ELEE 390 B – Developing Concurrent Software – Final Exam

Please answer each question as completely as possible.

**Name:**

1. What is the difference between a process and a thread?

**Threads share memory by default, processes do not. (Threads lecture, slide 3)**

2. If a std::condition\_variable is signaled, but the predicate passed to the wait method returns false, will the mutex be locked when the wait method exits?

**Yes (Asynchronous operations lecture, slide 9)**

3. What must you do to get the value of the std::future<T> returned by a call to std::async?

**A call to get waits until the future is ready, then returns the value of type T** **(Asynchronous operations lecture, slide 14)**

4. Name one of the two relationships that an atomic variable can be used to specify. Briefly explain it.

* ***synchronizes-with***
  + **Occurs only with atomics**
  + **Causes a load and a store to work together**
  + **An atomic write (store) synchronizes-with an atomic read(load) on the same memory location**
* ***happens-before***
  + **Not specific to atomics or threads**
  + **If A is sequenced B, A happens-before B**
  + **If A happens-before B, and B is on another thread, then we have an *inter-thread happens-before***
  + **A happens-before B and B happens-before C implies A happens-before C (transitive property)**

**(Atomics lecture, slide 11)**

5. Why do we need to use a std::atomic<bool> instead of a bool to signal a background thread to exit?

**A bool does not allow us to tell the compiler about the inter-thread happens-before relationship. (Atomics lecture, slide 14)**

6. Name both of the two reasons we design concurrent software.

* + **Maintain responsiveness during long operations**
  + **Divide work to accomplish more in less time**

**(Designing concurrent software lecture, slide 2)**

7. What is the value of *P* in Amdahl’s Law when *fs* = 1?

**P = 1, each extra processor is used at zero efficiency (Designing concurrent software lecture, slide 16)**

8. Which of the three approaches to dividing work does task-based concurrency employ?

**The best way to take advantage of Amdahl’s Law is to organize an algorithm to partition its input data. (Task-based concurrency lecture, slide 2)**

9. Describe the difference between performance and scalability.

* + ***Performance*: the time an action takes to complete**
  + ***Scalability*: how the performance changes with more resources or a larger problem**

**(Designing concurrent software lecture, slide 2)**

10. Name one of the three ways communication between processes is implemented in MPI.

* + - **Shared memory**
    - **TCP/IP**
    - **Infiniband**

**(Introduction to MPI lecture, slide 5)**

11. What technique can we use to write unit tests for code that uses MPI?

**Use dependency injection to isolate the MPI interface calls (Introduction to MPI lecture, slide 16)**

12. Describe why we are seeing more special purpose hardware (specifically GPUs) being developed?

**Recall the end of Dennard scaling.**

* + GPUs exhibit *worse* performance.
  + GPUs exhibit *better* performance per watt.

**(Hybrid concurrency lecture, slide 20)**

13. What special mode of execution does the Xeon Phi support that other GPUs do not support?

**Native compilation** **(Hybrid concurrency lecture, slide 13)**

14. Given an atomic<int> used to maintain a reference count, how can we get inter-thread happens-before on two statements (a decrement and a check for zero of the reference count)?

**Use the call stack, which is local to each thread. (Atomics and object lifetime lecture, slide 9)**

15. What is a data race, as defined by the C++ standard?

**C++ defines a data race as a problematic race condition, which cases undefined behavior (Mutexes lecture, slide 3)**

16. Name one of the three rules for avoiding deadlocks.

* + **Avoid nested locks – do not acquire another lock if you already have one**
  + **Avoid calling user-supplied code while holding a lock**
  + **Acquire locks in a fixed order – always (a lock hierarchy can help)**

**(Mutexes lecture, slide 9)**

17. Name one of the four callable types a thread can execute.

* + **Free function**
  + **Functor (function object)**
  + **Member function**
  + **Lambda expression**

**(Threads lecture, slide 9)**

18. In what two places has power become the leading optimization factor for transistor design?

* + **The data center – due to large scale**
  + **Mobile devices – due to batteries**

**(Dennard scaling lecture, slide 10)**

19. What is “the free lunch” and why is it over?

**“The free lunch” is the process by which hardware manufacturers produce processors that improve the single-thread performance of programs, without any change to the programs. It is over because clock speeds are not increasing nearly has fast as they used to increase. (Modern CPU Architecture lecture, slide 10)**

20. What is the purpose of a mutex object?

**A mutex is a tool that allows a thread to have mutually exclusive access to shared data. (Mutexes lecture, slide 5)**