

評分規則_公告

1. (15%)

- Draw the process parent-child relationship graph resulting and explain.
 - 流程關係有四個區塊(3%、4%、4%、4%)，某一區塊的流程加解釋對，就給部分分。

2. (10%)

- System call : 要求system service (3%)
- UNIX signals : 通知user process 某特定event發生 (3%)
- difference:
 - System call: User-> kernel(2%)
 - UNIX signals : kernel to user, 或user process 1 to kernel, and then to user process2(2%)

3. (15%)

a. (10%)

- Local var. (3%)
(local 只要敘述正確大多有分！)
 - only visible in the function where the variable declares different copies for different function invocations (Of course, different copies for different threads).
- Global var. (3%)
(Global 只要敘述正確大多有分！)
 - shared by all threads (in all functions)
- Thread-specific data (4%)
(要看得出有明顯知道這是什麼才有分，否則斟酌扣分！)
(case 1: 直接寫明 類似只適用於某thread的 global variable.)
(case 2: 直接寫明 共享於該thread 的function及他所call的function，且thread之間不共享)
(case 3: 只有寫 只適用於該thread的data，但(b)要選這選項，我才知道你懂；否則扣2分。)
 - private to a thread, not shared among threads. For the thread, shared between different function invocations.

b. (5%)

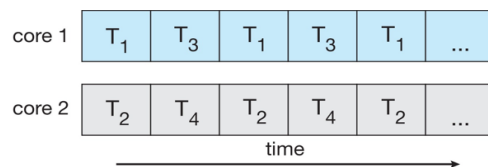
(不用解釋，多寫少寫一個扣3%，最多扣5%)

- global var. and thread-specific data

4. (10%)

參考答案與配分:

- Parallelism (3%) :
 - Parallelism implies a system can perform more than one task simultaneously
 - 能 **同一時間 執行** 二個或以上的 task/job。必定至少有二個或以上的 core 才可能做到 parallelism。
- Concurrency (3%) :
 - Concurrency supports more than one task making progress, scheduler can provide concurrency even on a single processor/core.
 - 能讓二個或以上的 task/job，在 **一段時間內 一起完成/有進展**。即便只有一個 core 也能做到 concurrency (需要 scheduler 或類似的功能來協助)。
- Co-exist (1%) :
 - Yes (只有寫對 Yes 得1分)
- Why or why not (3%):
 - 如果同學畫圖，就要畫一張至少有二個 core 的排程圖:
如下圖, T1&T3間是concurrent, 可同時run T1&T2 是parallel



- 題目沒有要求畫圖，同學亦可用文字描述，但內容觀念要對

5. (10%)

- 5 個選項各 1 分 (5%)
- 解釋 (5%)
 - keyword: cancellation point
- 如果逐項解釋則改成每個選項 2 分，選對 1 分，解釋對再 1 分
- 如果解釋完全錯誤，則視為完全不理解，本題不給分

6. (10%)

- Please describe why a medium term scheduler should consider whether processes are CPU-bound or IO-bound when it swaps in/out the processes.
- 有說明到維持CPU-Bound 與 I/O-Bound jobs 混合比例，未說明到高使用率、可降低**CPU idle** 或 **device idle** 之相關詞語 給 7 分
- 答案：
 - 主要是讓CPU and IO devices盡量都能夠維持在高使用率
 - 維持一個適合的process mix (若干CPU bound processes搭配若干IO bound processes)，這樣可降低CPU idle 或device idle

7. (10%)

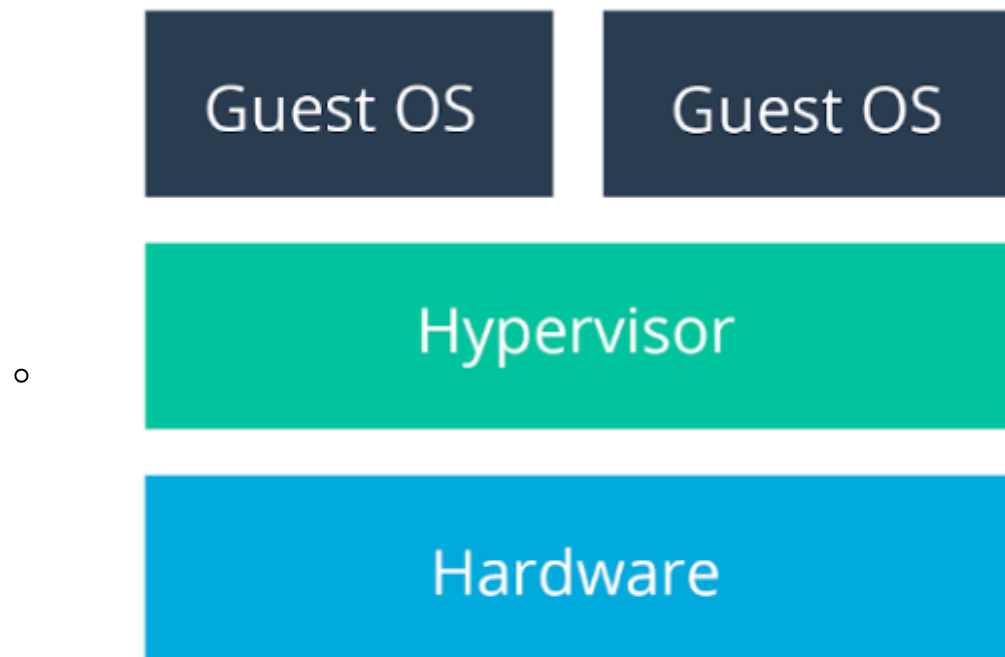
(10%) About virtualization and hypervisors

(a) (5%) What's the main difference between Type I and Type II hypervisors?

(b) (5%) Why virtualization is typically faster than emulation?

a. (5%)

- 差異性 (錯一個扣3分)
 - 寫反扣2
 - Type 1 主要提到運作在hardware 上 可用圖表示



TYPE 1 HYPERVISOR

- Type 2 運作在已存在os上 圖多一個host os
b. (5%)
- 只解釋virtualization or 解釋不完全 +2
- 有提到模擬硬體架構 可加3
- 舉例說明正確或提到instruction translation 加5

8. (10%)

基本配分規則

- case 1: simple 或 module 都沒寫或解釋全錯 => 0 分
 - 刪去法且未解釋最後的答案視為此類
- case 2: 至少有 simple 或 module => 扣分制，每多一個或少一個就 -3
 - 二對一錯 => 7 分
 - 二對二錯 => 4 分
 - 一對二錯 => 1 分 (缺一個答案+錯兩個)

解釋部份詳細配分

本題關鍵在於根據 scheduler、filesystem、device driver 等 subsystem 間的關係判斷 Linux 所屬類型，因此解釋應與以下相近：

- Simple Structure:
 - 重點:
 - 除了 scheduler、memory management 等必須功能之外，Linux 的 file system、device driver 等功能也屬於 kernel，因此符合 simple structure 的概念
 - a large number of functions in one level
 - 未提到重點 => 斟酌扣分：
 - monolithic 的意思與 simple structure 的概念擦邊 => -1
 - Linux 繼承了 Unix 的特性，因此是 simple => -2
- Modules:
 - 重點：
 - 將 Kernel 中的 file system, device driver 等子系統模組化管理（或依照功能分割成模組），並透過 interface 讓 module 能夠被動態加載 (dynamically loadable) 來擴充功能
 - 未提到重點 => 斟酌扣分：
 - 只寫「可以被模組化」，但沒解釋什麼東西可以被模組化 -2

9. (10%)

- Please briefly describe how an interrupt vector table (IVT) can be used when an interrupt occurs.
- 與參考答案意思相似才給分(10%)
- 沒寫到、寫對interrupt service routine(ISR)這個詞，但其餘意思一樣(4%) *唯一keyword*
 - 插入或修正為interrupt service routine，意思與參考解答一致
- 沒有提到interrupt vector(table)的內容(ISR 位置或指令)(-2%) 不完整
- 參考答案:
 - interrupt handler 也可稱為 ISR
 - **case1**: Interrupt通過interrupt vector(table)傳送控制給interrupt service routine (ISR)
 - interrupt vector 包含所有interrupt service routines的位置或指令(沒寫-2%)
 - **case2**:當 interrupt發生，CPU可以通過interrupt vector table (IVT)或跳至IVT對應的 entry，取得ISR位置。