國立臺北科技大學

九十二學年度資訊工程系碩士班入學考試

作業系統試題

填准考證號碼　　　　　　　　　　　　　　　　　第一頁　共二頁

注意事項：

1. 本試題共[redacted]題，配分共 100 分。
2. 請按順序標明題號作答，不必抄題。
3. 全部答案均須答在答案卷之答案欄內，否則不予計分。

1. Short Answers. (6% each)

   (a) What are the differences between user threads and kernel threads?

   (b) Define the following: (1) critical sections, (2) race conditions, and (3) mutual exclusion.

   (c) What are the differences between deadlock and starvation?

   (d) Describe the MS-DOS FAT and the UNIX inode disk-space allocation methods.

   (e) What problem introduced by the SCAN disk-scheduling policy can be avoided by using the C-SCAN disk-scheduling policy?

2. Consider the following set of processes, with the length of CPU burst time and arrival time given in milliseconds:

Process	Arrival time	Burst time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

(a) Draw a Gantt chart and determine the average waiting time for FCFS (First Come First Served) scheduling algorithm. (3%)

(b) Draw a Gantt chart and determine the average waiting time for RR (Round Robin) scheduling algorithm with a quantum time of 4 units. (3%)

(c) Draw a Gantt chart and determine the average waiting time for preemptive Shortest-Job-First (i.e., Shortest Remaining Time First) scheduling algorithm. (3%)

(d) What is the average turnaround time for the RR scheduling algorithm in part (b)? Can the turnaround time be improved if we increase the time-quantum size in part (b)? Justify your answer. (6%)

3. Consider a system with five processes P_0 through P_4 and two resource types A and B. Suppose that, at time T_0, the following snapshot of the system has been taken, where the Allocation matrix lists the number of resources allocated for each process, and the Max-Request matrix lists the maximum number of resources each process requests. Moreover, the number of available resources for types A and B at T_0 is 3 and X, respectively.

	Allocation		Max-Request		Available	
	A	B	A	B	A	B
P_0	2	0	2	5	3	X
P_1	5	2	10	2		
P_2	0	4	5	4		
P_3	1	1	4	1		
P_4	0	0	5	9		

(a) Suppose X=8, is the system state safe? If yes, provide a safe sequence for executing all processes. If not, explain it. (5%)

(b) What is the smallest value of X for which the system state is safe? Provide a safe sequence for executing all processes with the smallest value of X (In order to get a full credit, you must show how the value of X is found). (10%)

注意：背面尚有試題

4. A process has four page frames allocated to it. The time of the last loading of a page into each page frame, the time of the last access to the page in each page frame, the virtual page number in each page frame, and the referenced (R) and the modified (M) bits for each page are shown in the following table (the times are in clock ticks from the process start at time 0 to the event — not the number of ticks since the event to the present).

Virtual page number	Page frame number	Time loaded	Time referenced	R bit	M bit
2	0	60	161	0	1
1	1	130	160	0	0
0	2	26	162	1	0
3	3	20	163	1	1

Assume a page fault to virtual page 4 has occurred. Which page frame will have its contents replaced for each of the following memory management policies? Explain why in each case.

(a) FIFO (first-in-first-out) (3%)
(b) LRU (least recently used) (3%)
(c) Clock (3%)
(d) Given the above-mentioned state of memory just before the page fault, consider the following virtual page reference string:

4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2

How many page faults would occur if the working set policy were used with a window size of 4 instead of a fixed allocation? Show clearly when each page fault would occur. (6%)

5. Consider a system which is thrashing. Which of the following actions (if any) the system can take are likely to help? Explain your answer. (10%)

a. Increase the scheduling quantum of the process with the highest page fault frequency.
b. Increase the working set size of the process with the highest page fault frequency.
c. Increase the degree of multiprogramming.
d. Swap a process out.
e. Increase the size of paging disk.

6. Consider the following Savings Account problem.

> A savings account is shared by several people (processes). Each person may deposit or withdraw funds from the account. The current balance in the account is the sum of all deposits to date subtracts the sum of all withdrawals to date. The balance must never become negative, and withdrawals are serviced First-Come-First-Serve. For example, suppose the current balance is \$100, and customer A is waiting to withdraw \$300. Assume another customer, say B, arrives with the request of withdrawing \$200. Customer B must wait until customer A is serviced.

The following monitor is proposed to solve the Savings Account problem with two procedures, *Withdraw*(Amount) and *Deposit*(Amount), for customers to withdraw and to deposit funds, respectively.

```
type Account = monitor
var Balance : real, NeededAmt : real, Count : integer, MayWithdraw: condition;

Procedure entry Withdraw(var WithdrawAmt: real)
begin
    Count := Count + 1;
    if (NeededAmt > 0) then
      MayWithdraw.wait;
    else if (Balance < WithdrawAmt) then
    begin
        NeededAmt := WithdrawAmt - Balance;
        MayWithdraw.wait
    end;
    Balance := Balance - WithdrawAmt;
    Count := Count - 1;
    if Count > 0 then MayWithdraw.signal;
end;

Procedure entry Deposit(var DepositAmt: real)
begin
    Balance := Balance + DepositAmt;
    if (NeededAmt > 0) then
    begin
        if (DepositAmt >= NeededAmt) then
        begin
          NeededAmt := 0;
          MayWithdraw.signal
        end;
    end;
end;

begin
     Balance := ...;   // initial balance
     NeededAmt := 0.0;
     Count := 0;
end. {Initializations}
```

(a) Explain why the monitor does not work; and, give a scenario that clearly demonstrates the problem. (5%)

(b) Revise the monitor to solve the Savings Account problem. (10%)