1118200g 黃曉筠 電資三 OS HW1

1.14

① Interrupt：

A component 發送一個 event發生的訊號給B component

要求B 馬上根據 event 採取行動 → 快速處理緊急任務，不需 polling

② Traps：

由軟體產生（CPU內部指令）

處理 System Call、Exception 等特定情況

Synchronous

Interrupt：

外部硬體、timer

處理 events

Asychronous

③ Yes，

Java -> exceptions

C -> singals

④ Trap 用來通知 CPU 軟體異常事件發生

CPU 可以中斷 program 或跳入其他 program（System call、Debugging、Exception Testing）

2.15

① Shared memory model & Message passing model

②

	Shared memory	Message passing
pros	fast，不同process 透過讀寫記憶體溝通	less synchronization
cons	需要同步方法 分散式系統不能用	更多開銷（發送、接收、傳遞訊息時）

2.19

① Secure（在 user mode 完成的操作比 kernal mode 多）

Stable（簡單核心設計，user mode crash 不影響 kernal）

Easier to extend and maintain（新增 service 不需要更動核心）

② 1. user requests a service.

2. program sends IPC message to relevant system service.

3. Mirrokernal route message to user-space service.

4. The service process request and return another IPC.

Disadvantages

- Performance overhead (訊息有延遲且降低效能)
- Complexity ↑
- Slower System Calls

.14

Use ordinary pipes:

同一台機器有 A、B process 要建立通信

1. named pipes 更多 overhead
2. 不需要存取 A、B 通訊完的 pipes

ex Connect ls & grep

Use named pipes:

雙向、無父子關係、以網路通訊、需要 persistent communication ...

ex Logging

3.17

Programmer level : P

System level : S

(a) Synchronous Communication

S: √ 确定性高 X 低效

P: √ 易理解 X deadlock

Asynchronous Communication

S: √ 高效 X 复杂度增加

P: √ 程式灵活 X 问题调试困难

(b) Automatic Buffering

S: √简化实现 X 资源消耗大

P: √降低开发难度 X 缓衝区不透明

Explicit Buffering

S: √ 资源管理高效 X 复杂性提高

P: √ 性能优化大 X 复杂度提升

(c) Send by Copy

S: √安全性↑ X 性能耗费大

P: √易维护(避免共享问题) X 效率低

Send by Reference

S: √高效 X 安全性↓

P: √ 大数据共享 X 需要同步机制

(d) Fixed-Sized Messages

S: √管理简单 X 空间浪费

P: √ 格式统一 X 灵活性低

Variable-Sized Messages

S: √ 节省空间 X 管理复杂

P: √ 灵活性高 X 处理复杂