[1] "Circular wait" is one of the necessary conditions to lead to a deadlock in a concurrent system. To break the "circular wait", how can we do ? (5%)
Although deadlocks may rarely occur in a multitask operating system, they are very difficult to be prevented in practice, why ? (9%)

[2] For the following strategies, which are effective in reducing the page fault rate in a virtual memory system ? (10%, 每答錯一題扣兩分)
(A) Increase the real memory size of a computer
(B) Increase the virtual memory size allocated to a process
(C) Using a faster paging disk
(D) Using multiprocessors instead of a single processor
(E) Using a faster network
(F) Access a matrix in row major order in a FORTRAN program
(G) Limit the degree of multiprogramming
(H) Using file system cache for file access

[3] If we don't shut down a computer via the normal procedure, (say, by directly powering down the computer) the data on the disks may become inconsistent. Please explore the reasons in some details. (12%)

[4] Please choose the best answer from the items attached below for each of the following questions. (14%, 每答錯一題扣兩分)
(1) A (or An) _____ is the module that gives control of the CPU to the process selected by the short-term scheduler.
(2) A (or An) _____ is a record used to store most of the information about a specific file on the disk in Unix.
(3) _____ is a technique by which a program module in one address space can call another program module in another address space.
(4) The _____ is an instruction which is usually used for the efficient implementation of mutual exclusion mechanism in a multiprocessor system.
(5) _____ is a technique to simulate a dedicated and low speed device with a shared and high speed device.
(6) _____ is (or are) the important hardware support for the implementation of page replacement algorithm in a demand paging system.
(7) A _____ in Unix is a software mechanism that informs a process of the occurrence of asynchronous events.

Possible answers:

| | | |
|---|---|---|
| (A) privileged instruction | (B) Enable/Disable interrupt | (C) system call |
| (D) test-and-set instruction | (E) reference bit | (F) modify bit |
| (G) valid bit | (H) reference bit and modify bit | (I) page fault |
| (J) long-term scheduler | (K) medium-term scheduler | (L) signal |
| (M) dispatcher | (N) critical region | (O) mutual exclusion |
| (P) I/O redirection | (Q) pipe | (R) inode |
| (S) virtual disk | (T) directory | (U) file descriptor |
| (V) IPC | (W) far call | (X) RPC |
| (Y) message passing | (Z) Spooling | |

(5) For each of the following statements, please indicate TRUE or FALSE and briefly explain your reason. (30%)

(5-a) It is possible to implement a one-pass compiler but it is impossible to implement a one-pass assembler.

(5-b) A compiler must do code optimization.

(5-c) Shift-Shift conflicts may happen during the process of LR parsing.

(5-d) Every LR(1) grammar is also an SLR(1) grammar.

(5-e) Shift-Reduce conflict may happen during the process of LL parsing.

(5-f) A compiler does not generate any object code for the following program:
    main ( ) {   }

(5-g) The macro expansion and the generation of object code are done in the same pass by assembler.

(5-h) Assembler does not do code optimization.

(5-i) The linking operations must be done before a program is loaded for execution.

(5-j) Relocation bit is set by loader program during the loading process.

(6) Sketch the binary tree which assumes the outcomes of in-order traversal and post-order traversal as listed below:     (6%)
In-order:    A B C D E F G H J K
Post-order: A C E D B J H K G F

(7) Let $f(n) = \sum_{i=1}^{n} \log i$ , show that $f(n) = O(n \cdot \log n)$ .     (6%)

(8) When implementing a circular queue using an array, we reserve $n$ storage units as the components of the circular queue. Is there any advantage if we choose a number of $(2^m - 1)$, $m > 1$, as the value of $n$ ? Does the same apply to determining the number of entries in a hash table? Please give your reason.   (8%)