國立政治大學 107 學年度 碩士班 招生考試試題

考試科目	作業系統	系所別	資訊科學系	考試時間	2 月 3 日(六) 第 2 節

1. (16%) Virtualization and Operation Systems:

   (1) What is the "Trap-and-Emulate"?

   (2) What is the problem when we implement Trap-and-emulate in an old X86-based CPU?

   (3) How does binary translation solve this problem?

   (4) What is Light-weight Virtualization (e.g. Docker) ? What are the benefits of such technology and why?

2. (16%) Please define the following terms

   (1) Microkernel

   (2) Spin lock

   (3) Processor affinity

   (4) EDF Scheduling

3. (8%) Consider the classic bounded-buffer producer-consumer problem (in which the producer 'writes data into a share bounded buffer and the consumer reads data from the same buffer). Assume the size of the bounded buffer is n. Please enhance the following pseudo code by using binary and counting semaphores to ensure that :

   (1) Semaphores are properly initialized,

   (2) there is no buffer overflow and underflow, and

   (3) accesses to the buffer are mutual exclusive.

   Suppose we have an abstract data type **semaphore** that provides three operations: **wait () , signal(), and count()**, where operation **count()** is used for initializing a semaphore. For example, assuming *sem* is a variable of type **semaphore**, **count(5)** will initialize *sem* to the value of 5.

```
// Semaphore declaration and initialization code
…
// Producer
while (true) {
   …
   Add an item to buffer
   …
}
```

```
//Consumer
while (true) {
   …
   Read and remove an item from buffer
   …
}
```

4. (10%) Please “illustrate” and compare the layered structure of Android and iOS platforms with respect to kernel, services and library, and user interfaces.

5. (10%) System protection and security

- Buffer-overflow attacks can be avoided by adopting a better programming methodology or by using special hardware support. Briefly explain how each of these solutions prevents buffer-overflow.
- What is the purpose of using a “salt” along with the user-provided password? Where should the “salt” be stored, and how should it be used?

6. (15%) Consider a computer system with a 32-bit virtual address space where paging is used. Assuming the page size is 4K bytes and the memory is byte-addressable, please answer the following questions:

- How many pages can a process have at most? Suppose the maximum physical memory size is 32 GB. What is the number of bits for physical addresses? What is the maximum number of frames for the system?
- Let the memory access time and TLB access time be 100 ns and 20ns, respectively. If we want an effective memory access time of less than 140ns, what is the minimal TLB hit ratio that needs to be achieved?
- Suppose the virtual memory of the system adopts demand paging. Assume the effective memory access time of the computer system without any page fault is loons, and the service time for a page fault is 15ms. If the page fault rate is 0.0000004, what is the effective access time under demand paging?

7. (10%) A blockchain can be seen as a distributed file system which consists of continuously growing list of records (blocks). The blocks are linked and secured using cryptography. Each block typically contains a hash pointer as a link to a previous block, a timestamp and transaction data. The data stored in a blockchain are inherently resistant to modification of the data. A blockchain network is typically peer-to-peer and therefore is de-centralized. Any new data must be verified by peers before they can be added in to the file system. According to the above descriptions of blockchain, answer the following questions.

考 試 科 目	作業系統	系 所 別	資訊科學系	考 試 時 間	2 月 3 日(六) 第 2 節

(1) From the view point of File System, which “consistency semantics” should the blockchain uses and why?

(2) Assuming that you are going to design a new blockchain. Sketch an approach to store the data (in a distributed way) and make these data being consistent at the same time.

8. (15%) File systems and I/O: Consider DISK scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN, and LOOK),

(1) Which algorithm is best for heavily loaded system? Why?

(2) Which algorithm is truly fail, no starvation? Why?

(3) Which algorithm is best for DISK scheduling when reading data in SSD (Solid-State Disk)? Why?

備 註	一、作答於試題上者，不予計分。 二、試題請隨卷繳交。