國立中正大學
115學年度碩士班招生考試
試 題

[第4節]

科目名稱	計算機系統
系所組別	資訊工程學系

—作答注意事項—

※作答前請先核對「試題」、「試卷」與「准考證」之系所組別、科目名稱是否相符。

1. 預備鈴響時即可入場，但至考試開始鈴響前，不得翻閱試題，並不得書寫、畫記、作答。
2. 考試開始鈴響時，即可開始作答；考試結束鈴響畢，應即停止作答。
3. 入場後於考試開始40分鐘內不得離場。
4. 全部答題均須在試卷（答案卷）作答區內完成。
5. 試卷作答限用藍色或黑色筆（含鉛筆）書寫。
6. 試題須隨試卷繳還。

國立中正大學115學年度碩士班招生考試試題

科目名稱：計算機系統　　本科目共 3 頁　第 1 頁

系所組別：資訊工程學系

1. (30 points) Consider a 32-bit, byte-addressable system with a single cache that has the following specifications: total cache size = 4 KB (4096 bytes) and cache block size = 16 bytes. You will compare three cache organizations that all share the same total capacity and block size but differ in placement: (1) direct-mapped, (2) 4-way set-associative, and (3) fully-associative. Assume the cache is initially empty (all lines invalid). For the associative caches (4-way and fully-associative), assume LRU replacement. For writes, assume a write-back, write-allocate policy (i.e., on a write miss, the block is brought into the cache, and modified blocks are written back upon eviction).

   Now process the following 10 memory accesses in order, where each access is either a read (R) or write (W) to the given byte address: (1) R 0x12345678, (2) R 0x1234567C, (3) W 0x12345670, (4) R 0x12346678, (5) R 0x12347678, (6) W 0x12348678, (7) R 0x12345674, (8) R 0x12349678, (9) R 0x1234667C, (10) W 0x1234867C. For each cache organization (direct-mapped, 4-way set-associative, fully-associative), determine whether each of the 10 accesses is a cache hit or cache miss, and then report the total number of hits and misses for that organization.

2. (5 points) In Problem 1, if the main memory size is increased significantly (e.g., from 4 GB to 16 GB), is it mandatory to increase the total cache size (currently 4 KB) for the system to function correctly? (Increase, Decrease, or No Change Required)

3. (5 points) Our favorite program runs on Computer A in 10 seconds. Computer A uses a 5-stage pipeline, has an average CPI of 1.0, and operates at a clock rate of 2 GHz. A computer designer is considering a new machine, Computer B, with the goal of reducing the execution time of this program to 6 seconds.

   The Computer B uses a deeper 10-stage pipeline. The increased pipeline depth allows for a higher clock rate; however, it also introduces additional pipeline overhead, such as larger branch misprediction penalties. As a result, the average CPI of Computer B increases to 1.2. The clock rate of Computer B is not yet determined.

   Assume that the instruction count of the program is identical on Computer A and Computer B. Determine the clock rate that Computer B must achieve in order to meet the 6-second execution time target.

4. (10 points) Modern ARM processors often use a big.LITTLE architecture to reduce energy consumption by executing only performance-critical code on high-power cores while running the remaining code on energy-efficient cores. Consider an ARM-based SoC with one big core and one LITTLE core, where only one core is active at any time. The LITTLE core consumes 2 W when active, while the big core consumes 8 W when active. When a core is inactive, its power consumption can be ignored.

A given program takes 10 seconds to complete when executed entirely on the LITTLE core. Profiling shows that 30% of the execution time is performance-critical code that can benefit from the big core. The big core executes this portion 4× faster than the LITTLE core. There is no overhead for switching between cores.

The program is executed using the following strategy:

a. The performance-critical portion runs on the big core.
b. The remaining portion runs on the LITTLE core.

What is the average power consumption (in watts) of the program when executed using the ARM big.LITTLE strategy described above?

5. (10 points) CPU Scheduling & I/O Performance: CPU scheduling strategies significantly affect I/O system throughput.
   a. Explain why it is crucial to let I/O-bound processes acquire the CPU as early as possible to maximize I/O throughput.
   b. Using the Multilevel Feedback Queue algorithm as an example, explain how it dynamically identifies I/O-bound processes and adjusts their priorities to achieve this goal.

6. (10 points) Copy-on-Write (COW) Mechanism: When using the Copy-on-Write mechanism to optimize the fork() system call:
   a. Explain how the Page Table Entries (PTEs) for both the Parent and the Child should be configured immediately after fork() completes.
   b. When the Child process attempts to write to a shared page, a Page Fault is triggered. How does the Kernel distinguish whether this Page Fault is caused by the Copy-on-Write mechanism or by an invalid memory access?

7. (30 points) Simple Reader-Writer Spinlock: Consider the following Simple Reader-Writer Spinlock implemented using atomic operations. Assumption: atomic_sub(addr, val) atomically subtracts val from *addr and returns the old value (before subtraction). atomic_add performs atomic addition.
   a. Prove that Mutual Exclusion holds between a Writer and a Reader.
   b. Prove that multiple Readers can enter the Critical Section simultaneously (Concurrency).

```
#define MAXVAL 0x40000000
void init_spinlock(int* lock) { *lock = MAXVAL;}
void writer_lock(int* lock) {
    while (1) {
```

```
            int oldVal = atomic_sub(lock, MAXVAL);
            if (oldVal == MAXVAL) return; // Successfully acquired lock
            else atomic_add(lock, MAXVAL); // Failed, rollback
        }
    }
    void reader_lock(int* lock) {
        while (1) {
            int oldVal = atomic_sub(lock, 1);
            if (oldVal > 0) return; // Successfully acquired lock
            else atomic_add(lock, 1); // Failed, rollback
        }
    }
    void reader_unlock(int* lock) {   atomic_add(lock, 1);   }
    void writer_unlock(int* lock) {   atomic_add(lock, MAXVAL);   }
```