

# 國立成功大學

## 114學年度碩士班招生考試試題

編 號：147

系 所：電機資訊學院-資訊聯招

科 目：計算機組織與系統

日 期：0210

節 次：第 1 節

注 意：1.不可使用計算機  
2.請於答案卷(卡)作答，於  
試題上作答，不予計分。

※考生請注意：本試題不可使用計算機。請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

※考試科目：計算機組織與系統。

※The paper contains two parts including problem sets for Operating System (Part I) and Computer Organization (Part II). Please read the paper carefully; provide and summarize your answers, explicitly, as requested.

## Part I: Operating Systems [50%]

Please summarize your answers, true (○) or false (×), explicitly in a table exemplified as follows.

1	(a)	(b)	(c)	
2	(a) ×	(b) ○	(c) ×	
3	(a)	(b)	(c)	(d)

- 1) [15%] (memory system) Consider a memory hierarchy consisting of two-tier memory subsystems A and B. The access latencies averaged per unit data item for A and B are  $L_A$  and  $L_B$ , respectively. Additionally, denote respectively the storage capacities for A and B as  $S_A$  and  $S_B$ . Moreover, assume the costs per bit for A and B are  $C_A$  and  $C_B$ , respectively. Note that  $C_A > C_B$ . Consider the design that a processor reads (or writes) A if A can deliver requested data items. Otherwise, the processor reads (or writes) B, and supplies A with the missed data items so that the processor can then fetch data from A. If we intend to come up with a cost-effective design in minimizing average memory access latency, please identify true (denoted by ○) or false (×) for the following questions.
  - (a) [5%]  $L_A > L_B$
  - (b) [5%]  $S_A C_A > S_B C_B$
  - (c) [5%] None is true
- 2) [15%] (task scheduling) Recall that a random variable  $X$  follows the exponential distribution of the probability density function  $f(X=x) = \lambda e^{-\lambda x}$ , where is  $\lambda$  a given parameter. Assume that processes in a computer system have their execution time following the exponential distribution. Let two processes A and B have lifetimes represented by the random variables  $X_A$  and  $X_B$  with parameters  $\lambda_A$  and  $\lambda_B$ , respectively. Here,  $\lambda_A = 100 < \lambda_B = 200$ . Both processes activate their execution at time 0. The computer system hosts a single processor. Please identify true (○) or false (×) for the followings.
  - (a) [5%]  $Pr(X_A \geq 101 | X_A \geq 100) = Pr(X_A \geq 111 | X_A \geq 110)$
  - (b) [5%] If we employ the shortest job scheduling, then A shall be scheduled first and then B.
  - (c) [5%] For any scheduling policy, the total expected time to finish both tasks is 300.
- 3) [20%] (file system and synchronization) Consider a disk file  $\mathcal{F}$  that stores a set of key-value pairs. “ $y = READ(x)$ ” API is provided to return the value  $y$  for a designated key  $x$  if the key-value pair  $(x, y)$  is available in  $\mathcal{F}$ . “ $WRITE(x, y)$ ” API is to write the provided  $(x, y)$  into  $\mathcal{F}$  if  $(x, y)$  is absent; otherwise,  $WRITE(x, y)$  updates the key  $x$  with the value  $y$ . Here,  $READ(\cdot)$  and  $WRITE(\cdot)$  are independent processes (or threads). Please identify true (○) or false (×) for the potential designs, as follows, that optimize the read and write operations (in terms of access latency) over the file  $\mathcal{F}$ . Assume  $S$  be the set of key-value pairs stored in  $\mathcal{F}$ .

- (a) [5%] Partition  $S$  into a number of disjoint blocks  $\{s\}$ , i.e.,  $S = \cup\{s\}$ . Each block  $s \in S$  serves as a fetching unit between the memory and disk subsystems. Given  $(x, y) \in s$ , we allocate a memory block  $m$  to store  $s$  copied from the disk. Then, retrieve  $(x, y)$  from  $m$  and deliver the result for  $READ(\cdot)$ . For efficiency, the keys in  $m$  shall be organized in a balanced search tree, for example.
- (b) [5%] For  $WRITE(\cdot)$ , update  $(x, y)$  in  $m$ . When  $m$  is replaced,  $m$  shall be written back to  $\mathcal{F}$  and the original  $s \in \mathcal{F}$  may be overwritten.
- (c) [5%] We shall additionally create a local file index  $\mathcal{I}$  in the file system for locating  $(x, y)$  in  $S$ . That is,  $\mathcal{I}$  helps rapidly identify which of  $s \in S$  storing  $(x, y)$ .  $\mathcal{I}$  shall be replicated in the memory (denoted by  $\mathcal{I}^*$ ) to minimize the number of IO operations performed over the disk subsystem.  $\mathcal{I}^*$  is updated for newly arrivals  $(x, y)$  due to  $WRITE(\cdot)$ . Consequently,  $\mathcal{I}^*$  and  $\mathcal{I}$  may be inconsistent.
- (d) [5%] Possibly, multiple  $WRITE(\cdot)$  operations address the identical key  $x$ . These  $WRITE(\cdot)$  operations may be concurrently invoked by distinct clients. One potential implementation is that each  $WRITE(\cdot)$  successfully acquires the locks for accessing both  $m$  and  $\mathcal{I}^*$  before the  $WRITE(\cdot)$  can proceed to update the value of  $x$ . Once the value is committed, both locks are released. With such an implementation, it is NOT possible to introduce deadlock due to various  $WRITE(\cdot)$  operations.

## Part II: Computer Organization [50%]

- 4) [30%] Please indicate whether the following statements are Correct or Incorrect and provide your explanations to justify your answers.
  - (a) [5%] In the design of a microprocessor, the instruction set architecture of the processor completely hides all physical implementation details.
  - (b) [5%] For a given problem, the speedup achieved by a multicore processor is a critical factor in assessing the processor design. A multicore system exhibits strong scaling when the speedup is achieved without increasing the problem size,  $S$ , where the memory size utilized by each processor core is  $S$ .
  - (c) [5%] In the design of a multi-level processor cache, it is critical to balance between miss rate and miss penalty. The design of the first-level caches focuses on miss rate reduction, so larger cache blocks are desired to reduce the miss rate. The design of the second-level caches puts emphasis on miss penalty improvement to avoid higher memory access latencies.
  - (d) [5%] In the design of a pipelined processor, it is important to allow instructions to take fewer clock cycles in order to increase overall pipeline performance. A common practice is to make ALU instructions take fewer cycles to improve the processor throughput.
  - (e) [5%] In the design of arithmetic logic units, the speed of addition depends highly on the number of bits involved. As the number of bits increases, more 1-bit adders are required to calculate the sum of each bit and it leads to longer delays to obtain the sum for all bits involved.
  - (f) [5%] For the PC-relative addressing in RISC-V processors, the SB-format *branches* have a 12-bit address immediate and the UJ-format *jumps* have a 20-bit address immediate. These formats mean that a RISC-V program can branch within  $\pm 2^{10}$  words of the current instruction and jump within  $\pm 2^{18}$  words of the current instruction, if the PC is used as the register to be added to the address.

5) [20%] Please answer the following questions related to memory locality of matrix multiplication. The following C code having elements within the same row are stored contiguously. The nested loops perform the operations on the floating-point arrays, N and M. Each array element is a 32-bit floating-point number.

```
for (x=0; x<8; x++)
```

```
    for (y=0; y<8000; y++)
```

```
        N[x][y] = M[x][0] + N[x][y];
```

- (a) [5%] Which variables demonstrate temporal locality in their references?
- (b) [5%] Which variables demonstrate spatial locality in their references?
- (c) [5%] If the cache size is unlimited, how many cache blocks, each with 4 words, are required to store all the referenced matrix elements?
- (d) [5%] Assume that the data in the two arrays, N and M, are streaming over Internet (similar to a video streaming application) with predictable data accessing patterns. What technique can be used to bring up adjacent cache blocks into a buffer within the processor core to reduce the data accessing time? Note that when the data is found in the buffer, it is considered as a hit and moved into the cache for the next block. Please provide a detailed explanation of how this technique works.