1. Write down *yes* or *no* for the following statements for the relationship of a process and its child process. (10%)
   (a) The child shares the parent's files.
   (b) The parent can modify the values of child's variables.
   (c) They have the same process ID.
   (d) The child can not be executed until the termination of the parent process
   (e) The child can run its own code.

2. (a) The system will enter deadlock if you can not find a safe sequence for it, Yes or NO ? (5%)
   (b) Write down the safe sequence for the following example if it exists: (10%)

	Allocation	Max	Available
	ABCD	ABCD	ABCD
P0	0012	0012	1520
P1	1000	1750	
P2	1354	2356	
P3	0632	0652	
P4	0014	0656	

3. Explain why the time slice should be short from the viewpoint of interactive users while it is not true from the viewpoint of the system utilization. (10%)

4. For determining the minimal number of frames allocated to each process, list the factors as many as you can and shortly explain them.. (10%)

5. Write down the one drawback for contiguous, linked, and indexed file allocation methods respectively. (10%)

6. A solution to the critical-section problem must satisfy the following three requirements: mutual exclusion, progress, and bounded waiting. Answering the following questions briefly. (10%)
   (a) In what situation, we should do the busy waiting for the mutual exclusion.
   (b) A process which just left *exit section* can not participate the decision of entering the critical section next, why ?
   (c) Assume there are *n* processes trying to enter the critical section, how do you

(背面仍有題目,請繼續作答)

722002B

explain the bounded waiting value for a fairness service.

7.Usually, a high-level language compiler would compile a source code into an intermediate form, which may not be open to the public, and generate object codes for some specific target machine. Some languages, such as Pascal, Smalltalk, and Java modify this situation a little. Take Java as an example. A typical Java compiler compiles a Java program into a special object code for a Java Virtual Machine. The object codes, called .class files, could then be distributed across the network. To execute a .class file, one need to install a Java Runtime Environment, which generally requires a loader (to load the .class files) and an interpreter to execute (interpret, or simulate) the Java Virtual Machine instructions. Please answer the following questions:

(1) In general, what are the advantages and disadvantages between a compiler and an interpreter approach? (5%)

(2) In general, what fields are required in an object file? Please list at least three items. (Hint: Do you know the object file format of any operating system?) (5%)

(3) What are the advantages of using a virtual machine, as compared with the usual approach? (5%)

(4) The performance of the virtual machine approach is generally not as good as the direct execution of object codes. Why? (5%)

(5) Please describe an approach that can speed up (or improve the performance of) the interpreter. (5%)

(6) Suppose that some group of researchers think that they would just use Java as a language, just like C. And they write Java compilers which will directly generate object codes for some target machine, without any Java Virtual Machine codes. Do you think this approach is feasible? Why? (5%)

(7) Suppose that another group of researchers think that they would not modify Java compilers but they would translate the .class files into the target machine codes before the .class file are really executed. That is, they will develop a translator to translate the .class files into machine codes and execute the machine codes instead of interpreting the .class files. Do you think this approach is feasible? Can it improve performance? Why? (5%)