Test #1

截止时间 5月1日 23:59 **得分** 100 **问题** 40
可用 4月27日 14:00 至 5月1日 23:59 4 天 **时间限制** 60 分钟

说明

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.

尝试历史记录

	尝试	时间	分数
最新	尝试 1	41 分钟	100，满分 100 分

ⓘ 正确答案将于 5月2日 0:01 提供。

此测验的分数： **100**，满分 100 分
提交时间 4月30日 17:26
此尝试进行了 41 分钟。

问题 1 2.5 / 2.5 分

A Barrier is:

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

问题 2

2.5 / 2.5 分

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 thread gets to a barrier first
- It matters which stack holds a particular variable

问题 3

2.5 / 2.5 分

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

- 1/Fs
- 1/(Fp+Fs)
- 1/Fp

问题 4

2.5 / 2.5 分

In terms of 32-bit floating-point numbers, the size of a cache line is typically:

- 64 floating-point numbers
- 16 floating-point numbers
- 8 floating-point numbers
- 32 floating-point numbers

问题 5

2.5 / 2.5 分

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

- True
- False

问题 6

2.5 / 2.5 分

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

- L1

- L2
- L3
- L0

问题 7

2.5 / 2.5 分

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

- Global
- Static
- Shared
- Private

问题 8

2.5 / 2.5 分

In multithreading, the threads all share:

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

问题 9

2.5 / 2.5 分

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

- 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 changes the chunksize while the for-loop is running
- Dynamic scheduling divides only some of the for-loop passes among the threads at first

问题 10

2.5 / 2.5 分

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

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

问题 11

2.5 / 2.5 分

A way to prevent harm from race conditions is:

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

问题 12

2.5 / 2.5 分

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

- Parallel fraction doubles every 1.5 years
- The number of cores doubles every 1.5 years
- Transistor density doubles every 1.5 years
- Clock speed doubles every 1.5 years

问题 13

2.5 / 2.5 分

The Compute-to-Communicate ratio tells us:

How much computing a core can do before it has to share values with adjacent cores

How much computing a core can do before it has to stop at a barrier

How much computing a core can do before it has to broadcast values to all other cores

How much computing a core can do before it hits a Mutex

问题 14

2.5 / 2.5 分

A good way to make a piece of code not Thread Safe (such as strtok) is to:

Use a chunksize of 1

Use a mutual exclusion lock

Keep internal state

Use a private variable

问题 15

2.5 / 2.5 分

How many multiplies can an SSE instruction perform at once?

4

- 2
- 16
- 8

问题 16

2.5 / 2.5 分

Our class's "Inverse Amdahl's Law" that you used in Projects #0, #1, and #2 computes:

- n, given Sn and Fp
- Sn, given Fp and n
- Thread Efficiency, given Sn and n
- (•) Fp, given Sn and n

问题 17

2.5 / 2.5 分

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
- Nothing -- they are different words for the same thing
- Sections are deprecated

问题 18

2.5 / 2.5 分

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

- Force this block of code to be divided up into individual OpenMP sections
- 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 executed in single-file order by each thread

问题 19

2.5 / 2.5 分

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

- Time the simulation
- Draw a picture of what is going on in the simulation
- Print results and update environmental variables
- Figure out what the animal or plant threads need to do next

问题 20

2.5 / 2.5 分

You cannot use multithreading without having a multicore system.

- True
- False

问题 21

2.5 / 2.5 分

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

问题 22

2.5 / 2.5 分

SSE SIMD performs:

- 4 floating-point multiplies in one instruction
- 4 byte-multiplies in one instruction
- 4 double-precision multiplies in one instruction

问题 23

2.5 / 2.5 分

A Deadlock condition is when:

- (●) 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
- ( ) The CPU chip cannot find any more instructions to execute while waiting for a memory fetch
- ( ) When you keep internal state

问题 24

2.5 / 2.5 分

A thread's state consists of:

- ( ) Stack, Program counter, Registers
- ( ) Stack pointer, Program counter, Stack
- ( ) Stack pointer, Stack, Registers
- (●) Stack pointer, Program counter, Registers

问题 25

2.5 / 2.5 分

Gustafson's Observation on Amdahl's Law says:

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

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

More cores often results in memory contention and decreases performance

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

问题 26

2.5 / 2.5 分

Speedup Efficiency is defined as:

Sn/n

n

Fp/n

Fp

问题 27

2.5 / 2.5 分

A "Mutex" is:

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

问题 28

2.5 / 2.5 分

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

- Depends on the type of CPU
- Depends on the amount of physical memory you have
- Depends on the compiler
- Depends on how well you use the OpenMP task clauses

问题 29

2.5 / 2.5 分

A Private variable differs from a Shared variable in that:

- Each thread has its own copy of it
- 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

Writing to it automatically triggers a cache line reload

问题 30

2.5 / 2.5 分

The advantage of using the OpenMP *reduction* clause is

It is less likely to result in a compiler error

No advantage, it is just cleaner code

Actually a disadvantage -- it can produce wrong, non-deterministic answers

It greatly speeds, and makes thread-safe, reduction operations

问题 31

2.5 / 2.5 分

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

Because caches are easily broken

Because cache lines always have a dozen bytes in them

No logical reason -- it looks cool

Bringing home a dozen eggs when you only need 3 today is like reading a cache line when you only need one memory value

问题 32

2.5 / 2.5 分

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?

- Forgot to turn degrees to radians.
- Accidentally switched cos and sin
- Called the wrong trig functions.

问题 33

2.5 / 2.5 分

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

问题 34

2.5 / 2.5 分

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

- Running the CPU in the Penguin Encounter at Sea World
- Running the CPU outside at the north pole
- Cooling the chip with liquid nitrogen
- Cooling the chip with four fans

问题 35

2.5 / 2.5 分

False Sharing happens because

- Two cores have loaded cache lines for adjacent memory locations
- A core writes to the same cache line that another core is reading from
- Two cores are reading from the same cache line
- Two cores are not using the same cache line, but should be

问题 36

2.5 / 2.5 分

SPMD stands for:

- Significant Parallelism, Multiple Data
- Single Program, Much Data
- Significant Parallelism, Much Data
- (●) Single Program, Multiple Data

问题 37

2.5 / 2.5 分

If you have a working multicore program, can you compute the $F_{parallel}$?

- No, it's too compliicated.
- (●) Yes, measure a speedup and use the inverse Amdahl's Law
- Yes, but it will require more knowledge than we are covering here
- No, it's too unpredictable

问题 38

2.5 / 2.5 分

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 left and right half and we are only computing the volume of the left 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

问题 39

2.5 / 2.5 分

MESI stands for:

Modified-Exclusive-Shared-Instructions

Modified-Exclusive-Shared-Invalid

Nothing -- it's someone's name

Multicore-Exclusive-Shared-Invalid

Modified-Exterior-Shared-Invalid

Modified-Exclusive-Single-Invalid

问题 40

2.5 / 2.5 分

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

- Spatial and Thermal
- Systemic and Temporal
- Systemic and Thermal
- Spatial and Temporal

测验分数： **100**，满分 100 分