe frames are true? [5 pts]

   √ Contains storage for local variables.

   √ Contains a return address.

   √ Contains an operand stack.

   ◯ Contains the encoded instructions for the frame's method.

18. Which of the following statements about the indirection array in our implementation of the maTe VM are true? [5 pts]

   √ An index into the indirection array serves as the Reference for the object pointed to by that index's entry in the array.

   ◯ The indirection array stored only pointers to string literal values.

   √ We needed the indirection array because agate is a 64-bit machine and the maTe VM is a 32-bit machine.

19. When a maTe instruction is executing (after all components of the encoded instruction have been read), the pc points to: [5 pts]

   ◯ The instruction that is executing.

   √ The next instruction in memory after the instruction that is executing.

   ◯ The next method that should be invoked.

   ◯ The method that should be returned to.

   ◯ none of the above.

20. Which of the following statements are true? [5 pts]

   √ An assembler has two passes because the definition of a label may come after its use.

   √ The first pass of an assembler determines the address of each label defined in the program.

   √ The second pass of an assembler is necessary to finish encoding the instructions.

   √ The assembler uses a symbol table to store labels and their addresses.