HW1 2025/03/23(日) p1.14 p2.15、p2.19 p3.14、p3.17 111590004 張意昌 資工三

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?

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

① Microkernel 好處是降低擴展作業系統及修改核心自內成本，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 是雙向 or 單向通訊？

② if 是雙向，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.

a. Synchronous and asynchronous communication

b. Automatic and explicit buffering

c. Send by copy and send by reference

d. Fixed-sized and variable-sized messages

a. Synchronous and Asynchronous Communication

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，缺點是同步通訊的阻塞傳送有時並非必要，可能浪費資源。

對於程式設計師而言，優點是好理解與除錯，確保一致性，缺點是會造成程式在執行等待時降低效能。

Asynchronous:

優點：發送方不用等待接收方，提高並行及效率。缺點是需要設計額外的機制處理未處理的消息。

對於程式設計師，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。

b. Automatic and explicit buffering

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。

對於程式設計而言，雖然降低難度，但可能會有無法預測的行為。

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。

對於程式設計而言，可以更精細的控制資料，但是要手動管理 buffer 可能會出現錯誤。

c. Send by copy and send by reference

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。缺點是需要額外的記憶體儲存副本，處理大數據時較會浪費效能。

對於程式設計而言，變數較為安全且不影響原始資料，但在處理大型資料時較為浪費效能。

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。

對於程式設計師而言，設計時可以直接存取原始資料，但可能會發生預期外的修改，導致問題。

d. Fixed-sized and variable-sized messages

Fixed-sized:

優點是訊息易於實作與管理，緩衝區容量明確。缺點是訊息為固定值，無法適應不同變量的要求。

對於程式設計師設計上較為方便，不須考慮緩衝，但對於大於長度之訊息較不靈活。

Variable-sized:

優點是適應不同大小的訊息，方便提高記憶體存取效率。缺點是需要設計額外的機制來管理與回收記憶體，增加系統複雜度。

對於程式設計師友善，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。