

Test #1

- Due May 5 at 11:59pm
- Points 100
- Questions 40
- Available May 1 at 12:01am - May 5 at 11:59pm
- Time Limit 60 Minutes

Instructions

Canvas calls this a "Quiz", but it is really Test #1.

It consists of 40 multiple choice questions to be done in 60 minutes. It is Open Notes.

Once you start, you must finish. Canvas will not let you pause and come back.

Attempt History

	Attempt	Time	Score
LATEST	<u>Attempt 1</u>	60 minutes	100 out of 100

⚠ Correct answers will be available on May 6 at 12:01am.

Score for this quiz: 100 out of 100

Submitted May 1 at 11:25pm

This attempt took 60 minutes.



Question 1

2.5 / 2.5 pts

The advantage of using the OpenMP *reduction* clause is

- ☐ Actually a disadvantage -- it can produce wrong, non-deterministic answers
- ☐ It is less likely to result in a compiler error
- ☒ It greatly speeds, and makes thread-safe, reduction operations
- ☐ No advantage, it is just cleaner code



Question 2

2.5 / 2.5 pts

Hyperthreading is:

- ☒ Keeping one or more extra thread states within a core

- ☐ Adding extra cache space
- ☐ Adding more memory bandwidth
- ☐ Adding one or more cores



Question 3

2.5 / 2.5 pts

In multithreading, the threads all share:

- ☐ Execution instructions, Global variables, and the same Stack
- ☐ Heap, Global variables, and the same Stack
- ☐ Heap, Execution instructions, and the same Stack
- ☒ Heap, Execution instructions, and Global variables



Question 4

2.5 / 2.5 pts

A Chunksize of 2:

- ☐ Breaks your array into 2 uneven pieces
- ☐ Breaks your array into 2 even pieces
- ☒ Deals two for-loop passes to each thread and then goes around to each thread again, etc.
- ☐ Uses two threads only



Question 5

2.5 / 2.5 pts

MESI stands for:

- ☐ Modified-Exterior-Shared-Invalid
- ☐ Modified-Exclusive-Shared-Instructions
- ☐ Multicore-Exclusive-Shared-Invalid
- ☐ Modified-Exclusive-Single-Invalid
- ☐ Nothing -- it's someone's name
- ☒ Modified-Exclusive-Shared-Invalid



Question 6

2.5 / 2.5 pts

The purpose of the Watcher Thread in our Functional Decomposition example program is to:

- ☐ Draw a picture of what is going on in the simulation
- ☐ Figure out what the animal or plant threads need to do next

☒ Print results, update the current month/year, and update environmental variables

☐ Time the simulation



Question 7

2.5 / 2.5 pts

OpenMP Reductions are faster than Atomic or Critical because:

- ☐ They momentarily disable interrupts to keep the summing equation from being corrupted
- ☒ They sum into a separate variable per thread and then perform power-of-two addition
- ☐ They sum into a user-supplied array and then let the programmer decide how to best sum them
- ☐ They sum into an array whose elements are a Fibonacci series in size



Question 8

2.5 / 2.5 pts

The line “#pragma omp single” is used to:

- ☐ Force this block of code to be executed in single-file order by each thread
- ☐ Force this block of code to be divided up into individual OpenMP sections
- ☐ Force this block of code to undergo a single reduction
- ☒ Force this block of code to be executed by one thread only



Question 9

2.5 / 2.5 pts

A good way to make a piece of code *not* Thread Safe is to:

- ☐ Use a mutual exclusion lock
- ☐ Use a chunksize of 1
- ☐ Use a private variable
- ☒ Keep internal state



Question 10

2.5 / 2.5 pts

The two types of coherence that caches want to see in order to deliver maximum performance are:

- ☒ Spatial and Temporal
- ☐ Systemic and Thermal
- ☐ Spatial and Thermal
- ☐ Systemic and Temporal

⋮
Question 11

2.5 / 2.5 pts

The difference between static and dynamic scheduling of an OpenMP for-loop is:

- ☐ Dynamic scheduling changes the chunksize while the for-loop is running
- ☐ Dynamic scheduling allows you to change how the for-loop passes are divided up while they are running
- ☐ Dynamic scheduling divides all the for-loop passes among the threads at first
- ☒ Dynamic scheduling divides only some of the for-loop passes among the threads at first

⋮
Question 12

2.5 / 2.5 pts

Declaring a variable inside an OpenMP for-loop automatically makes it:

- ☒ Private
- ☐ Static
- ☐ Shared
- ☐ Global

⋮
Question 13

2.5 / 2.5 pts

In terms of 8-byte double-precision numbers, the size of a cache line is:

- ☐ 16 double-precision numbers
- ☐ 4 double-precision numbers
- ☒ 8 double-precision numbers
- ☐ 32 double-precision numbers

⋮
Question 14

2.5 / 2.5 pts

Intel recently broke the CPU clock speed record by:

- ☐ Running the CPU outside during a colder-than-usual winter
- ☐ Cooling the chip with liquid FlourInert
- ☐ Cooling the chip with four fans
- ☒ Cooling the chip with liquid helium

⋮
Question 15

2.5 / 2.5 pts

What does this code cause to happen?

```
#pragma omp atomic
```

```
sum = sum + partialSum;
```

- ☐ Imitates the same functionality as an OpenMP collapse clause
- ☒ Guarantees that the entire statement happens with no chance of interruption
- ☐ Automatically makes the variable sum a private variable
- ☐ Automatically makes the variable sum a shared variable



Question 16

2.5 / 2.5 pts

Which of these is an example of a forbidden *inter-loop dependency*?

- ☐ $a[i] = 2 * a[i];$
- ☒ $a[i] = a[i-1] + 1.;$
- ☐ $a[i] = (\text{float})(i);$
- ☐ $a[i] = b[i] + 1.;$



Question 17

2.5 / 2.5 pts

Why is there a photo of a carton of eggs in the Cache notes?

- ☐ Because the size of a cache line is a dozen floats
- ☐ It explains Temporary Coherence
- ☒ Bringing home a dozen eggs when you only need 2 today is like the way cache works
- ☐ It explains Stationary Coherence



Question 18

2.5 / 2.5 pts

When multiplying two arrays together, you've decided you want to do it **4-pairs-of-numbers** at a time. The type of SIMD you should use is:

- ☐ MMX
- ☒ SSE
- ☐ AVX-512
- ☐ AVX



Question 19

2.5 / 2.5 pts

SPMD stands for:

- ☐ Significant Parallelism, Much Data
- ☒ Single Program, Multiple Data
- ☐ Significant Parallelism, Multiple Data
- ☐ Single Program, Much Data



Question 20

2.5 / 2.5 pts

A thread's state consists of:

- ☐ Stack, Program counter, Registers
- ☒ Stack pointer, Program counter, Registers
- ☐ Stack pointer, Program counter, Stack
- ☐ Stack pointer, Stack, Registers



Question 21

2.5 / 2.5 pts

In an n-core multicore program, what do you need to do to compute the F_{parallel} ?

- ☐ Go find out the size of the cache and use the inverse Amdahl's Law
- ☐ Measure just the 20-core performance and use the inverse Amdahl's Law
- ☐ Figure out how many CPU sockets are in use and use the inverse Amdahl's Law
- ☒ Measure the Speedup and use our inverse Amdahl's Law



Question 22

2.5 / 2.5 pts

False Sharing happens because:

- ☐ One core is writing to a cache line at the same time another core is reading or writing a different cache line
- ☐ Two cores are not sharing the same cache line, but should be
- ☐ Two cores are reading from the same cache line
- ☒ One core is writing to a cache line at the same time another core is reading or writing the same cache line



Question 23

2.5 / 2.5 pts

A Deadlock condition is when:

- ☐ When you keep internal state

- ☐ The CPU chip cannot find any more instructions to execute while waiting for a memory fetch
- ☒ Two threads are each waiting for the other one to do something
- ☐ When it is a race to see which of two threads get to a piece of code first



Question 24

2.5 / 2.5 pts

Using “default(none)” in an OpenMP #pragma is:

- ☐ Required
- ☐ A deprecated feature of an older version of OpenMP
- ☐ A way to possibly increase performance
- ☒ A good idea, but not required



Question 25

2.5 / 2.5 pts

To get an A in CS 475/575 requires:

- ☐ A weighted average of 96%
- ☐ A weighted average of 93%
- ☒ 1060 total points



Question 26

2.5 / 2.5 pts

A Private variable differs from a Shared variable in that:

- ☐ Writing to it automatically triggers a cache line reload
- ☐ Writing to it automatically triggers a power-of-two reduction operation
- ☒ Each thread has its own copy of it
- ☐ When each thread writes to it, the value goes to the same memory address



Question 27

2.5 / 2.5 pts

The cache that is smallest and fastest is named:

- ☐ L3
- ☒ L1
- ☐ L2
- ☐ L0



Question 28

2.5 / 2.5 pts

Our class's "Inverse Amdahl's Law" that you used in Projects #0 and #1 computes:

- ☐ Thread Efficiency, given S_n and n
- ☐ S_n , given F_p and n
- ☒ F_p , given S_n and n
- ☐ n , given S_n and F_p



Question 29

2.5 / 2.5 pts

One way to prevent harm from race conditions is:

- ☐ Dynamic scheduling
- ☐ Shared variables
- ☒ Mutual Exclusion Locks
- ☐ Private variables



Question 30

2.5 / 2.5 pts

Speedup Efficiency is defined as:

- ☐ F_p/n
- ☒ S_n/n
- ☐ F_p
- ☐ n



Question 31

2.5 / 2.5 pts

A "Mutex" is:

- ☐ A sound you make when you sneeze
- ☒ Another term for a "mutual exclusion lock"
- ☐ A "multiple texture" for graphics processing
- ☐ A "mutual text" message



Question 32

2.5 / 2.5 pts

The OpenMP *collapse* clause is used to:

- ☒ Allow the parallelization of more than one nested for-loop
- ☐ Unroll a for-loop

- ☐ Turn a group of constants multiplied together into a single constant
- ☐ Turn cascading if-statements into a single compound if-statement



Question 33

2.5 / 2.5 pts

The word “deterministic” means:

- ☐ The program outputs change every time you run the program
- ☒ The same inputs will always produce the same outputs
- ☐ It describes a quantity that you are attempting to determine
- ☐ The program outputs change whenever you change the number of threads



Question 34

2.5 / 2.5 pts

How many total Bonus Days are you allowed in CS 475/575?

- ☐ 6
- ☐ 3
- ☐ 2
- ☒ 5
- ☐ 4



Question 35

2.5 / 2.5 pts

The *observation* that *clock speed* doubles every 2 years:

- ☒ Was the case for a while, but does not apply anymore
- ☐ Is only correct for CPUs, not GPUs
- ☐ Was never actually observed on real systems
- ☐ Has been correct starting in 1965 and is still happening



Question 36

2.5 / 2.5 pts

Coarse-grained parallelism is:

- ☐ Dividing the problem into equal-size pieces
- ☒ Dividing the problem into a small number of large pieces
- ☐ Dividing the problem into pieces, of all which have to be a different size

☐ Dividing the problem into a large number of small pieces



Question 37

2.5 / 2.5 pts

Which of these would *not* be a good use for `#pragma omp single` ?

☒ When we want all threads to reach a certain point in the code before proceeding

☐ When we want to read a data file

☐ When we want to print some values

☐ When we want to create OpenMP tasks to traverse a linked list



Question 38

2.5 / 2.5 pts

The theoretical maximum speedup that you can ever achieve, no matter how many cores you add, is:

☒ $1/(1-F_p)$

☐ $1/(F_p+F_s)$

☐ $1/F_p$

☐ F_s



Question 39

2.5 / 2.5 pts

A Barrier is:

☒ A place in the code that all threads must reach before any of them are allowed to continue

☐ A place in the code where threads can spawn other threads

☐ A place in the code where the first thread to get there issues an interrupt

☐ A place in the code that threads are not allowed to pass ever



Question 40

2.5 / 2.5 pts

The cache that is closest to the Arithmetic Logic Unit (ALU) is named:

☐ L3

☒ L1

☐ L0

☐ L2

