Test #1

Due May 1 at 11:59pm **Points** 100 **Questions** 40
Available Apr 27 at 2pm - May 1 at 11:59pm 4 days **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 and Closed Friends.

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

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	41 minutes	100 out of 100

Correct answers will be available on May 2 at 12:01am.

Score for this quiz: **100** out of 100
Submitted Apr 29 at 12:20am
This attempt took 41 minutes.

Question 1

2.5 / 2.5 pts

The difference between omp_set_lock( ) and omp_test_lock( ) is:

○ The second one blocks, the first one doesn't

○ The second one sends an interrupt, the first one doesn't

● The first one blocks, the second one doesn't

○ The first one sends an interrupt, the second one doesn't

Question 2

2.5 / 2.5 pts

AMD recently achieved a remarkably-high CPU clock speed by:

○ Cooling the chip with four fans

○ Running the CPU outside at the north pole

● Cooling the chip with liquid nitrogen

○ Running the CPU in the Penguin Encounter at Sea World

Question 3

2.5 / 2.5 pts

Moore's Law (as Gordon Moore *actually* phrased it) says:

● Transistor density doubles every 1.5 years

○ The number of cores doubles every 1.5 years

○ Parallel fraction doubles every 1.5 years

○ Clock speed doubles every 1.5 years

Question 4

2.5 / 2.5 pts

In multithreading, the threads all share:

○ Heap, Execution instructions, and the same Stack

○ Execution instructions, Global variables, and the same Stack

● Heap, Execution instructions, and Global variables

○ Heap, Global variables, and the same Stack

Question 5

2.5 / 2.5 pts

One of the nice features of OpenMP is that it guarantees *identical behavior* across different vendors and hardware.

○ True

● False

Question 6

2.5 / 2.5 pts

The *observation* that clock speed doubles every 1.5 years:

○ Is only correct for CPUs, not GPUs

○ Was never actually observed

● Was the case for a while, but does not apply anymore

○ Has been correct starting in 1965 and is still happening

Question 7

2.5 / 2.5 pts

The cache that is smallest and fastest is:

○ L3

○ L2

● L1

○ L0

Question 8

2.5 / 2.5 pts

SPMD stands for:

○ Significant Parallelism, Much Data

○ Significant Parallelism, Multiple Data

● Single Program, Multiple Data

○ Single Program, Much Data

Question 9

2.5 / 2.5 pts

In Project #2 (Numeric Integration), why do you need to double the volume you compute (the Z axis is up-down, the X axis is left-right)?

○ Because superquadrics have both a left and right half and we are only computing the volume of the right half

● Because superquadrics have both a top and bottom half and we are only computing the volume of the top half

○ Because superquadrics have both a top and bottom half and we are only computing the volume of the bottom half

○ Because superquadrics have both a left and right half and we are only computing the volume of the left half

Question 10

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

○ When each thread writes to it, the value goes to the same memory address

● Each thread has its own copy of it

Question 11

2.5 / 2.5 pts

When using OpenMP Tasks to apply parallelism to traversing a binary tree, the uniformity of the distribution of tasks among the threads:

○ Depends on how well you use the OpenMP task clauses

○ Depends on the amount of physical memory you have

● Depends on the compiler

○ Depends on the type of CPU

Question 12

2.5 / 2.5 pts

Jane Parallel runs Project #1 (Monte Carlo) with a NUMTRIALS set to one.

What will be her computed probability?

○ 50%

● 0% or 100%

○ Approximately 29%

Question 13

2.5 / 2.5 pts

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

○ 6

○ 2

○ 3

● 5

○ 4

Question 14

2.5 / 2.5 pts

When adding up the elements of a 2D array in C or C++, it is faster to add the elements:

○ It makes no speed difference either way

● Horizontally (i.e., across the rows) first

○ Vetically (i.e., down the columns) first

Question 15

2.5 / 2.5 pts

A way to prevent harm from race conditions is:

○ Shared variables

● Mutual Exclusion Locks

○ Private variables

○ Dynamic scheduling

Question 16

2.5 / 2.5 pts

The reason that our OpenMP programs have a NUMTRIES for-loop is to:

○ Determine the median performance

○ Determine the range of performance numbers

● Determine the peak performance

○ Determine the standard deviation of performance

Question 17

2.5 / 2.5 pts

A thread’s state consists of:

○ Stack, Program counter, Registers

○ Stack pointer, Stack, Registers

○ Stack pointer, Program counter, Stack

● Stack pointer, Program counter, Registers

Question 18

2.5 / 2.5 pts

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

○ Shared

○ Static

○ Global

● Private

Question 19

2.5 / 2.5 pts

A Barrier is:

○ A location in code that threads are not allowed to pass ever

○ A location in the code where threads can spawn other threads

● A location in the code that all threads must reach before any of them are allowed to pass

○ A location in the code where the first thread to get there issues an interrupt

Question 20

2.5 / 2.5 pts

The word “deterministic” means:

○ The program outputs change every time you run the program

○ The program outputs change whenever you change the number of threads

● The same inputs will always produce the same outputs

○ It describes a quantity that you are attempting to determine

Question 21

2.5 / 2.5 pts

You cannot use multithreading without having a multicore system.

○ True

● False

Question 22

2.5 / 2.5 pts

False Sharing happens because

○ Two cores are not using the same cache line, but should be

○ Two cores are reading from the same cache line

● A core writes to the same cache line that another core is reading from

○ Two cores have loaded cache lines for adjacent memory locations

Question 23

2.5 / 2.5 pts

Coarse-grained parallelism is:

○ Dividing the problem into a large number of small pieces

● Dividing the problem into a small number of large pieces

○ Dividing the problem into equal-size pieces

○ Dividing the problem into pieces, of all which have to be a different size

Question 24

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 25

2.5 / 2.5 pts

The advantage of using the OpenMP *reduction* clause is

○ No advantage, it is just cleaner code

○ 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

Question 26

2.5 / 2.5 pts

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

● 1/Fs

○ 1/Fp

○ 1/(Fp+Fs)

Question 27

2.5 / 2.5 pts

The difference between using OpenMP Tasks vs. using OpenMP Sections is that:

● Tasks are dynamically allocated, sections are static

○ Tasks are statically allocated, sections are dynamic

○ Sections are deprecated

○ Nothing -- they are different words for the same thing

Question 28

2.5 / 2.5 pts

A "Mutex" is:

○ A "mutual text" message

○ A sound you make when you sneeze

○ A "multiple texture" for graphics processing

● Another term for a "mutual exclusion lock"

Question 29

2.5 / 2.5 pts

OpenMP Reductions are faster than Atomic or Critical because:

○ They sum into a user-supplied array and then let the programmer decide how to best sum them

○ They momentarily disable interrupts to keep the summing equation from being corrupted

○ They sum into an array whose elements are a Fibonacci series in size

● They sum into a separate variable per thread and then perform power-of-two addition

Question 30

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 executed by one thread only

○ Force this block of code to undergo a single reduction

○ Force this block of code to be divided up into individual OpenMP sections

Question 31

2.5 / 2.5 pts

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

○ L2

○ L3

● L1

○ L0

Question 32

2.5 / 2.5 pts

In Project #1 (Monte Carlo) Joe Graphics coded this:

float sthd = sths[n];
float svx = sv * cos(sthd);
float svy = sv * sin(sthd);

and got the wrong probability. Why?

○ Called the wrong trig functions.

○ Accidentally switched cos and sin

● Forgot to turn degrees to radians.

Question 33

2.5 / 2.5 pts

How many multiplies can an SSE instruction perform at once?

○ 8

● 4

○ 2

○ 16

Question 34

2.5 / 2.5 pts

A Deadlock condition is when:

● Two threads are each waiting for the other one to do something

○ The CPU chip cannot find any more instructions to execute while waiting for a memory fetch

○ When it is a race to see which of two threads get to a piece of code first

○ When you keep internal state

Question 35

2.5 / 2.5 pts

"Prefetch Distance" is:

○ How many bytes you expect to use from a particular cache line

○ How many CPU cycles get used while waiting for a cache line to load

○ How many cache lines you are loading at a time

● How far ahead you are loading a cache line before you need it

Question 36

2.5 / 2.5 pts

Speedup Efficiency is defined as:

○ Fp

○ n

○ Fp/n

● Sn/n

Question 37

2.5 / 2.5 pts

Our class’s “Inverse Amdahl’s Law” that you used in Projects #0, #1, and #2 computes:

○ Thread Efficiency, given Sn and n

○ n, given Sn and Fp

● Fp, given Sn and n

○ Sn, given Fp and n

Question 38

2.5 / 2.5 pts

A “race condition” is one where:

○ You get the same result regardless of which thread gets to a piece of code first

● You get a different result depending on which thread gets to a piece of code first

○ It matters which stack holds a particular variable

○ It matters which thread gets to a barrier first

Question 39

2.5 / 2.5 pts

Gustafson’s Observation on Amdahl’s Law says:

● More cores often results in more data, which results in a larger parallel fraction

○ Amdahl’s law was applicable when it was formulated, but doesn’t apply now

○ More cores often results in memory contention and decreases performance

○ Amdahl’s Law only applies when you have a number of cores that is less than or equal to 8

Question 40

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

Quiz Score: **100** out of 100

▶