transaction.
執行單一邏輯函數的指令 or operation，其執行順序稱 schedule.

Operating Systems, Second Term Exam. Chapter 5 ~ 8. Dec. 13, 2004

所有變更都成功地完成
or 完全沒有更改發生.

兩個 transaction 並行
執行 = 以某種順序依序執行

1. True or false: (10%)

(1)_____ The specification for the JVM(Java Virtual Machine) indicates very clearly that Java threads are to be mapped to the kernel threads of the underlying operating systems.

(2)_____All the threads of the same multi-threaded process can share the same stack.

(3)__F__ Using multithreading will always give improved performance over a single-threaded version. busy waiting

(4)__T__SJF scheduling algorithm is provably optimal, in that it gives the minimum average waiting time for a given set of processes.

(5)__F__ Suppose that each resource may have multiple instances. If an operating system finds that there is a cycle in the resource-allocation graph, there must be a deadlock.

LWP: lightweight process.
輕量級行程, 即 thread.
first

2. Please the full name of the following abbreviations: (10%)

(1) FIFO, (2) RR (3) FCFS, (4)SJF (5) LWP (hint: the last word is 'process')

First in First out, Round-Robin, First come first served, Shortest Job First

3. Please make brief explanations for the following: (30%)

CPU排程，從準備佇列選擇行程，將CPU分配

(1) short term scheduler, (2) preemptive scheduling, (3) turnaround time, (4) starvation, (5) time slice, (6) load sharing, (7) atomic instruction, (8) spinlock, (9) thread safe, (10) deadlock prevention

eadlock prevention.
利用消除死結條件
會降低資源使用率)

4. Please answer the following questions briefly. (20%)

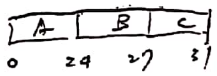
(1) List at least two the major two differences between a kernel thread and a user-level thread.

(2) Show an example that requires 'critical section'.

(3) Show an example of 'deadlock'.

(4) Show an example of a "Gantt chart".

(5) Show an example program that CPU-bursts and I/O bursts.



eadlock avoidance.
避免產生死結.

afe state.
安全狀態, 指存在
程順序, 可讓系統
依順序把資源
配置, 而不死結
為安全序列)

5. What is called 'priority inversion'? Please give an example. How to solve the priority inversion problem? Please use the same example to show your solution. (10%)

數個行程在臨界區執行
→ 競爭條件

6. (1)Please write down the P() (or wait) and V() (or signal) operations for semaphores. Consider that a set of processes use a semaphore to synchronize.

(2)Suppose that a process interchanges the order in which P() and V() operations on the semaphore *mutex* are executed, resulting in the following execution:

```
mutex.V();
criticalSection();
mutex.P();
```

eadlock.
等待的行程其所要求的資源被其他行程所持有而無限期等待. (循環等待不成立)
當死結被偵測到 { 終止行程.
回收資源
(禁止搶先不成立)

Monitor: 高階的同步工具，由變數宣告、函式組成，能保證只有一行程在監督程式中執行，程式設計者不需指令.

```
monitor monitor_name.
{ procedure body P1 ( ) { }
  procedure body P2 ( ) { } }
```

Ïhý uú

What may happen?

(3) Suppose that a process replaces mutex.V() with mutex.P(). That is, it executes

```
        mutex.P()→ wait
        criticalSection();
        mutex.P();          發生死結.
```

What may happen ? (15%)

7. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free. (10%)

回覆時間 turnaround time.
執行開始到結束
waiting time
response time.

Ïhý uú