

①

1.14 What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose? ②

① Interrupts 的目的在於中斷目前 CPU 的工作並讓 CPU 即時處理其他事件。

② Trap 算是一種特異化的 Interrupt。
通常是以錯誤或使用者請求引起。

Trap 是一種主要為軟體生成 (software-generated) 的 Interrupt。

③ 可以，使用者可以在自己的 program 內部呼叫 system call 或是進行一些非法操作 (除以 0, 非法記憶體操作) 觸發 Trap。

④ 使用者觸發 Trap 目的：

① 呼叫 System Call 去於 Operation System routines 進行互動。

② 處理使用者程式的算術錯誤。

2.15 What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches? ①

- ① Two models : ① message-passing model
② shared-memory model

②

	message-passing	shared-memory
Strengths	① 比起 shared-memory 更好實作。 ② 在較小資料量交換時十分方便，因為沒有衝突產生。	① 由於以記憶體交換資料，得以最大速度及方便性進行。
Weaknesses	① 速度上較 shared-memory model 慢	① 在 protection 及 synchronization (同步) 有些問題需要處理。

2.19 ① What is the main advantage of the microkernel approach to system design? ② How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach? ③

- ① Microkernel 好處是降低擴展作業系統及修改核心的成本，由於盡可能的移除非必要組件，使得 kernel 較簡單。
(也同時提高可靠性及安全性)
容易移植 OS 到新架構。

- ② 在架構中，Program 與 System service 在 user space 中進行互動，以 message passing 的模式溝通。

- ③ 由於需要在兩個服務溝通，訊息會被複製，且系統需要切換不同的 process 以交換訊息，複製訊息及切換 process 的開銷皆為 microkernel 的缺點。

3.14 Give an example of a situation in which ordinary pipes are more suitable than named pipes and an example of a situation in which named pipes are more suitable than ordinary pipes.

Pipe 的設計考慮四個問題。

- ① 雙向 or 單向通訊?
- ② 若雙向，資料只能單向 or 雙向傳輸?
- ③ 通訊中的 processes 是否要有某種關係?
- ④ Pipe 是否能使用網路。

Ordinary Pipe

- ① 要有父子關係的進程
- ② 只能用在同一個機器
- ③ 只允許單向通訊
- ④ 通訊後消失
- ⑤ 一對 processes

比 named pipe 適用情況

- ① 不要 pipe 可存取時
- ② 在同一個機器內通訊

Named Pipe

- ① 不需要父子關係的進程
- ② 允許不同機器，需用 Sockets
- ③ 可以雙向通訊
- ④ 通訊後存在
- ⑤ 多個 processes

比 ordinary pipe 適用情況

- ① 不同機器通訊
- ② 不需要父子關係的通訊

- 3.17 What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.
- Synchronous and asynchronous communication
 - Automatic and explicit buffering
 - Send by copy and send by reference
 - Fixed-sized and variable-sized messages

a. Synchronous and Asynchronous Communication⁶⁴

Synchronous: 優點是可以讓發送者與接受者在 rendezvous (會合點) 進行，確保雙方可以同時進行訊息的傳送與接收，同步通訊的阻塞傳送有時並非必要，可能浪費資源。⁶⁴

Asynchronous: 優點是程式設計師可以讓兩個進程不會阻塞有更好的效能表現，缺點是較難設計，需確保訊息及時傳達給接收端、系統層需額外設計核心層的緩衝區以支援訊息暫存等問題。⁶⁴

b. Automatic and explicit buffering⁶⁴

Automatic: 優點是提供無限長度佇列，發送者不會因無可用緩衝空間而阻塞。缺點是系統可能預留過大記憶體，導致資源浪費。⁶⁴

Explicit: 優點是指定緩衝區大小，有效管理記憶體使用。缺點是緩衝區滿時發送者需等待，可能造成延遲。⁶⁴

c. Send by copy and send by reference⁶⁴

Send by copy: 優點是保護資料，接收端無法改動參數狀態，適合網路通訊。⁶⁴

Send by reference: 優點是對大型資料結構更有效率，不需大量複製，且便於分散式系統的開發。缺點是涉及資料同步與一致性問題，設計難度較高。⁶⁴

d. Fixed-sized and variable-sized messages⁶⁴

Fixed-sized: 優點是訊息易於實作與管理，緩衝區容量明確。缺點是訊息對程式設計師而言較不靈活。⁶⁴

Variable-sized: 優點對程式設計師友善，適用於多樣資料長度。缺點是需核心層動態管理緩衝，設計較複雜。⁶⁴