

# 國立中正大學

## 113 學年度碩士班招生考試

### 試題

[第 3 節]

科目名稱	計算機系統
系所組別	資訊工程學系-甲組

#### —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 113 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 5 頁 第 1 頁

系所組別：資訊工程學系-甲組

## 1. (30%) 選擇題 (2% each)

- (1) In OSs for modern machines, what is the immediate result if a process generates an address that is outside its legal address range?
  - A. Blue Screen of Death.
  - B. A hardware interrupt (trap) will occur.
  - C. The program will probably not work right.
  - D. The O/S call will return an error.
  - E. None of the above is the immediate result of generating an illegal address.
- (2) When a controller generates an interrupt, what happens?
  - A. The OS checks between each instruction to see if an interrupt occurred.
  - B. The OS checks to see which controller caused the interrupt.
  - C. The hardware jumps to the handler through the interrupt vector.
  - D. The running application calls the OS to handle the interrupt.
  - E. None of the above happens after a controller generates an interrupt.
- (3) Which best describes the architecture of Linux?
  - A. Microkernel
  - B. Microkernel with Dynamically Loadable Drivers
  - C. Monolithic
  - D. Monolithic with Dynamically Loadable Drivers
  - E. Macrokernel
- (4) What common technique used in uniprocessor systems allows a programmer to exploit parallelism on SMP systems?
  - A. Memory mapped files
  - B. Multithreading
  - C. Critical sections
  - D. Semaphores
  - E. Data sharing

# 國立中正大學 113 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 5 頁 第 2 頁

系所組別：資訊工程學系-甲組

- (5) What is the biggest problem with the FCFS CPU scheduling algorithm?
- A. It is hard to implement.
  - B. It is not fair.
  - C. A long job can delay shorter ones.
  - D. It is not intuitive to understand.
  - E. None of the above is true about the FCFS algorithm.
- (6) What is the problem with the Shortest Run Time First process scheduling algorithm?
- A. We don't know the next CPU burst length.
  - B. It may lead to starvation.
  - C. It can make shorter jobs wait behind longer jobs.
  - D. It uses too much CPU time to run.
  - E. None of the above is a problem with SRTF job scheduling.
- (7) When an OS is using a Multilevel Feedback Queuing system for scheduling, how is the CPU time divided between the queues?
- A. Each level gets a predefined percentage of the CPU time.
  - B. The top level is exhausted and then the next level gets a turn, etc.
  - C. Each level gets the same percentage of the CPU time.
  - D. The percentage of time allocated to a queue varies with the performance of the jobs in that queue. If they run too long the percentage is decreased.
  - E. Each OS is designed differently.
- (8) Why were multi-level page tables developed?
- A. It made lookup faster.
  - B. So they did not have to include the process identifier.
  - C. Page tables were so large that they caused external fragmentation.
  - D. So a system call was not necessarily.
  - E. None of the above
- (9) We could spend lots of CPU cycles to avoid making a disk access in the VM system when we try to find a free frame. Which would be an example of time well spent in such a pursuit?
- A. Swap out processes that are in a wait state.
  - B. Do garbage collection on the memory holes.
  - C. Sort the page table in order by reference count.
  - D. Clean dirty pages.
  - E. Predict the next page reference.

- (10) What is the buddy system?
- A. A method of sharing user IDs
  - B. A distributed election protocol
  - C. Running two CPUs in tandem so that they share the workload
  - D. Keeping track of free memory blocks in lists of like size
  - E. Keeping track of shared files
- (11) Authentication makes use of some special mechanism to verify the identity of an entity. Most often we are concerned with verifying the identity of a user. Which of the following did we NOT say was something that could be used to verify the identity of a user?
- A. Something you have
  - B. Something you see
  - C. Something you know
  - D. Something you are
  - E. All of the above can be used to verify a user's identity.
- (12) What is the "sandbox model" about?
- A. Executing untrusted code on a separate computer
  - B. Executing untrusted code on a separate CPU
  - C. Executing untrusted code using a separate network
  - D. Executing untrusted code in a virtual OS environment
  - E. Executing untrusted code in a controlled environment like a virtual machine
- (13) What is internal fragmentation in a file system?
- A. The blocks of a file are scattered all over the disk.
  - B. The available space is in pieces that are too small to be useful.
  - C. The allocation granularity does not exactly match a requested allocation.
  - D. A hard drive is divided into multiple partitions.
  - E. None of the describes internal fragmentation.

- (14) The disk scheduling algorithm that produced the minimum seek delay was the LOOK algorithm. But we kept tinkering with the algorithm and showed several other algorithms that had a longer seek time. What were we trying to accomplish with these algorithms?
- A. We wanted the scheduler to be fairer.
  - B. We wanted the scheduler to be faster.
  - C. We wanted to increase the reliability of the scheduler.
  - D. We were trying to make the scheduling algorithm take less CPU time.
  - E. We were searching for a better average seek time, but were not having any luck.
- (15) What application layer protocol is used in the Internet?
- A. Hypertext Transfer Protocol - HTTP
  - B. File Transfer Protocol - FTP
  - C. Simple Mail Transfer Protocol - SMTP
  - D. Post Office Protocol - POP3
  - E. All of the above are application layer protocols used in the Internet.

2. (10%) Consider the following code segment:

```
pid_t pid;
pid = fork();
if (pid == 0) {
    fork();
    thread_create( . . . );
}
fork();
```

- (a) How many unique processes are created?
- (b) How many unique threads are created?

3. (10%) Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

# 國立中正大學 113 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 5 頁 第 5 頁

系所組別：資訊工程學系-甲組

4. (20%) The following are 10 sequential “word addresses” in decimal:

**18, 121, 26, 123, 168, 169, 126, 68, 85, 89.**

- (a) Determine each access has a cache “hit”, a “compulsory” miss, a “conflict” miss, or a “capacity” miss, if there exists a direct-mapped cache with 10 blocks and each block contains only 1 word. (i.e., Choose one of the 4 access types for each of the 10 word addresses given above.)

- (b) Repeat (a) if each cache block contains 10 words.

5. (15%) The CPU performance equation is defined as

$$\text{CPU execution time} = \text{IC} \times \text{CPI} \times \text{clock cycle time}.$$

An  $N$ -stage pipelined CPU has an ideal speedup of  $N$  over a non-pipelined CPU. However, it is obvious that we cannot have an infinite speedup with  $N = \infty$ .

- (a) Describe how pipelining affects the CPU performance by “CPI” and “clock cycle time” respectively. (Note that pipelining does not change “IC”).

- (b) List two “CPI”-related issues that prevent the infinite speedup with  $N = \infty$ .

- (c) List two “clock cycle time”-related issues that prevent the infinite speedup with  $N = \infty$ .

6. (15%) The following is a MIPS assembly code to move data.

```
LOOP:  lw  $t0, -4($s0)      # assume $s0 has an initial value of 400
        sw  $t0, 996($s0)
        addi $s0, $s0, -4
        bne $s0, $zero, LOOP
        trap
```

- (a) Identify all pipeline hazards for executing the above code on a classical 5-stage pipelined MIPS (i.e., **IF/ID/EXE/MEM/WB**).

- (b) If there is no hardware for hazards (i.e., no interlock, no branch predictor, no forwarding ...), rewrite the code with minimally-inserted NOP(s) to prevent hazards. DO NOT change the instruction order, and assume branches (i.e., bne) are resolved at **EXE**.

- (c) How many cycles are needed for executing the code in (b)? (i.e., till trap completes **WB**)