國立中央大學資訊工程學系博士班 112 學年度第二學期資格考試題紙

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

1. (15%) 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) (8%) Will this scheduling algorithm favor CPU-bound processes or I/O-bound processes? Why?

   (b) (7%) Please discuss if the starvation situation will permanently occur in this scheduling algorithm?

2. (15%) Suppose a thread is running in a critical section of code. It means that the thread has acquired all the locks through proper arbitration. Can this thread get context switched? Please explain the reasons.

3. (20 %) 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) Inverted page table is used to reduce the size of memory committed to page tables
   (b) An inverted page table has one entry for each page
   (c) TLS is used by IPSec to provide network security
   (d) The nsloopup tool allows you to query the DNS to obtain the mapping between domain name and IP address.
   (e) Ordinary pipes are unidirectional.

4. (10%) Consider a computer with a 64-bit logical address and a 4 KB page size. The system supports up to 2 GB of physical memory.
   (a) (5%) How many entries are there in a single-level page table?
   (b) (5%) How many frames are there in the physical memory?

5. (15%) Consider the following page reference string:

   1 2 3 2 4 6 2 7 3 7 6 3 7 4 2 5 3 6

   Assume that the system has N **frames**, which are initially empty. How many page replacements could occur if the following page replacement algorithms are used?
   (a) (5%) LRU algorithm for N=4
   (b) (5%) FIFO with pre-fetching the subsequent page, for N=4 (i.e. Assume that referencing page 1 causes a page fault. Then, the system evicts two pages and puts page 1 and page 2 into the frames)
   (c) (5%) Which of the above two algorithms has the best performance? Why?

背面還有 Please Turn Over

6. (25%) Java supports the automatic garbage collection mechanism. On the other hand, C++ does not support this mechanism. Please answer the following questions:
   (a) (5%) Explain why garbage collection makes applications slow at runtime.
   (b) (10%) The virtual memory space of a process/application is usually partitioned into several conceptual regions: code, static data, heap, and stack. What region (or regions) should be scanned by the garbage collector? Why?
   (c) (10%) What should a C++ programmer do in order to develop a robust application, in the case that C++ does not support automatic garbage collection?