Name: CS520

Final Exam

13 December 2016, 120 minutes, 26 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 4 points. Checking any box that shouldn't be checked results in an *incorrect* answer, worth zero.

1. [0 pts]
   - ◯ Return my exam to my Kingsbury mailbox.
   - ◯ Hold my exam in your office. I will pick it up prior to February 15.
   - √ Shred my exam. I never want to see it again.

2. Interpret `0x8079A56F` (shown in hexadecimal) as an IEEE single-precision floating-point value. Which of the following statements about this value are true? [4 pts]
   - √ It is negative.
   - ◯ Its stored exponent is `0x16`.
   - ◯ Its actual exponent is `-22` (in decimal).
   - √ It is a denormalized value.

3. Concerning the producer-consumer queue assignment, which of the following statements are true: [4 pts]
   - ◯ the producers never wait.
   - ◯ the consumers never wait.
   - √ unless there is an error, the producers always signal.
   - √ the consumers always signal.

4. Decode the following UTF-8 sequence (shown as hexadecimal) to a Unicode character: `E0 B2 B2`. The answer represented in UTF-32 in hexadecimal is: [4 pts]
   - ◯ `0x0000B2B2`.
   - ◯ `0x0001B2B2`.
   - √ `0x00000CB2`.
   - ◯ `0x0000CBB1`.
   - ◯ none of the above.

5. Consider the following Intel-64 instruction expressed in Linux assembly language: `movq (%rax),%rbx`. What does this instruction do? [4 pts]
   - ◯ move the quadword on top of the stack to RBX.
   - √ move the quadword pointed to by RAX to RBX.
   - ◯ move the quadword in RAX to the quadword pointed to by RBX.
   - ◯ move the quadword pointed to by RBX to the quadword in RAX.
   - ◯ none of the above.

6. Consider the following C function: [4 pts]

```
unsigned int f(void)
{
  float x = 0;
  return *(unsigned char *) &x;
}
```

On a machine with a byte-addressable memory, and where a C float is implemented as an IEEE single-precision floating-point value, the function will:

- ◯ return 0 if the machine is little-endian and 1 otherwise.
- ◯ return 1 if the machine is little-endian and 0 otherwise.
- √ always return 0.
- ◯ none of the above.

7. Which of the following statements about the maTe virtual machine are true: [4 pts]

- √ local variables are stored in frames.
- ◯ the intermediate results of computations are stored in registers.
- √ it is a stored-program machine.
- ◯ it does not have a run-time stack.

8. Add the following two IEEE single-precision floating-point values shown in hexadecimal: `0xC0000000` and `0xC0000000`. The result in hexadecimal is: [4 pts]

- ◯ `0xC0C00000`.
- ◯ `0x80800000`.
- ◯ `0x00000000`.
- √ `0xC0800000`.
- ◯ none of the above.

9. Which of the following statements about maTe instructions are true: [4 pts]

- ◯ all encoded instructions are the same length.
- √ all encoded instructions start with the opcode.
- ◯ maTe class files contain **only** encoded instructions.
- √ the program counter (PC) register points to the next instruction to be executed.

10. In an implementation of an exception mechanism for C programs, the **catchException** function: [4 pts]

- √ pushs the current RBP, the saved RBP and the saved RIP onto the "snapshot" stack.
- ◯ clears the "catch" bit of the RAX register.
- ◯ pops the "snapshot" stack.
- ◯ none of the above.

11. A memory cache with one line per set is known as a: [4 pts]

- √ direct-mapped cache.
- ◯ set-associative cache.
- ◯ fully-associative cache.
- ◯ translation lookaside buffer.
- ◯ none of the above.

12. The purpose of the mark phase of a mark-and-sweep garbage collector is to: [4 pts]

- ◯ gather all the garbage blocks at one end of the heap.
- ◯ gather all the non-garbage blocks at one end of the heap.
- √ identify all the reachable, allocated blocks.
- ◯ clear the mark bit on all blocks.
- ◯ none of the above.

13. Which of the following statements about Von Neumann's design for the IAS computer are true? [4 pts]

- ◯ It supported floating-point numbers.
- √ It was a stored-program computer.
- √ It had a PC register.
- ◯ It had a byte-addressable memory.

14. Consider how the following two C loops would be accessed by a memory system with 256 words and a direct-mapped cache with 16 sets and a block size of 4 words. [4 pts]

```
for (i = 0; i < 64; i++)
  a[i] = i;
sum = 0;
for (i = 0; i < 64; i++)
  sum += a[i];
```

Assume only the array is in the cache and the first word of the array is at address zero. If the cache is initially empty, how many cache hits would there be for writes and how many cache hits would there be for reads?

- ◯ 0 read hits and 0 write hits.
- √ 64 read hits and 48 write hits.
- ◯ 64 read hits and 64 write hits.
- ◯ 48 read hits and 48 write hits.
- ◯ none of the above.

15. Consider how -79 (base 10) would be represented in the memory of a Little Endian machine as a 16-bit 2's complement integer. The two bytes, shown left to right in increasing memory address order, would be: [4 pts]

- ◯ `0xFF 0xB0`.
- ◯ `0xF1 0x1F`
- ◯ `0xFF 0x22`.
- ◯ `0x80 0x4F`.
- √ none of the above.

16. Consider this C declaration: [4 pts]

```
int (*f)(long);
```

This declares a:

- √ pointer to a function that takes a long, and returns an int.
- ◯ function that takes a long, and returns a pointer to an int.
- ◯ pointer to a function that takes an int, and returns a long.
- ◯ none of the above.

17. The UTF-16 sequence (shown in hexadecimal) `0xDAAA 0xDD33` is represented in UTF-32 (in hexadecimal) as: [4 pts]

- ◯ two Unicode characters, `0x0000DAAA` and `0x0000DD33`.
- ◯ two Unicode characters, `0x0000000B` and `0x0000A933`.
- ◯ a single Unicode character, `0x000AA933`.
- √ a single Unicode character, `0x000BA933`.
- ◯ none of the above.

18. We used a POSIX condition variable to implement a producer-consumer queue in order to: [4 pts]

- √ block a producer thread that needs to wait for an available slot in the queue.
- ◯ track the length of the queue.
- ◯ protect against concurrent access to the struct maintaining the state of the queue.
- ◯ none of the above.

19. In our implementation of a garbage collector for the maTe VM, we used an indirection array because: [4 pts]

- ◯ maTe supports a load indirect instruction.
- √ maTe has 32-bit addresses and agate has 64-bit addresses.
- ◯ the mark phase of the garbage collector processed the roots of the garbage collection indirectly.
- ◯ maTe class files stores the class table indirectly.
- ◯ none of the above.

20. Which of the following are examples of a program exhibiting temporal locality: [4 pts]

- ◯ iterating through all members of an array in order.
- √ repeatedly incrementing a loop counter variable inside a loop.
- ◯ a sequence of instructions being executed in order without any branch or call instructions.
- ◯ putting the function return address in the RAX register.

21. Add together the following two 8-bit 2's complement integers (shown in hexadecimal): `0x01` and `0xFE`. Which of the following are true statements about the result? [4 pts]

- √ The result (in hexadecimal) is `0xFF`.
- √ The result is negative.
- ◯ The result overflows.
- √ The result in decimal is -1.
- ◯ The result in decimal is -127.

22. Which of the following statements about calling `pthread_cond_signal` are true? [4 pts]

- ◯ If there are no waiters on the condition variable, the signal is remembered and the next thread that calls `pthread_cond_wait` will not need to wait.
- ◯ A thread waiting on the condition variable starts executing immediately.
- √ The waiting thread that is signalled must next wait to re-acquire the mutex associated with the condition variable.
- ◯ All threads waiting on the condition variable are removed from the wait queue for the condition variable.

23. Which of the following statements are true about maTe class files: [4 pts]

- ◯ They are human-readable (contain only ASCII characters).
- √ When they are executed, execution begins at the beginning of the "main block."
- √ They contain a class table.
- ◯ They are always the same length.

24. Which of the following statements are true? [4 pts]

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

25. In Program 6, the `lock` function: [4 pts]

- √ empties the specified cache, first writing any modified words to main memory.
- ◯ loads all words in memory into the cache.
- ◯ sets all dirty-bits in all lines in the cache to 1, indicating that all words in the cache have been modified since they were loaded into the cache.
- ◯ none of the above.

26. In Intel 64 stack frames, the currently executing function's frame: [4 pts]

- ◯ contains the code for the function.
- √ is the top frame on the stack.
- ◯ is always 128 bytes long.
- ◯ none of the above.