

# 國立臺灣師範大學 98 學年度碩士班招生考試試題

科目：計算機系統

適用系所：資訊工程學系

注意：1.本試題共 5 頁，請依序在答案卷上作答，並標明題號，不必抄題。2.答案必須寫在指定作答區內，否則不予計分。

---

1. (12 分，請列計算過程) Consider a program is executed on a computer. In the program, 20% of the instructions are branch instructions, and the other 30% are memory access instructions. The computer has average CPI of 2 for perfect cache and perfect branch prediction. The penalty for each cache miss is 20 cycles, and for each branch prediction miss is 1 cycle. If there are 1% of instruction cache miss, 10% data cache miss, and 10% branch prediction miss, what is the actual CPI?
2. A pipelined datapath has five stages, i.e., instruction fetch, decode, execution, memory, and write back. It has a hazard detection unit. An instruction will be stalled at the decode stage if any of its operands is not ready. Consider executing the following code on this datapath:  

```
lw  $2,10($1)
sub $4,$3,$2
add $5,$6,$7
and $8,$4,$5
or  $9,$8,$3
```

  - (a) (8 分) At the end of the fifth cycle of execution, what instruction will be in each of the pipeline stages?
  - (b) (8 分) If the datapath has a forwarding unit to resolve the data hazard, what instruction will be in each of the pipeline stages at the end of the fifth cycle?
3. (12 分，請列計算過程) A 32-bit address computer has a 4-way set associative cache with 256K bytes of data and 4-word blocks. How many total memory bits are required for the cache?
4. (10 分) What is the common standard for representing real numbers in modern computer systems? Explain the format of this standard.
5. (3 分) What kind of problems can be caused in an operating system if a processor lacks of a hardware-supported dual mode?

6. (3 分) There are two approaches for multiple-processor scheduling in SMP systems. One is that all processes in the ready state are stored in a common queue. The other is that each processor may have its own private queue of ready processes. It seems that the later one tends to have problems in load balancing. However, most of the contemporary operating systems employ the later one. Why?
- 7.(5 分) Illustrate how an operating system handles an invoking of a system call by an application program. Note that you should show all the steps in details.
8. (3 分) Compare the overhead of creating a process with creating a thread in an operating system.
9. Regarding to the System V system calls of Unix, answer the following questions.

(a) (3 分) What is the possible execution result of the following code fragment?

```
main()
{
    int pid;

    printf("Start of test\n");
    pid=fork();
    printf("My process ID is %d\n",pid);
    pid=fork();
    printf("My process ID is %d\n",pid);
}
```

(b) (3 分) What is the possible execution result of the following code fragment? Assume there are only two files, xyz.c and xyz.o, in the current working directory.

```
main()
{

    printf("My program starts to execute\n");

    execlp("ls","ls","-a",NULL);

    printf("Program running here\n");
}
```

(c) (3 分) Write a C program to fulfill the shell command “who|grep John>foo”. Your program should create processes and pipes in the System V system calls of Unix.

10. (a) (2 分) What is the meaning of the safe state regarding to the deadlock avoidance?

(b) (3 分) Assume that we already have a subroutine “IsSafe(S)” which can check if a state is safe. Use this subroutine to implement a deadlock avoidance algorithm. Note that the input parameter S in “IsSafe(S)” represents a resource allocation state of the system. The subroutine “IsSafe(S)” returns “True” if “S” is a safe state. Otherwise, it returns “False”.

Write the pseudo code of your algorithm.

11. Assume that “counter” is a shared variable which can be modified and read by two concurrent processes.

(a) (3 分) Show that the concurrent execution of statements “counter=counter+X” and “counter=counter-Y” may cause an incorrect execution. The correct execution results of the two statements should have the value of “count” become “count+X-Y”. Note that “X” and “Y” are not shared variables.

(b) (3 分) Give your concurrent program which use the *atomic* swap() instruction (shown below) to make sure a correct concurrent execution of “counter=counter+X” and “counter=counter-Y”.

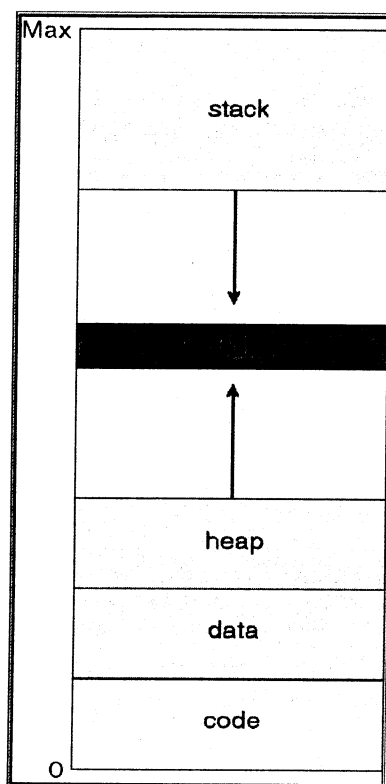
```
void Swap (boolean *a, boolean *b)
{
    boolean temp = *a;
    *a = *b;
    *b = temp;
}
```

12. The following figure shows the structure of a process in memory. A process is a program in execution. A process is more than of the program code, which is sometimes known as the text section. It also includes the current activity, as represented by the value of the program counter and the contents of the processor’s registers. A process generally also includes the process stack,

which contains temporary data (such as function parameters, return addresses, and local variables), and a data section, which contains global variables. A process may also include a heap, which is memory that is dynamically allocated during process run time. Refer to this figure to answer the following questions.

(a) (3 分) Why is the one-level paging insufficient?

(b) (3 分) Explain why two-level paging can solve this problem. Please use a diagram to show the structure of all the used page tables.



13. (5 分) The implementation of the `open()` and `close()` operations is more complicated in an environment where several processes may open the file at the same time. This may occur in a system where several different applications open the same file at the same time. How an operation system can support this kind of concurrent file access? You should consider the following requirements. First, different processes can open the file with different access rights. Second, the system maintains only a single cache for multiple opens of a file from different processes. Third, the system should be able to release the cache when a file has not been opened by any process.

14. (a) (3 分) Please illustrate why the dynamic library linking of routines should be supported by the operating system.

(b) (2 分) Please give an application program in which the dynamic link is more appropriate.