國立中央大學資訊工程學系博士班 115 學年度第一學期資格考試題紙

科目：**作 業 系 統** (Operating System)　　第一頁　共二頁（page 1 of 2）

1. (15%) Consider process creation and communication in a operating system.
   (a) (6 %) Explain the return values of fork() in the parent and child, whether their ordinary memory writes affect each other, and what a successful exec() changes. Does exec() create a new process?
   (b) (6%) A parent creates a pipe, forks a child that writes more than the pipe capacity, and calls wait() before reading. Explain how both processes can block. Give a corrected sequence, identify the pipe ends each process must close, and state when the parent observes EOF. Assume blocking I/O and no errors.
   (c) (3%) An AI agent executes generated code in a child process. Is fork() followed by exec() sufficient to isolate untrusted code? Identify two inherited capabilities or shared resources that remain accessible, and propose controls that restrict access and resource use.

2. (20%) Consider deadlock and resource allocation in an operating system.
   (a) (8 %) State and explain the four necessary conditions for deadlock. For each condition, describe a prevention rule that breaks it. Discuss one limitation of applying these rules in practice.
   (b) (8 %)There are three units each of exclusive, non-preemptible resource types A and B. Allocations for P1, P2, P3 are (1,0), (1,1), (0,1); maximum total demands are (3,2), (1,2), (2,1). Each process releases its allocation when it finishes. Compute remaining needs and exhibit a safe completion sequence. Should an additional request (1,1) from P1 be granted under deadlock avoidance? Show the tentative state and distinguish an unsafe state from an existing deadlock.
   (c) (4 %) Two GPU jobs each hold memory needed for their current state while waiting for additional memory to launch their next kernel. Relate this situation to deadlock. Propose a prevention or recovery strategy, state the assumptions needed for progress, and explain its utilization or recovery cost.

3. (15%) Consider file semantics and crash consistency on a local file system.
   (a) (6 %) Distinguish a file descriptor, a directory entry, and an inode. Explain how hard links differ from symbolic links. If a process unlinks an open file, can it still access the file, and when may its storage be reclaimed?
   (b) (6 %) Design an update that atomically replaces file F with complete new contents and guarantees the new version after success is reported, even if power immediately fails. Use a temporary file in the same directory. Assume atomic rename(), file and directory fsync() support, and a device that honors flushes. Explain the operation order and why rename() alone is insufficient.
   (c) (3 %) An AI training checkpoint contains model parameters and optimizer state. A background thread saves them while training modifies both. Does journaling plus fsync() ensure a consistent checkpoint? Propose a snapshot and publication method, and explain one cost and how you would test recovery correctness.

4. (16 %) Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past.
   (a) Will this scheduling algorithm favor CPU-bound processes or I/O-bound processes? Why?
   (b) Please discuss if the starvation situation will permanently occur in this scheduling algorithm?

5. (10%) Explain the key drawback of spinlocks with busy-waiting compared to binary semaphores with sleeping.

6. (24%) Are the following statements true or false? For each statement, you will get 4 points for correct answer, zero point for blank, or -2 point for incorrect answer.
   (a) The optimal maximum spin time for a spinlock is independent of the time required to perform a context switch.
   (b) Pretending that deadlocks never occur in the system is a method to deal with deadlocks.
   (c) The virtual memory manager runs the guest operating systems, manages their resource use, and protects each guest from the other.
   (d) The nsloopup tool allows you to query the DNS to obtain the mapping between domain name and IP address.
   (e) The BGP protocol is an intra-AS routing protocol.
   (f) Side-channel attacks are fundamentally designed to exploit mathematical weaknesses or logical flaws within a cryptographic algorithm's design.