

# Test #1

截止日期 5月1日 23:59 分數 100 問題 40  
可用 4月27日 14:00 - 5月1日 23:59 4 天 時間限制 60 分鐘

## 說明

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.

## 嘗試記錄

	嘗試	時間	分數
最新的	<a href="#">嘗試 1</a>	43 分鐘	得分 : 100 ; 總分 : 100

 ① 正確答案將於 5月2日 0:01 可用。

此測驗的分數：得分：**100**；總分：100

已提交4月29日 14:58

此嘗試持續 43 分鐘。

問題 1	2.5 / 2.5 分數
<p><b>MESI stands for:</b></p> <p><input type="radio"/> Multicore-Exclusive-Shared-Invalid</p> <p><input type="radio"/> Nothing -- it's someone's name</p> <p><input type="radio"/> Modified-Exclusive-Shared-Instructions</p> <p><input type="radio"/> Modified-Exclusive-Single-Invalid</p>	

Modified-Exclusive-Shared-Invalid

Modified-Exterior-Shared-Invalid

## 問題 2

2.5 / 2.5 分數

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

Dynamic scheduling divides only some of the for-loop passes among the threads at first

Dynamic scheduling divides all the for-loop passes among the threads at first

Dynamic scheduling allows you to change how the for-loop passes are divided up while they are running

Dynamic scheduling changes the chunksize while the for-loop is running

## 問題 3

2.5 / 2.5 分數

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

A good idea, but not required

- Required
- A deprecated feature of an older version of OpenMP
- A way to possibly increase performance

**問題 4****2.5 / 2.5 分數**

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

- Vertically (i.e., down the columns) first
- It makes no speed difference either way
- Horizontally (i.e., across the rows) first

**問題 5****2.5 / 2.5 分數**

**You cannot use multithreading without having a multicore system.**

- True
- False

**問題 6****2.5 / 2.5 分數**

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

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

### 問題 7

2.5 / 2.5 分數

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

- Sections are deprecated
- Tasks are dynamically allocated, sections are static
- Tasks are statically allocated, sections are dynamic
- Nothing -- they are different words for the same thing

### 問題 8

2.5 / 2.5 分數

A Deadlock condition is when:



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



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



When you keep internal state

### 問題 9

2.5 / 2.5 分數

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



6



2



5



3



4

### 問題 10

2.5 / 2.5 分數

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



Shared

- Global
- Static
- Private

### 問題 11

2.5 / 2.5 分數

A “race condition” is one where:

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



### 問題 12

2.5 / 2.5 分數

Speedup Efficiency is defined as:

- n
- Fp
- Fp/n

Sn/n**問題 13****2.5 / 2.5 分數****The word “deterministic” means:**

The program outputs change whenever you change the number of threads



It describes a quantity that you are attempting to determine



The program outputs change every time you run the program



The same inputs will always produce the same outputs

**問題 14****2.5 / 2.5 分數****"Prefetch Distance" is:**

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



How many cache lines you are loading at a time



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



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

**問題 15****2.5 / 2.5 分數**

**Coarse-grained parallelism is:**

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



Dividing the problem into a small number of large pieces



Dividing the problem into equal-size pieces



Dividing the problem into a large number of small pieces

**問題 16**

2.5 / 2.5 分數

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



Determine the range of performance numbers



Determine the standard deviation of performance



Determine the median performance



Determine the peak performance

**問題 17**

2.5 / 2.5 分數

**A Chunksize of 2:**



Breaks your array into 2 even pieces



Deals two for-loop passes to each thread and then goes around to each thread again, etc.

- Breaks your array into 2 uneven pieces
- Uses two threads only

### 問題 18

2.5 / 2.5 分數

**Our definition of Speedup is (P=performance, T=elapsed time):**

- $S = T_n/T_1$
- $S = n/(n-1)$
- $S = P_n/P_1$



### 問題 19

2.5 / 2.5 分數

**In multithreading, the threads all share:**

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

**問題 20****2.5 / 2.5 分數**

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

 L2 L3 L0 L1**問題 21****2.5 / 2.5 分數**

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

 The second one sends an interrupt, the first one doesn't The second one blocks, the first one doesn't The first one sends an interrupt, the second one doesn't The first one blocks, the second one doesn't**問題 22****2.5 / 2.5 分數**

**SPMD stands for:**

 Significant Parallelism, Much Data

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

**問題 23****2.5 / 2.5 分數****A thread's state consists of:**

- Stack, Program counter, Registers
- Stack pointer, Program counter, Stack
- Stack pointer, Stack, Registers
- Stack pointer, Program counter, Registers

**問題 24****2.5 / 2.5 分數****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
- It greatly speeds, and makes thread-safe, reduction operations
- Actually a disadvantage -- it can produce wrong, non-deterministic answers

**問題 25****2.5 / 2.5 分數****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.
  - Called the wrong trig functions.
  - Accidentally switched cos and sin
- 

**問題 26****2.5 / 2.5 分數****The two types of coherence that caches want to see in order to deliver maximum performance are:**

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

**問題 27****2.5 / 2.5 分數**

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

- $a[i] = b[i] + 1.;$
- $a[i] = a[i-1] + 1.;$
- $a[i] = 2.*a[i];$

### 問題 28

2.5 / 2.5 分數

False Sharing happens because

- 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
- Two cores have loaded cache lines for adjacent memory locations



### 問題 29

2.5 / 2.5 分數

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 amount of physical memory you have
- Depends on the type of CPU

- Depends on how well you use the OpenMP task clauses
- Depends on the compiler

**問題 30****2.5 / 2.5 分數****The cache that is smallest and fastest is:**

- L3
- L0
- L1
- L2

**問題 31****2.5 / 2.5 分數****A good way to make a piece of code **not** Thread Safe (such as strtok) is to:**

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

**問題 32****2.5 / 2.5 分數**

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

**What will be her computed probability?**

- 50%
- Approximately 29%
- 0% or 100%

**問題 33****2.5 / 2.5 分數**

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

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

**問題 34****2.5 / 2.5 分數**

**A "Mutex" is:**

- A sound you make when you sneeze

- A "multiple texture" for graphics processing
- A "mutual text" message
- Another term for a "mutual exclusion lock"

**問題 35****2.5 / 2.5 分數****The line “#pragma omp single” is used to:**

- Force this block of code to undergo a single reduction
- Force this block of code to be executed by one thread only
- 
- 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

**問題 36****2.5 / 2.5 分數****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
- 
- Time the simulation
- 
- Figure out what the animal or plant threads need to do next
- 
- Print results and update environmental variables

**問題 37****2.5 / 2.5 分數****Why is there a photo of a carton of eggs in the Cache notes?**

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

- No logical reason -- it looks cool
- Because caches are easily broken
- Because cache lines always have a dozen bytes in them

**問題 38****2.5 / 2.5 分數****Hyperthreading is:**

- Adding extra cache space
- Adding more memory bandwidth
- Keeping one or more extra thread states within a core
- Adding one or more cores

**問題 39****2.5 / 2.5 分數**

## SSE SIMD performs:

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

### 問題 40

2.5 / 2.5 分數

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

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



測驗分數：得分：**100**；總分：100