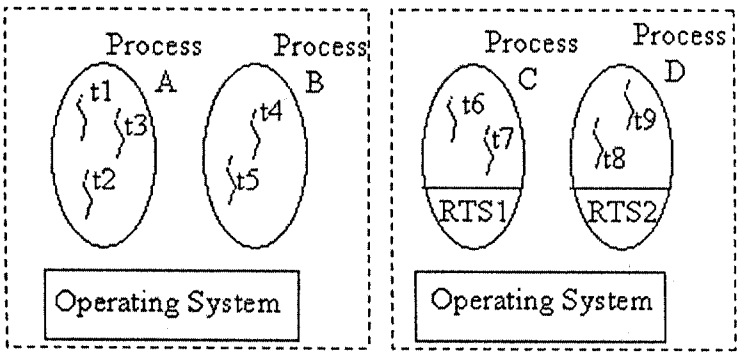
[1] In the following, diagram (a) shows a system that supports threads in operating system and diagram (b) shows a system that does ***not*** support threads in the operating system. Both systems use demand paging in memory management.



(a) Kernel-level thread (b) User-level thread

⟨ represents a thread ( ) represents a process

RTS1, RTS2 represent run time system

(A) When the running thread is going to be switched from t1 to t3 (t1 is running and t3 is going to run next), what would be changed ***by the operating system***? Please answer this question by choosing some items listed in (E). (3%, 答錯一項扣一分)

(B) When the running thread is going to be switched from t6 to t7 (t6 is running and t7 is going to run next), what would be changed ***by the operating system***? Please answer this question by choosing some items listed in (E). (3%, 答錯一項扣一分)

(C) When the running thread is going to be switched from t2 to t5 (t2 is running and t5 is going to run next), what would be changed ***by the operating system***? Please answer this question by choosing some items listed in (E). (3%, 答錯一項扣一分)

(D) When t8 issues a system call for I/O and blocks, is it possible that t9 (assume that t9 and Process C are ready to run) gets to run next? Please explain briefly. (3%)

(E) Items to be chosen for questions (A),(B) and (C)

(1) The contents of general purpose registers
(2) The contents of stack pointer
(3) base address of the page table being used
(4) protection bits in page table entries
(5) the contents of base-limit registers for threads
(6) the contents of program counter

[2] Given the data structures listed below that would be used by an operating system for file service,

(a) a table (called TABLE1 in this question) to keep the status of open files for each process, ***one table per process***

(b) a table (called TABLE2 in this question) to keep the status of open files, ***one table for the whole system***

(背面仍有題目,請繼續作答)

(A) what would be the information stored in TABLE1 and TABLE2 ? (6%)
(B) please draw a diagram to show the relationship between TABLE1 and TABLE2. (4%)
(C) when an open file request is being served, how would these tables be accessed and manipulated ? (4%)

[3] Please insert the following six statements into the appropriate places (totally 8 to be inserted) in the producer-consumer program (shown below) in which the producer and consumer communicates via message buffer synchronized by means of semaphores.

The operations to be inserted are

(a) P(s1); (b) V(s1); (c) P(s2); (d) V(s2); (e) P(s3); (f) V(s3);

Please write the producer code and consumer code on your answer sheet. (8%)
(答案寫在答案紙上)

```
/* ( the operations of P(s) are     s=s-1 ; wait if s<0; )   */
/* (the operations of V(s) are     s=s+1 ; signal if s <=0; )   */
/* define buffer and Initialization */
     message buffer-slots[n];     n is the number of message buffer slots
     semaphore s1 =1;
     semaphore s2 = n;
     semaphore s3=0;

/* Producer code */
while (true) {
(1) Producing a piece of message, M ;
(2)
(3)
(4) Put-message(M);   /* copy message M into message buffer */
(5)
(6)
}

/* Consumer code */
while (true) {
(7)
(8)
(9) Get-message(Buffer);   /* get one message and copy to Buffer */
(10)
(11)
(12) Processing the message in Buffer;
}
```

[4]

(A) Please explain briefly why multiple processor modes (say, user mode and system mode) are required to support protection in a multi-user operating system. (4%)

(B) Please list two instructions or mechanisms that will switch the processor mode from user to system. (4%)

[5] In a demand paging virtual memory system,

(a) how does the system determine the size of the page table of a process ? (4%)

(b) will the size of the page table be changed during the execution of the process? Please explain your answer briefly. (4%)

[6] The following questions are related to compiler design and implementation:

[6-1] Describe the components of BNF grammar. (4%)

[6-2] Define a BNF grammar for arithmetic expressions in postfix form. Your grammar should cover binary and unary arithmetic operations (e.g. $+x$ and $-y$), and you may define your own notation. (5%)

[6-3] Based on your grammar shown in [6-2], describe the left-most derivation for the postfix form of expression: **(A-B)×(-C)+D/E** (5%)

[6-4] In which phase or phases during the process of compiling will the grammar be referenced or used by the compiler (3%)

[6-5] Which part(s) of a compiler has(have) connection or dependence on the design of linking loader in the system. (3%)

[7] For each of the following questions, please pick all proper choices: (12%)

[7-1] Which data structure(s) can be used to implement a comparison-based sorting algorithm?

(A) Stack (B) Priority Queue (C) Doubly linked list

(D) Network (E) Binary tree (F) Hash table

[7-2] Which is(are) true for bubble sort?

(A) unstable sorting algorithm (B) comparison-based sorting algorithm

(C) time complexity $O(n)$ (D) time complexity $\Omega(n)$

(E) space complexity $O(n^2)$ (F) space complexity $\Theta(n)$

[7-3] Which among the following sorting algorithms is(are) suitable for efficient implementation of external sorting method?

(A) Insertion sort (B) Quick sort (C) Heap sort

(D) Merge sort (E) Bucket sort (F) Radix sort

[7-4] Which among the following is(are) true?

(A) AVL trees are binary search trees

(B) The height of an AVL tree of N nodes is $O(\log N)$

(C) A complete binary tree is not always height-balanced

(D) The height of a binary tree of N nodes is $\Omega(\log N)$

(E) A B-Tree of order M is a height-balanced search tree

(背面仍有題目,請繼續作答)

[8] Given an undirected graph ***G***, how would you represent ***G*** in a computer program so that it can be efficient to find the shortest length of a path that is also a cycle among all the simple paths of ***G***:

[8-1] Describe your representation of ***G*** and give your reason. (5%)

[8-2] Show your algorithm (in any algorithmic language or programming language) to find the shortest length among all the cycles. (8%)

[8-3] What is the time complexity of your algorithm given in [8-2] ? (5%)