Operating Systems- Term Exam, Chapter 7 and Chapter 8, Jan. 3, 2002

1. Why kind of mechanisms is required to ensure the orderly execution of cooperating processes? (5%)

2. (1)What is an atomic instruction?(2%) (2)How to ensure an instruction to be atomic?(5%)(3)What is an atomic transaction? (3%)(4)How to ensure an atomic transaction to be atomic?(5%)

3. What is a race condition? (2%)Can you give an example ?(3%)

4. (1)What is 'busy waiting'? (2%) (2)What is spin-lock (or spinlock)?(3%) (3)Why it brings overhead into the system, especially in multiprocessor systems?(5%)

5. What is a semaphore?(2%)Please write down its wait and signal operation.(8%)

6. (1)What is mutual exclusion?(3%) (2)What is a 'mutex'?(2%) (3)Can you show an example on how a mutex is used as a semaphore to implement a critical section?(5%)

7. (1)What is 'deadlock'? (2%)(2)What is 'starvation'? (2%)(3)Can you show an example for them separately?(6%)

8. What is a 'critical region'?(5%) Can you show an example that uses critical region?(5%)

9. (1)What is a monitor? (5%) (2)Can you show the structure of a monitor (in a pseudo code (or programming language like) style)? (5%)(3)How to declare and use a conditional variable in a monitor?(5%)

10. How can a system recover from a deadlocked state, if it is detected?(5%)

11. Given 5 processes P_0 through P_4; 3 resource types A (10 instances), B (5instances, and C (7 instances). Suppse we have a snapshot at time T_0:

	Allocation	Max	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

Can request for (3,3,0) by P_4 be granted? Why or why not?(10%)

atomic instruction : an instruction that completes in its entirety without interruption.

race condition : 數個 processes 存資料 and 操作，且結果取決於次序

critical section : 一段不能被多個 process 同時執行的程式碼（需等待進行：有限等待 busy waiting：此種等待，使用 busy waiting 的號誌也叫 spinlock（執行迴圈等待）

critical region (region v when B do S)：把問題描述在 critical region 的

變數 v 在B條件下執行 S 敘述 (ex 生產者和消費者)

的臨界區域

=> 利用 mutex，first delay（若B為false，在此等待）

Ïhý uú

2007 考古題 <子負責任解答>

▲1. ?

▲2. (1) atomic instruction → 一個在執行中不可被中斷的指令.

(2) ① test memory word & set value.

② swap contents of two memory words.

(3) atomic transaction → 一組在執行間不可被中斷的單一邏輯函數指令.

(4) ① log-based recovery

② checkpoints.

▲3. (1) race condition → 多個行程同時存取或處理同筆資料時，執行的結果取決於存取的順序.

(2) EX: 設 counter 初值為 5，有兩個 process P_1、P_2

① 分別處理 count
```
register = count
register = register+1
count = register
---
register = count
register = register-1
count = register
```
⇒ count == 5 (○)

② 並行處理 count
```
register1 = count
register1 = register1+1
register2 = count
register2 = register2-1
count = register1
count = register2
```
⇒ count = 4 or 6 (×)

▲4. (1) busy waiting → 當一個行程在其臨界區間時，其餘行程在進入程式時會形成迴路.

(2) spin lock → 在 busy waiting 時，行程等待鎖住的同時一直盤旋著.

(3) spin lock 好處是一行程等待不需做 context switch，而 context switch 耗時，故希望鎖住短時間，所以 spin lock 在 multiprocessor system 很有用.

▲5. (1) semaphore ⇒ 用來處理多行程間臨界區間問題的一種同步工具.

(2)
```
wait(S){
  while S<=0
  S--;
}

signal(S){
  S++;
}
```

▲6. (1) mutual exclusion → 當一行程在臨界區間內，不允許其他程式進入其區間.

(2) mutex → 表示互斥的一種 semaphore.

(3) Ex:
```
do {
  wait(mutex)
    臨界區間
  signal(mutex)
    剩餘區間
}while(1)
```

▲7. (1) deadlock → 多個行程無窮地等待一項僅能由等待行程所引發事件的情形.

(2) starvation → 當一個行程無法移出一個所需等待的 semaphore queue 的情形.

(3) Ex:

P_0	P_1
wait(A)	wait(B)
wait(B)	wait(A)

⇒ deadlock.

▲8. critical region → 每個行程皆有一段此區段，在這段程式中行程可能會改變共同變數、更新表格等工作.

Ex:
```
while(true){
  flag[i]=true
  turn = j
  while(flag[j]&&turn==j)
  flag[i]=false
}
```

▲9. monitor → 高階同步架構，可確保在同一時間內只有一個行程在監督程式內活動.

Structure:
```
monitor monitor A
{
  procedure P1(...)
     :
  procedure Pn(...)
  initialization code(...)
}
```

▲10 ① aborted all deadlocked process.
② aborted one process at a time.

▲11.

	Allocation ABC	Max ABC	Need ABC	Available ABC
P_0	010	753	743	332
P_1	200	322	122	
P_2	302	902	600	
P_3	211	222	011	
P_4	002	433	431	

(3,3,0) by P_4?

332 − 330 = 002 → 無法配給剩餘任一 process

故 request 不可!!