

第一次自幹作業系統核心 就上手！

Jim Huang (黃敬群) <jserv@0xlab.org>

Developer, 0xlab / JuluOSDev - <http://www.juluos.org/>

Aug 19, 2012 / COSCUP

作為 LXDE/PCManFM 的開發者，我
也好想去聽隔壁的議程

~>_<~



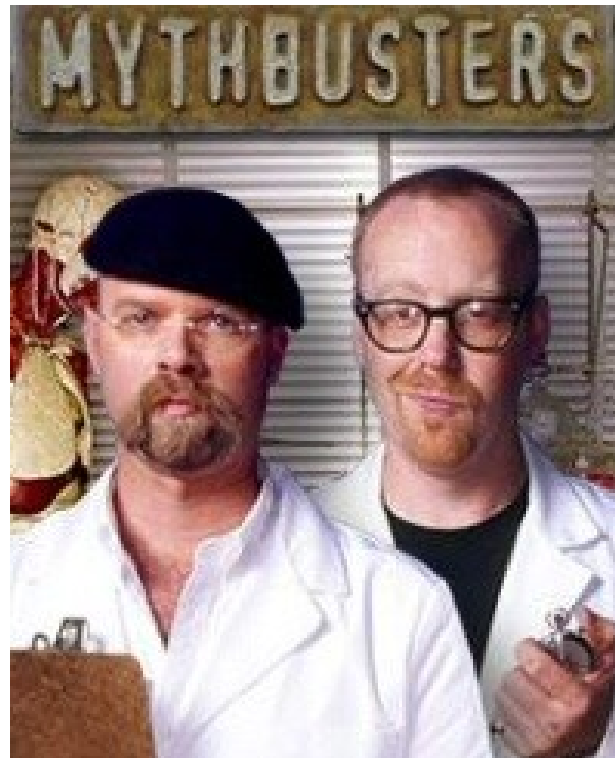
「是個正常人，做正常人能做的事情」

PCMan 自介 (2006)

<http://www.openfoundry.org/en/os-people/629>



設計與實做一套新的作業系統（核心）絕對是正常人能作的事情，而且對生活有益



Agenda

- (1) 開發 OS Kernel 的動機
- (2) 以微核心來建構 OS Kernel
- (3) 立馬體驗

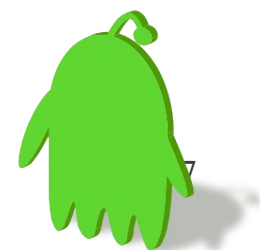


開發 OS Kernel 的動機

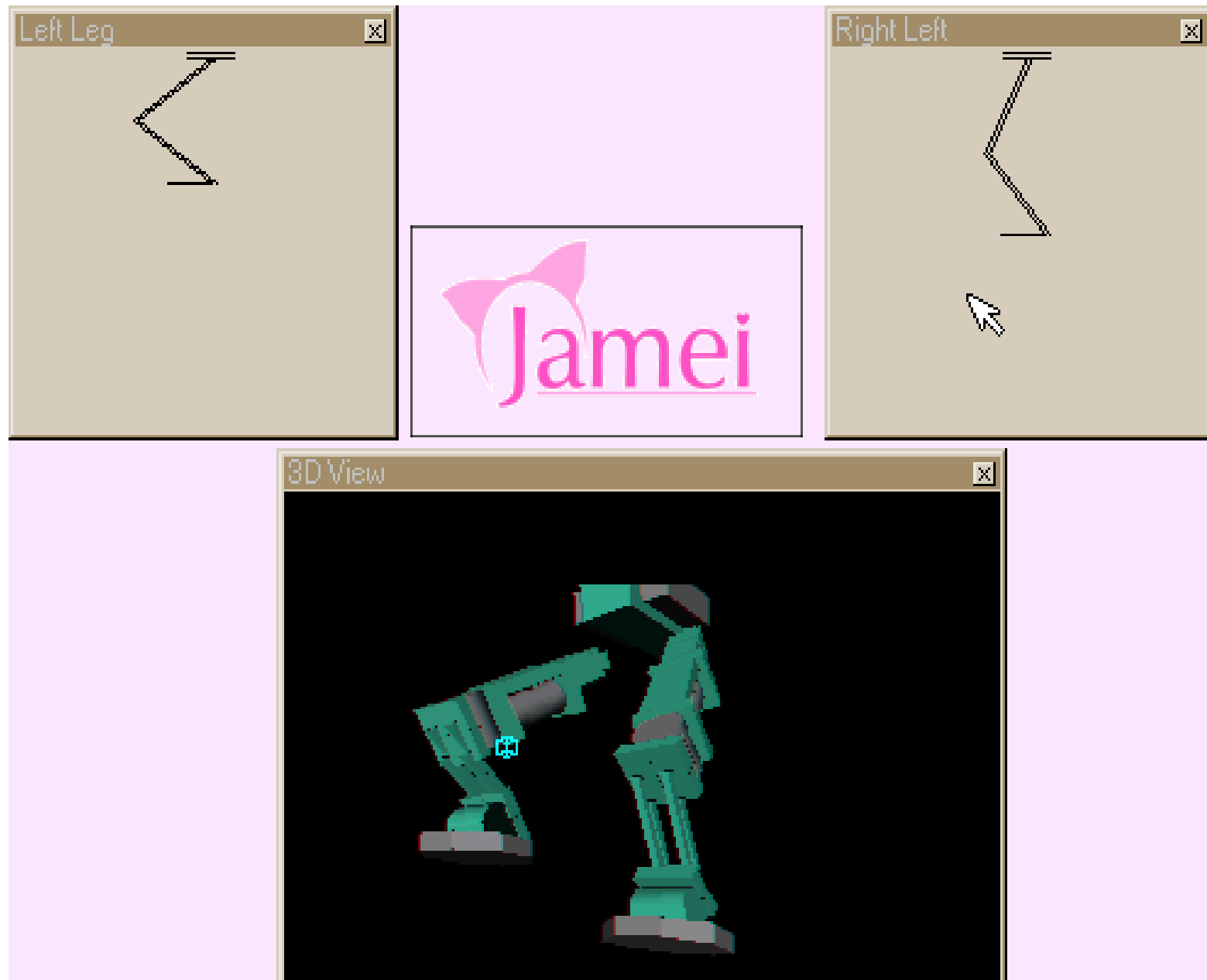


每年寫一套（玩具） 作業系統當作業

- JK (2001; NCKU homework)
- Orz Microkernel (2006)
- RT nanokernel (2007; OSDC.tw)
- Jamei RTOS (2007; COSCUP)
- CuRT (2009; COSCUP)
- TMK (2010; ICOS)
- ??? (2012; Assign homework to NCKU students)

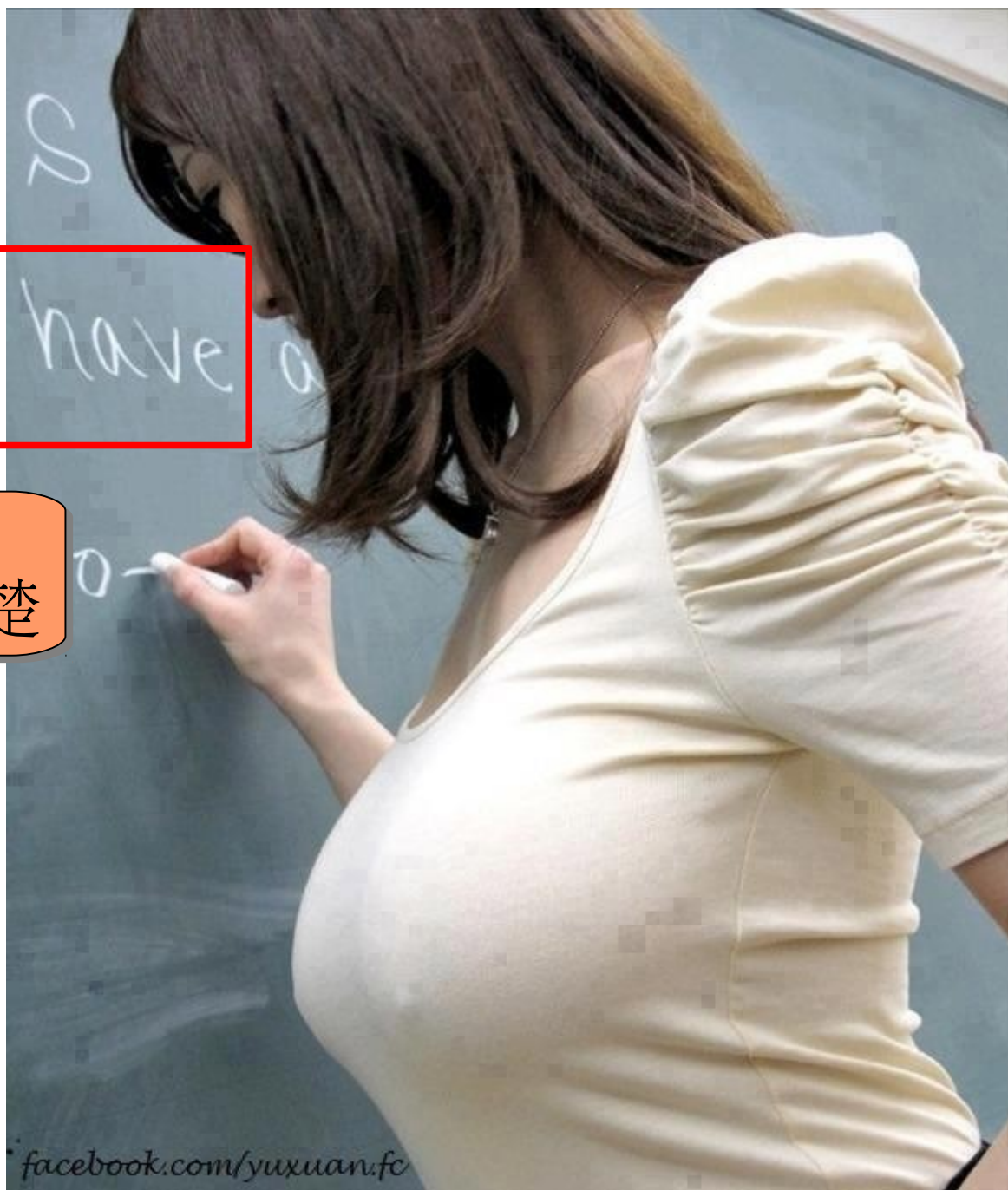


Show Me the Robot!



等等，開發作業系統核心有意義嗎？



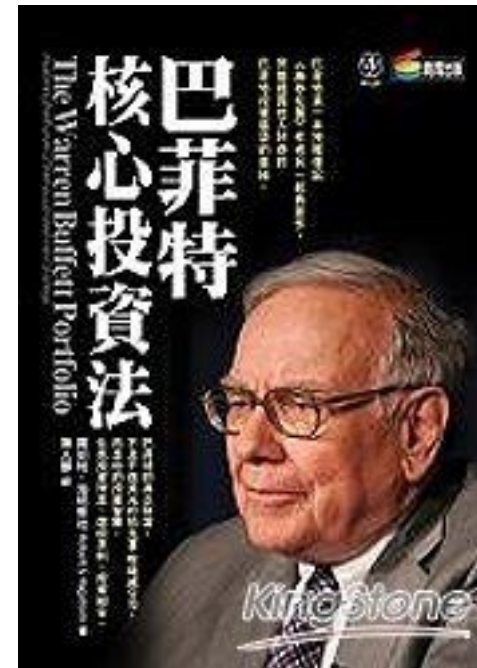


明明就有，
但你沒看清楚



現在更需要開發作業系統核心的技能

- 許多 Web/Application Framework 師法先進作業系統的效能技巧
- 虛擬化技術
 - Hypervisor, Resource Kernel (KVM), ...
 - Intel VT-d / ARM Cortex-A15
- 資訊安全領域
 - 以特製的 VM 作動態分析與追蹤
- 逃離鬼島
 - NASA 需要你！

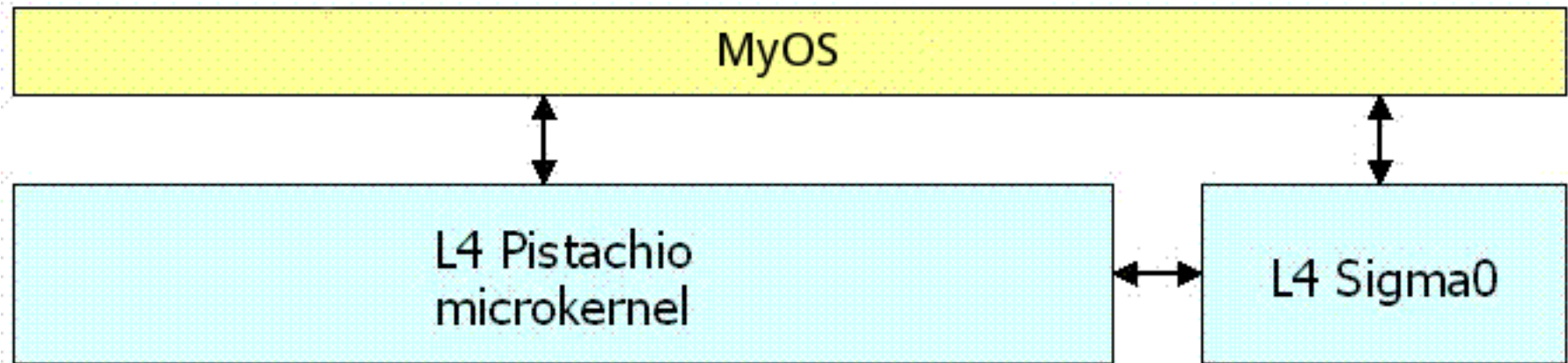


以微核心建構 OS Kernel



策略：Nested Kernel

- 直接將 OS Kernel 架構於一個成熟的 microkernel 之上，避免直接碰觸硬體
- 回歸「設計」本質
- 將 microkernel 的 IPC/RPC 服務再進一步抽象





MyOS

L4 Sigma0

L4 microkernel

MyOS

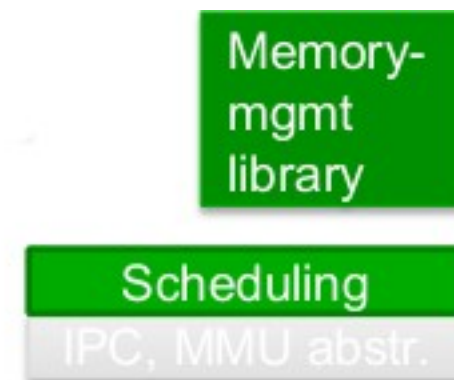
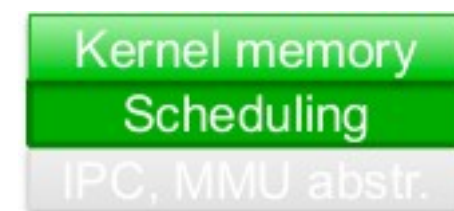
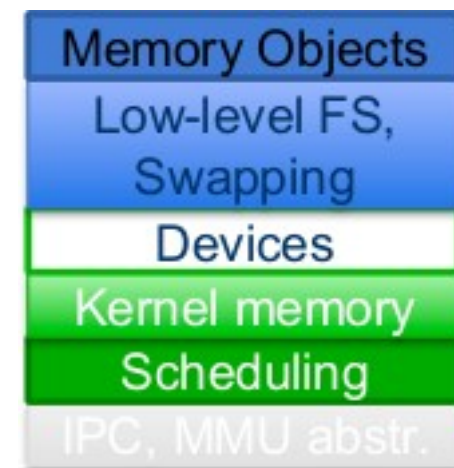
L4 Pistachio
microkernel

L4 Sigma0



Microkernel 的三個世代

- Mach, Chorus (1985-1994)
 - 將 UNIX pipes 替換為 IPC (更通用)
 - 改善系統穩定度 (與 monolithic 相比)
 - 低劣的執行效能
- L3 & L4 (1990-2001)
 - 針對 IPC 大量的效能改善
 - 以組合語言開發，低可攜性
 - 只提供同步的 (sync) IPC
 - 非常小的實做，大部分的功能移到 userspace
- seL4, Coyotos, Nova (2000- 目前)
 - 儘量作到與平台無關
 - verification, security, multiple CPUs, etc.



第二代 Microkernel (L4) 典型設計

Device Drivers

User Program

Memory
Managers

User Mode

Address spacing

Thread
Management
and IPC

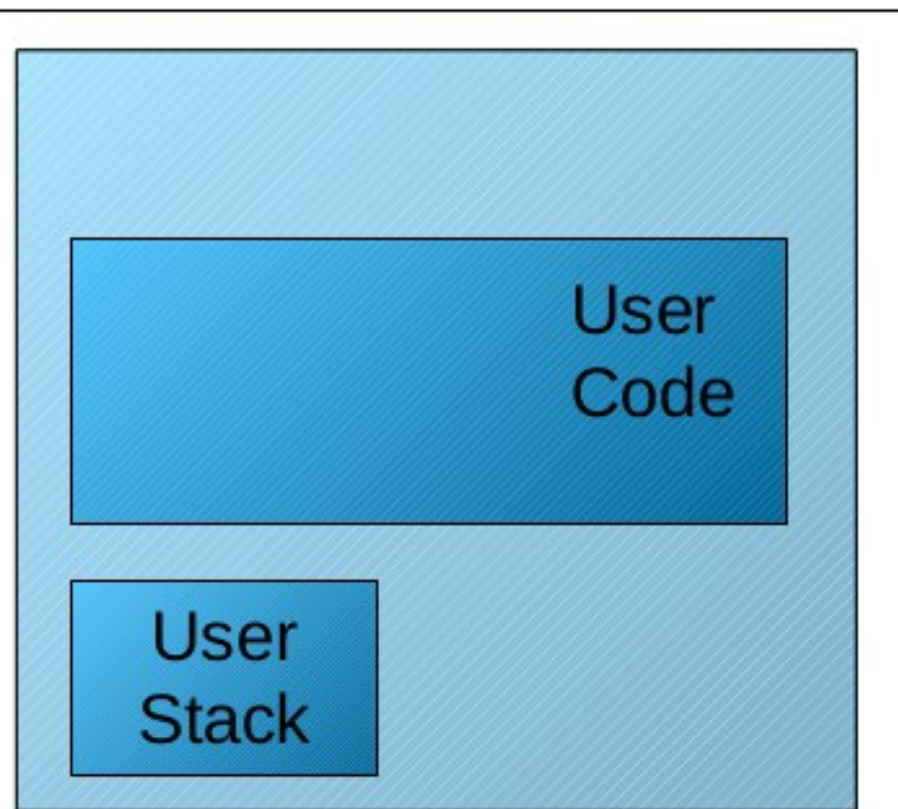
Unique
Identifiers

Microkernel Mode

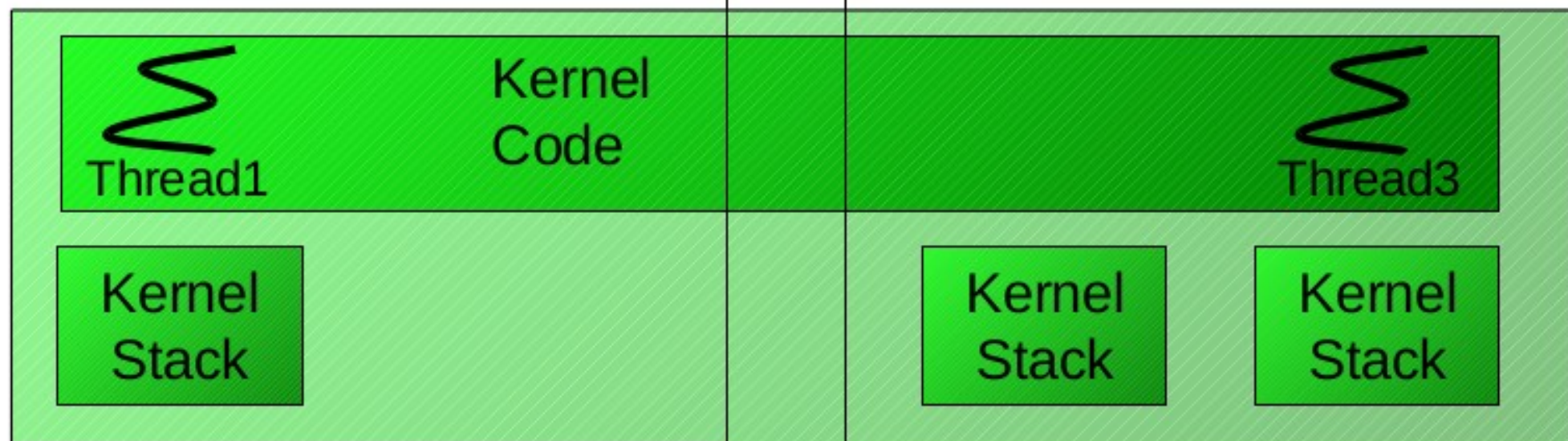
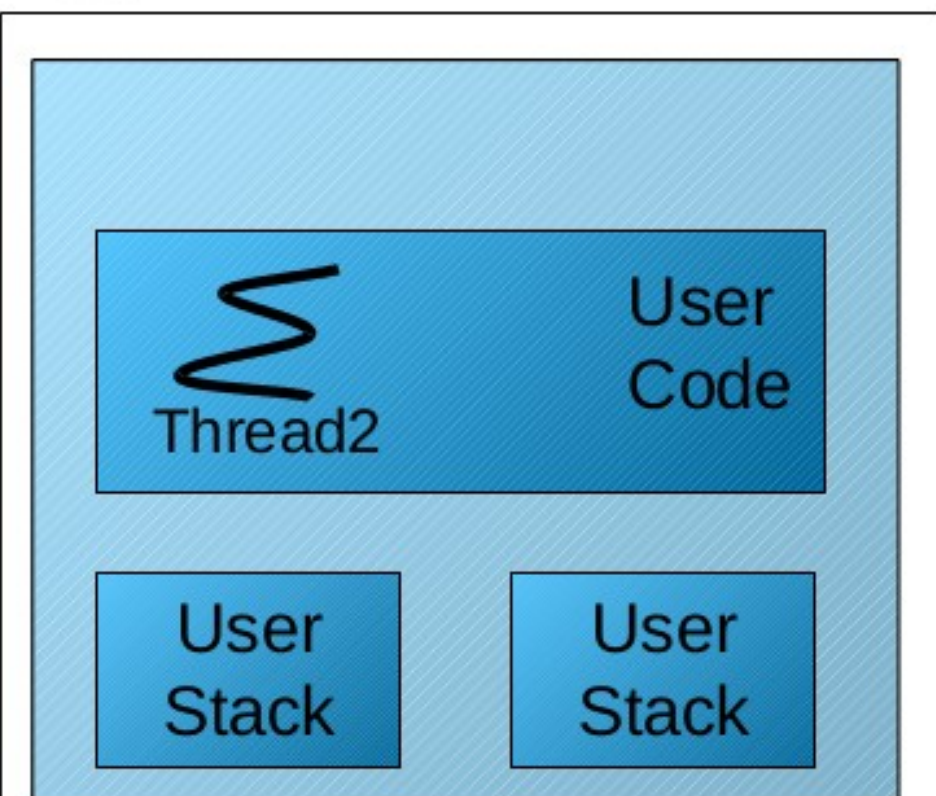
Hardware



Task A

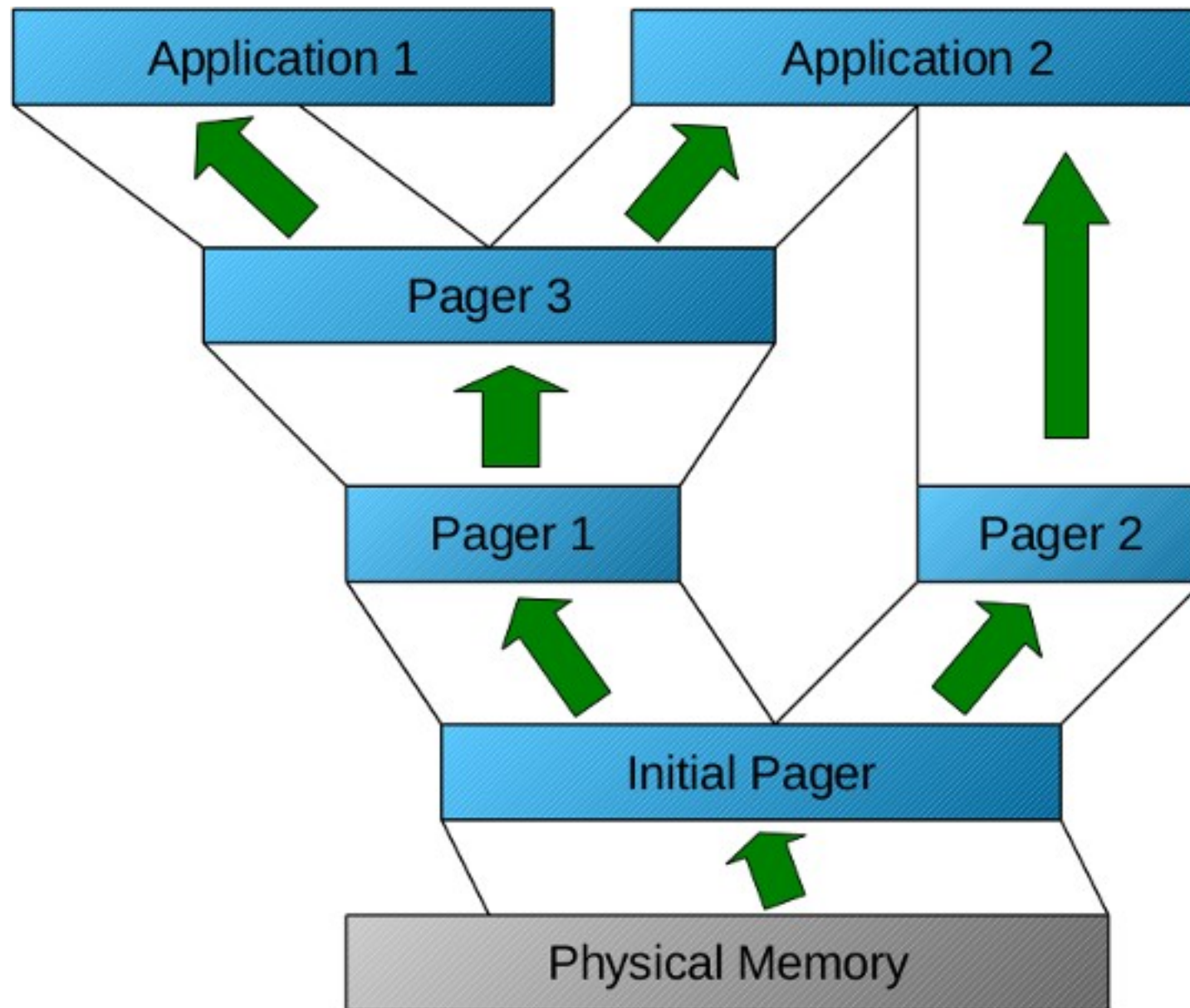


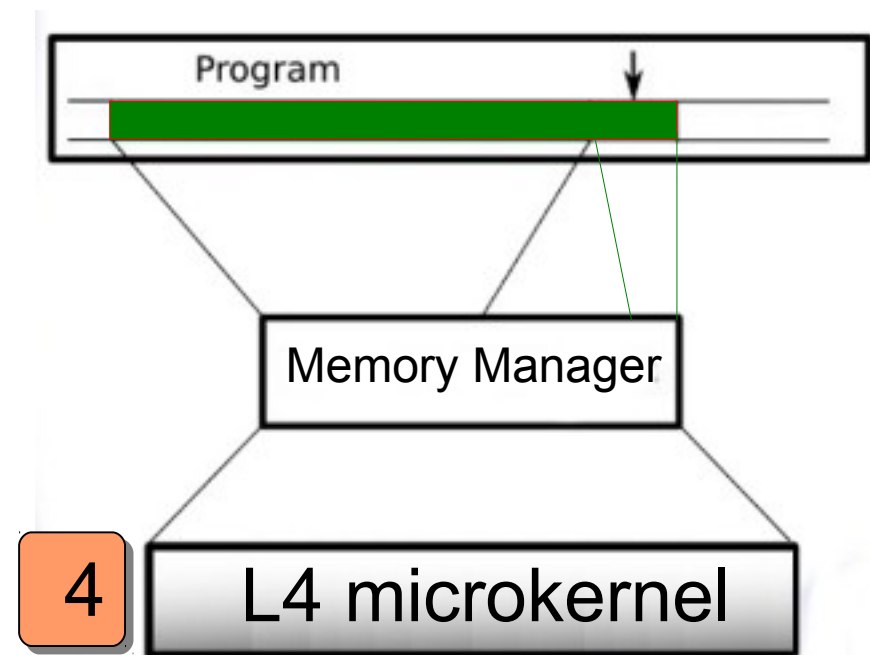
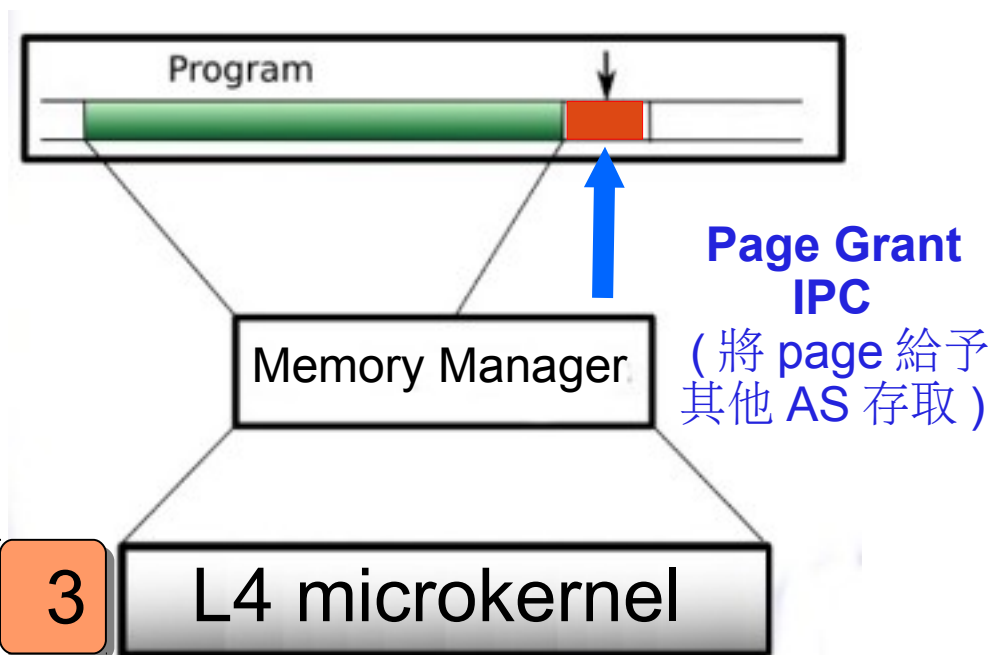
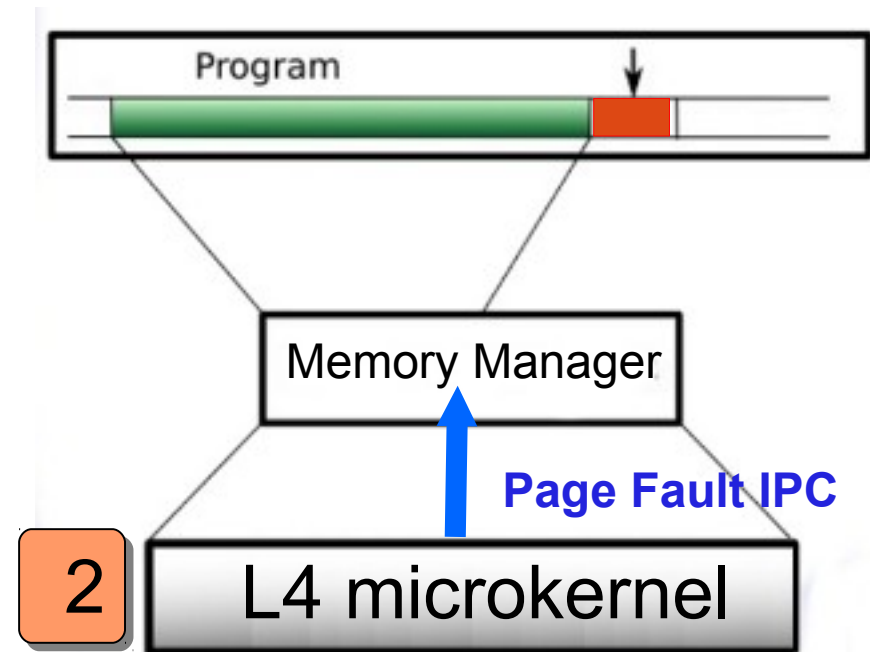
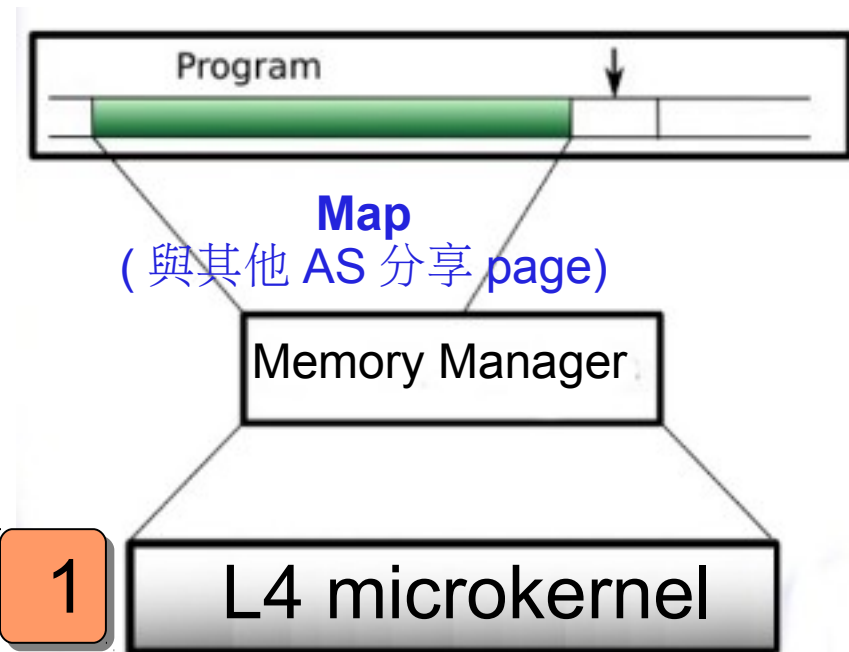
Task B

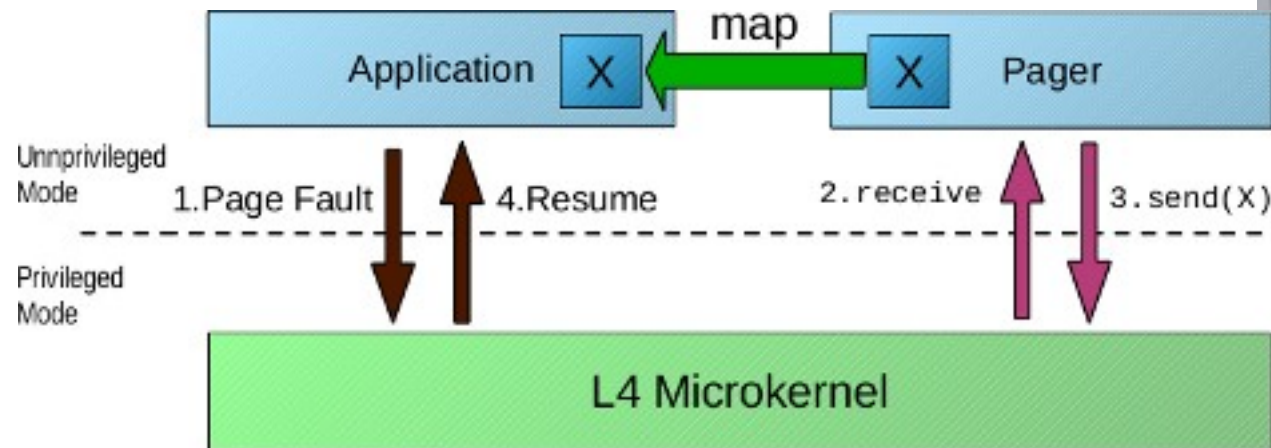


Microkernel

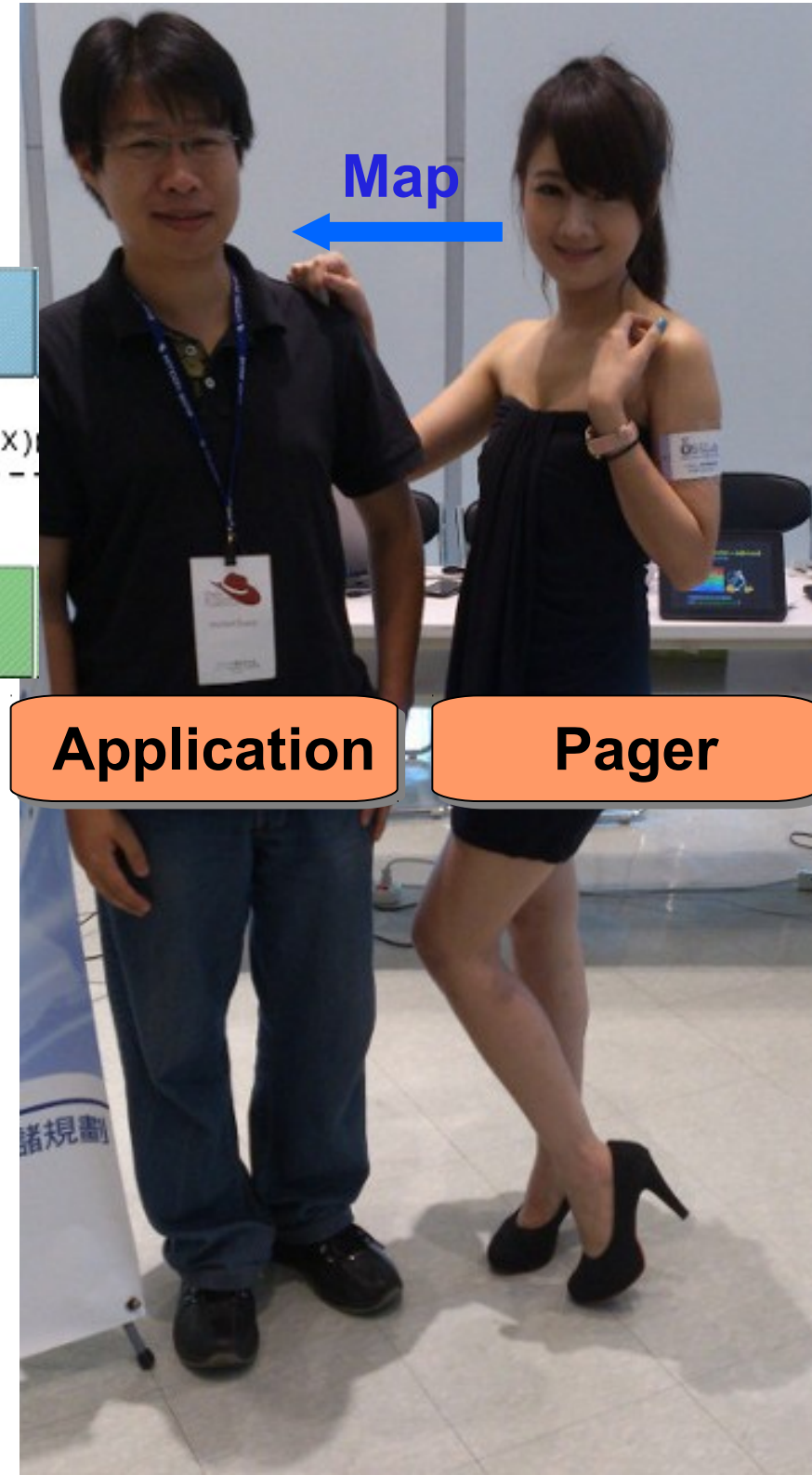
遞迴式的 Address Space

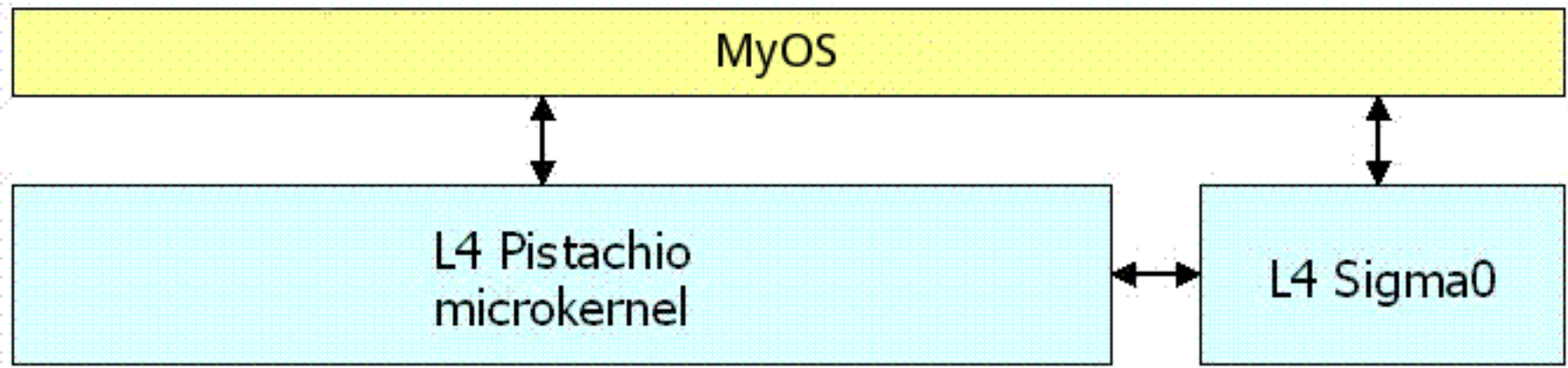






- 若系統存在兩個不同 address space 的執行單元 (如 Process)，而其中一者 (userA) 想要存取另一者 (userB) 的記憶體，只要將 userA 設置為 userB 的 pager (記憶體管理單元) 並提供 userB 的 page fault handler 即可



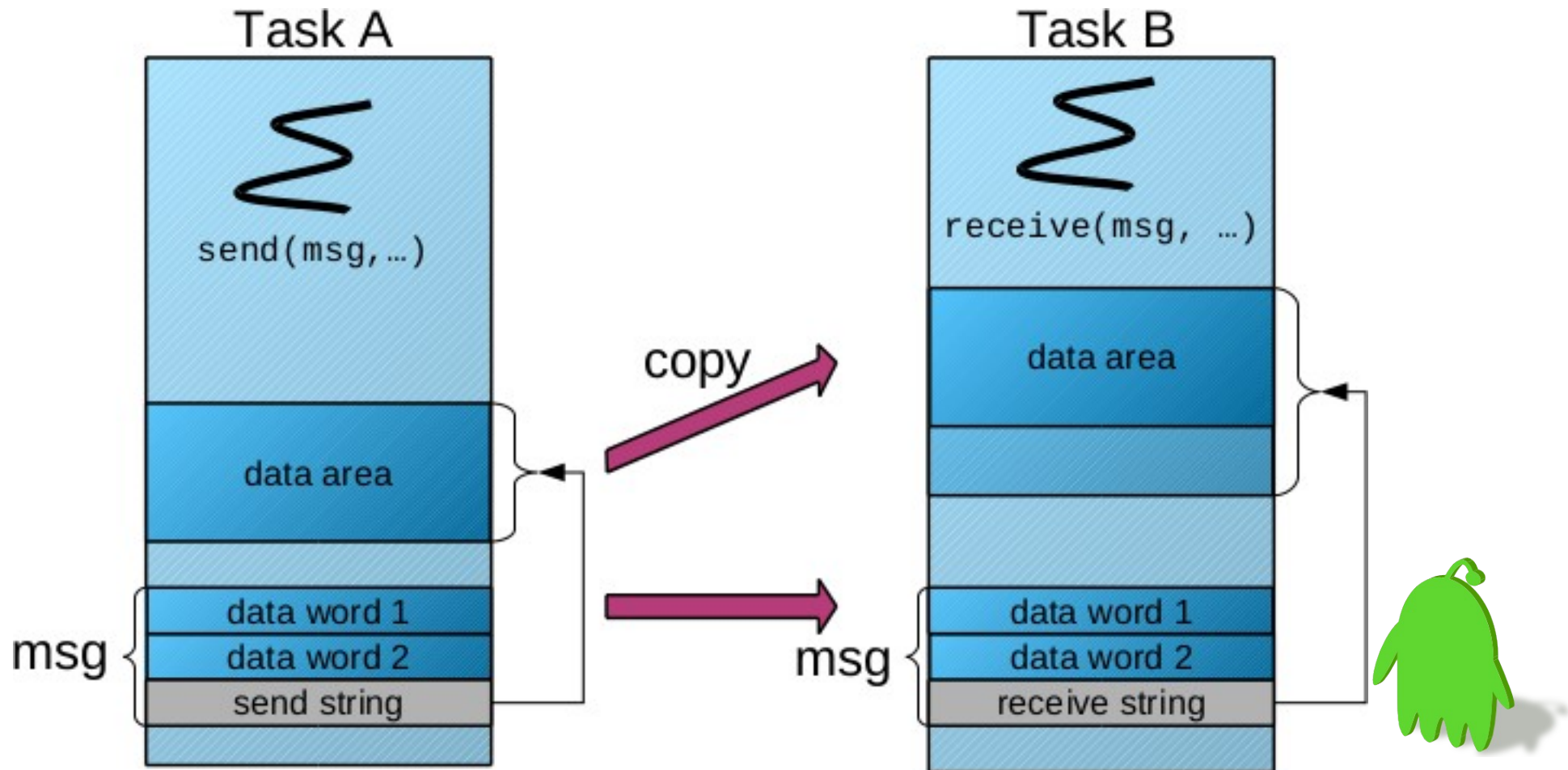


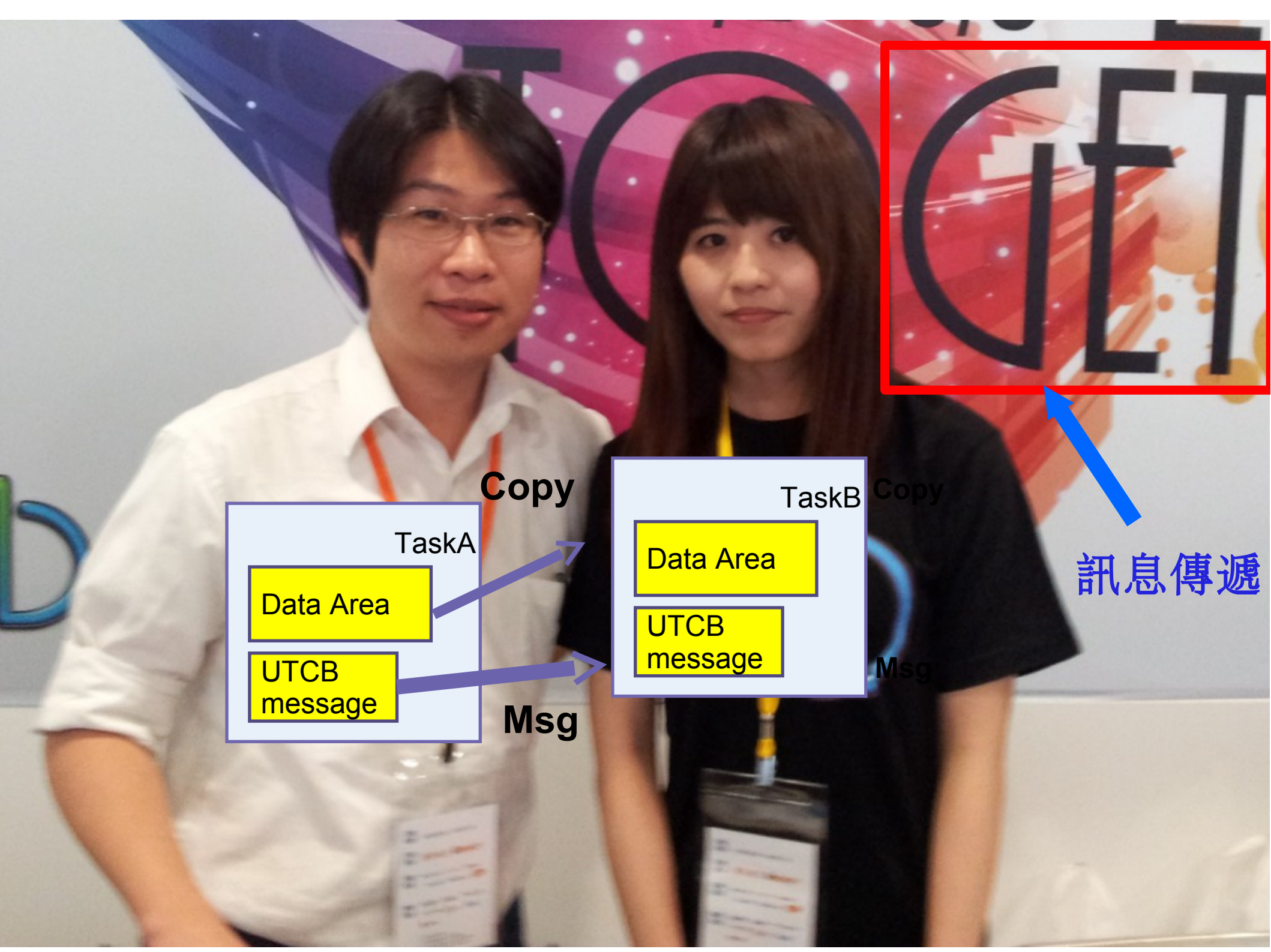
- Sigma0 RPC 通訊協定是 L4 除了 FastIPC 外，另一個主要的設計，用以處理記憶體管理
- 系統的 initial pager 為 sigma0，而 MyOS 在 L4 的術語叫做 root task / server



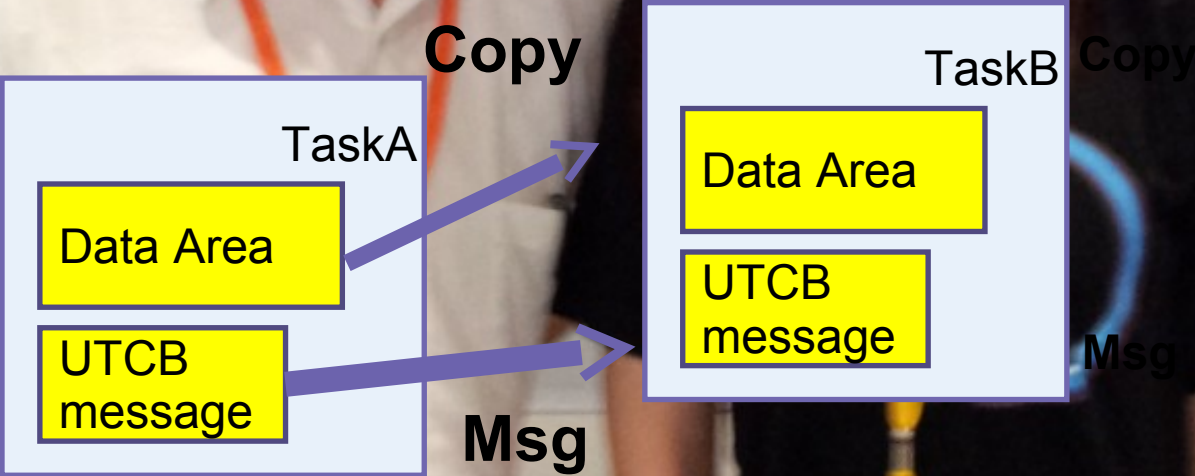
訊息傳遞機制：Copy Data

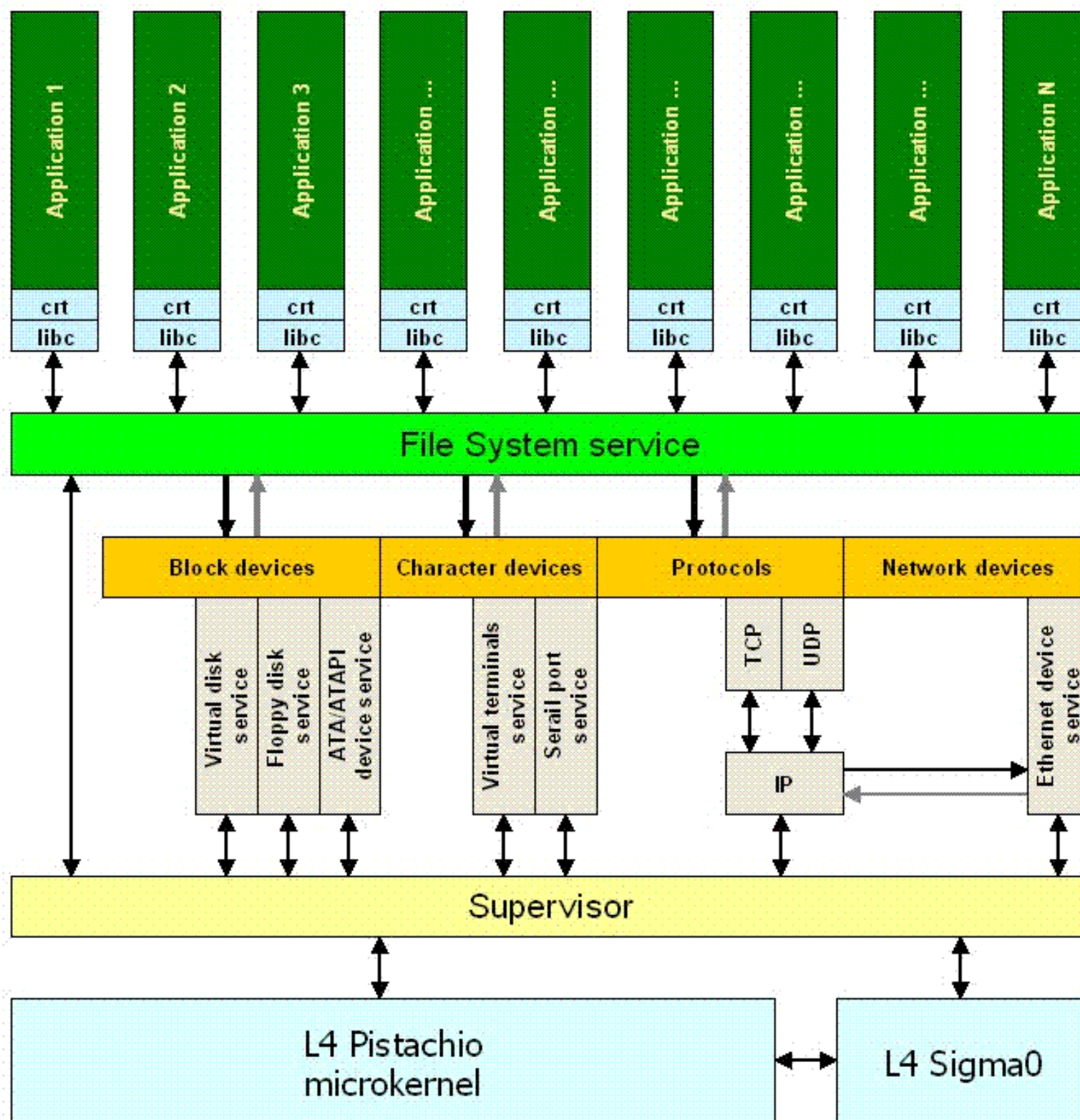
- 直接與間接的資料複製
- UTCB message (special area)
- Special case: register-only message
- Pagefaults during user-level memory access possible





訊息傳遞

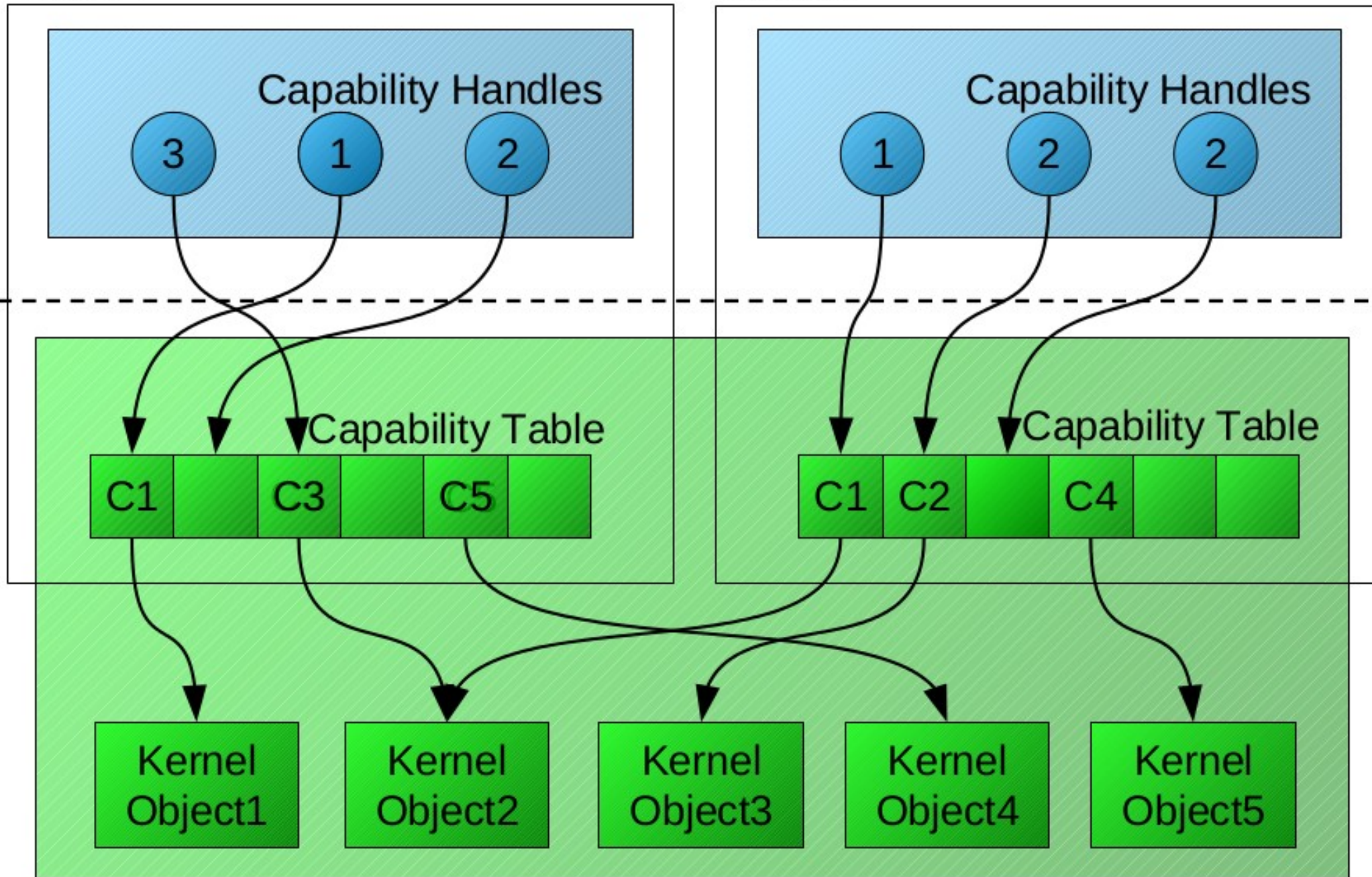




Capabilities (執行特定 IPC 的權級單位)

Task A

Task B



立馬體驗： 自幹 MyOS



Blu-ray 3D™



MDR-NC300D

數位降噪耳機

購下此張光碟，即下送大名鼎鼎的數位降噪耳機/索尼MDR-NC300D
索尼音樂公司(台北市光復北路111號35樓)古倫部門經
銷有獎券抽獎 Sony數位降噪式耳機 MDR-NC300D
活動期間→即日起至2011/02/15 詳情請洽
各大索尼門市或索尼台灣分公司
www.sony.com.tw



加盟索尼音樂首張重量級大碟

LANG LANG

郎朗

Live in Vienna

維也納音樂會現場實況

最新3D特效科技錄製





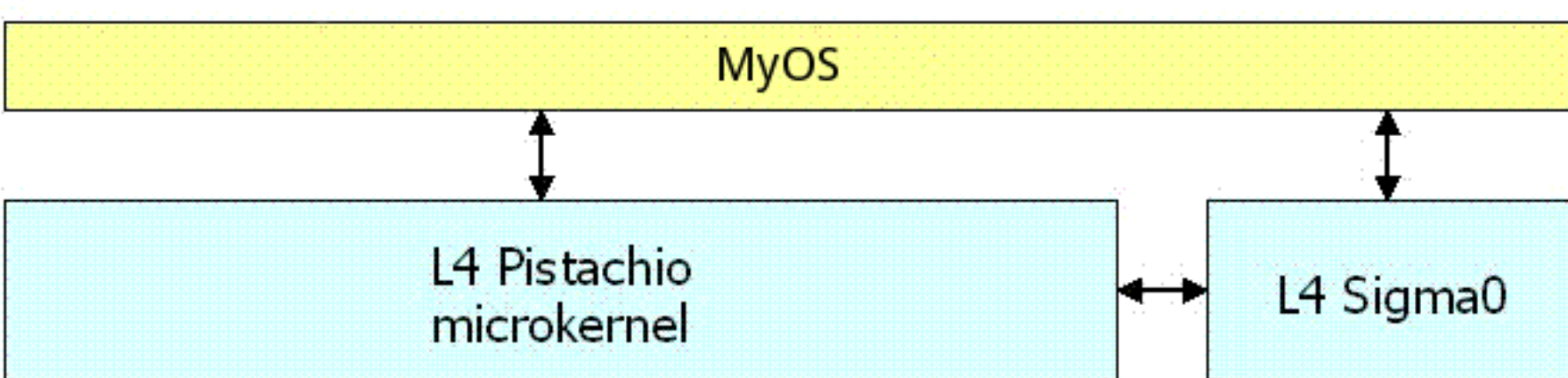
在 OSDC.tw 2012 挑戰失敗，
群群轉戰 COSCUP 繼續自幹

- 真的能動的 OS!
 - 建構於 L4Ka (Pistachio) 的基礎之上

- 以 CORBA 介面串連個別系統元件

```
interface File {  
    boolean Read(in Capability_t cap,  
                 in unsigned long offset,  
                 out byteseq_t buffer,  
                 inout unsigned long size);  
    unsigned long Size(in Capability_t cap);  
    ...  
}
```

- Name System
- 實做基本的檔案系統
- 提供 Tiny Ruby 一類的程式語言執行環境



「寫一個 OS 是多麼美好的事，在有限
的生命中千萬不要遺漏了它」

王佑中博士 (2009)





MOS BURGER

簡單的事重複做
就會是專家
重複的事用心做
就會是贏家



Reference

- Microkernel Construction (SS2012), TU Dresden Operating System Group
- 將 CuRT 用作大學作業系統教材：
<http://sites.google.com/site/embedded2009/introduction-to-curt-v1>
- Write Your Own Operating System [FAQ]:
<http://www.superfrink.net/athenaeum/OS-FAQ/os-faq.html>
- SigOPS, How to Write an Operating System:
http://www.acm.uiuc.edu/sigops/roll_your_own/
- OSDev Wiki: <http://wiki.osdev.org/>





<http://0xlab.org>