

輸入／輸出系統

大綱

- 裝置管理基本問題
- I/O系統階層式模型
- 常見的I/O裝置
- 裝置驅動程式
- Linux I/O系統實例
- 本章重點回顧

前言

■ 輸入輸出系統(Input/Output system)

- 作業系統提供的另外一個子系統
- 管理通訊裝置與其他儲存媒體之用

■ 在作業系統設計上的問題

- 需針對新的裝置設計驅動程式，但新裝置變更發展太快
- 各種I/O裝置差異很大，不容易管理
- 如何安裝裝置的驅動程式設計

裝置的連線通道

■ 周邊裝置的設計：

- 朝向標準化，讓作業系統更容易整合裝置
- 硬體元件會透過匯流排來連接
 - 常用的串接就是在PC匯流排上
 - 再由PCI匯流排擴充出SCSI、串列埠等
 - 粗分為通訊裝置與儲存裝置

裝置管理基本問題

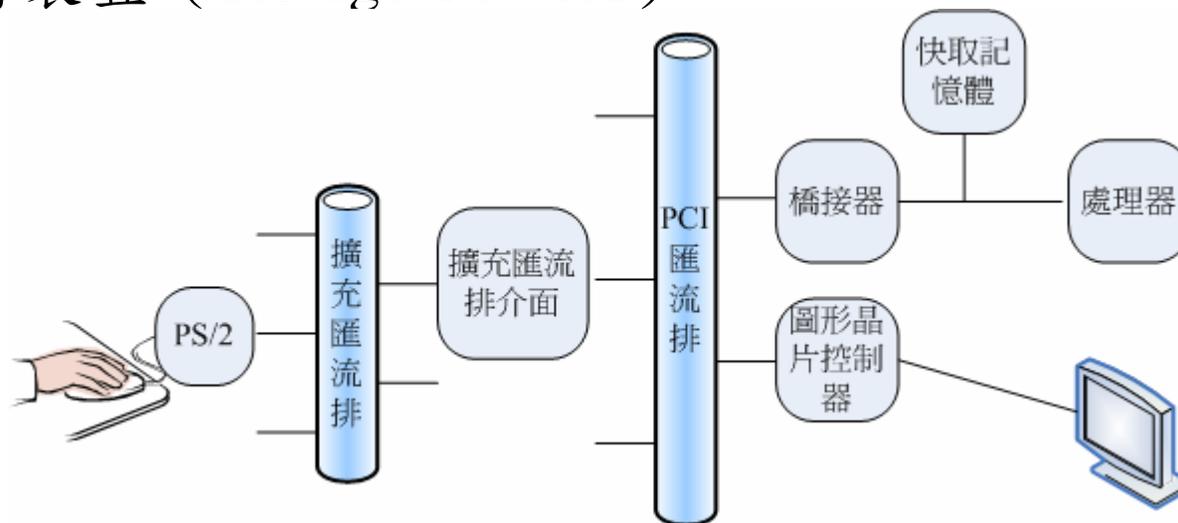
■ 硬體裝置分類：

□ 通訊裝置 (Communication Devices)

■ 輸入裝置

■ 輸出裝置

□ 儲存裝置 (Storage Devices)

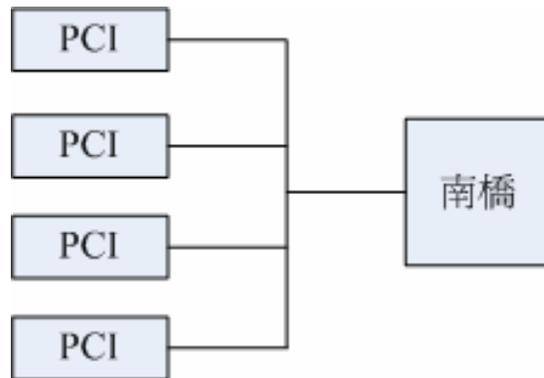


常見的匯流排系統

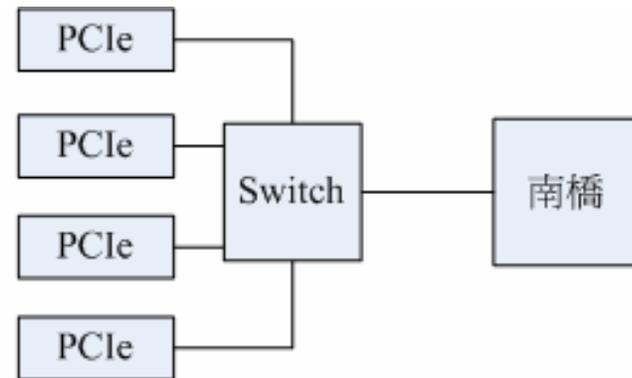
■ 主機板晶片組：

- 北橋：連結CPU、RAM、VGA等高速裝置
- 南橋：連結USB、硬碟、網路等I/O裝置

■ 南橋匯流排架構有 PCI，PCI-Express (PCI-E)



PCI 架構



PCI Express 架構

由PCI轉向PCI-Express的原因，就是因為PCI-E的傳輸介面比起PCI要快的多！

PCI Type	Data Bus	Frequency	Bandwidth
PCI 32/33	32 Bit	33MHz	133.3 MBps
PCI 32/66	32 Bit	66MHz	266.7MBps
PCI 64/33	64 Bit	33MHz	266.7MBps
PCI 64/66	64 Bit	66MHz	533.3MBps

PCIe Type	Bandwidth single direction	Bandwidth dual direction
PCI Express x1	250MBps	500MBps
PCI Express x2	500MBps	1000MBps
PCI Express x4	1000MBps	2000MBps
PCI Express x8	2000MBps	4000MBps
PCI Express x12	3000MBps	6000MBps
PCI Express x16	4000MBps	8000MBps

■ I/O系統基本功能：

- 提供使用者或程序以邏輯或是抽象的方式來檢視與使用這些裝置
- 有效率地使用通訊裝置和儲存裝置。
- 共享這些通訊裝置和儲存裝置。

■ I/O系統與使用者之間：

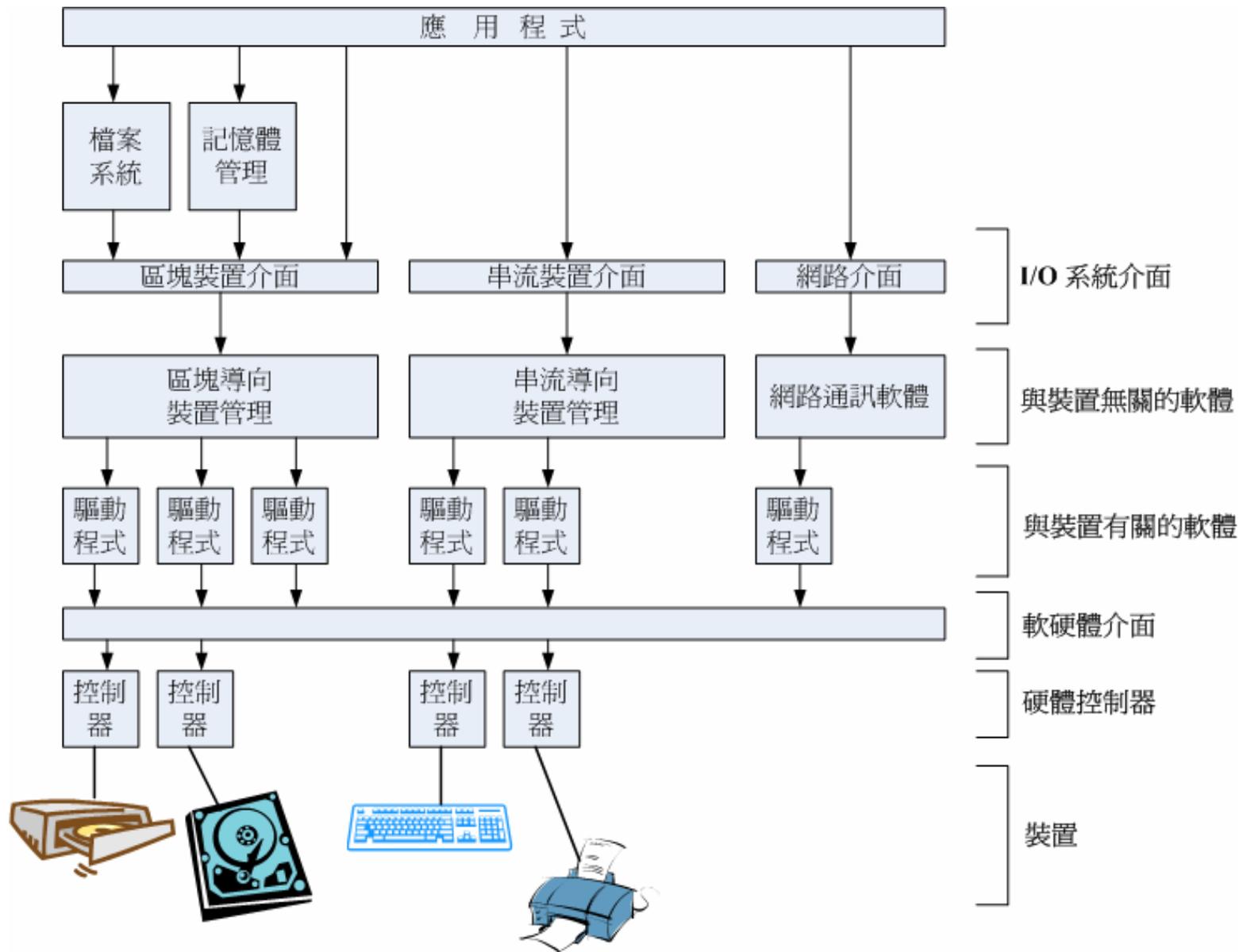
- 使用者或程序不需要知道I/O系統的基本指令
- 使用者或程序僅使用邏輯指令即可(其他的讓I/O去轉換)

■ CPU與I/O之間

- CPU必須要快速的回應這些裝置
- 當多個程序共享裝置時(如磁碟存取)，I/O必須要確定各程序存取的資料為正確的
- 需要針對裝置的配置與存取排程來設計

I/O系統階層式模型

- I/O系統是介於低階硬體裝置介面與作業系統其他較高階子系統之間的軟體層
- I/O系統介面（I/O System Interface）
 - 介於使用者(程序)與硬體間的橋樑
 - 可透過裝置控制器來存取硬體
- I/O系統的階層：
 - 軟體：與裝置沒有直接關係
 - 軟硬體介面（Software-hardware Interface）



I/O介面的類型

■ 區塊裝置介面(block-device)

- 儲存裝置如光碟、磁碟等
- 可隨機存取，以區塊作為組織資料

■ 串流裝置(Stream-device)

- 字元裝置，控制產生或消耗字元串流的裝置
- 例如鍵盤或磁帶等

■ 網路通訊介面

- 利用網路承接口(socket)介面來傳輸資料

I/O的子系統

■ I/O子系統：

- I/O排程
- 緩衝區（Buffer）與快取（Cache）
- 線上同時週邊作業
- 錯誤處理
- 資料串列機制

■ I/O位址：

- /proc/ioproports

常見的I/O裝置-終端機

■ 組成元件：

- 鍵盤、顯示器、滑鼠或軌跡球

■ 顯示器：

- 字元導向：例如純文字介面終端機
 - 25條直線、80個字元
- 圖形導向：透過像素顯示
 - 一個像素的顏色：
 - $2^8=256$ 色

常見的I/O裝置-印表機與掃描器

■ 印表機：

- 輸出裝置，由檔案內的資料輸出到紙張
- 撞針式：點陣式、行列式等
- 非撞針式：噴墨、雷射等

■ 掃描器：

- 輸入裝置，由紙張傳輸到電腦轉存成檔案
- 利用感光器通過影像時，將數值資料轉成位元串流再經轉換得到影像資訊。

常見的I/O裝置-磁碟

■ 現在硬碟組成：

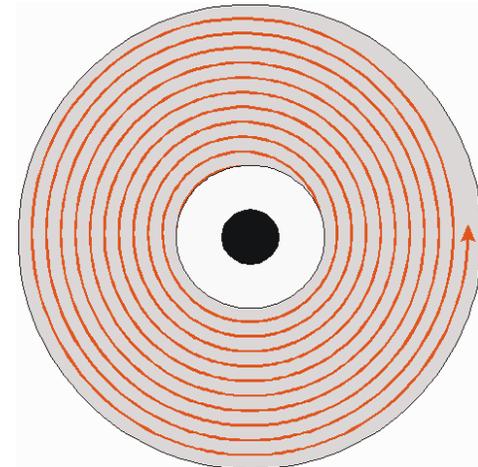
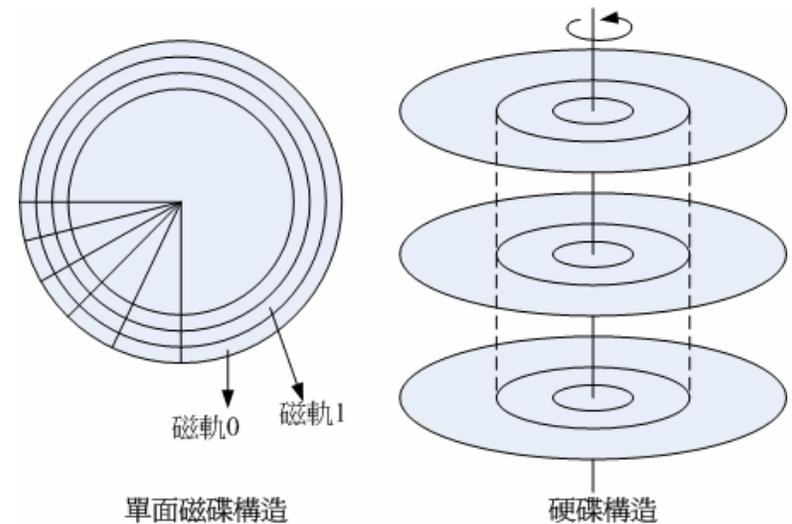
□ 最小單位：

■ 磁區(sector)

□ 磁柱

□ 磁頭

□ 馬達轉速



不同儲存裝置的特性

特性	軟碟	硬碟	光碟
磁區位元個數	512	512 ~ 4,096	2,048
磁區個數	9,15,18,36	100-400	333,000
磁柱個數	40 , 80 , 160	1,000 – 10,000	
磁柱表面個數	1 – 2	2 – 24	1 - 2
臨軌搜尋時間	3 – 5 ms	0.5 – 1.5 ms	N/A
平均搜尋時間	30 – 100 ms	5 – 12 ms	80 – 400 ms
旋轉速度	400 - 700 rpm	3,600 – 10,000 rpm	(200-530)*k rpm

網路裝置的特性

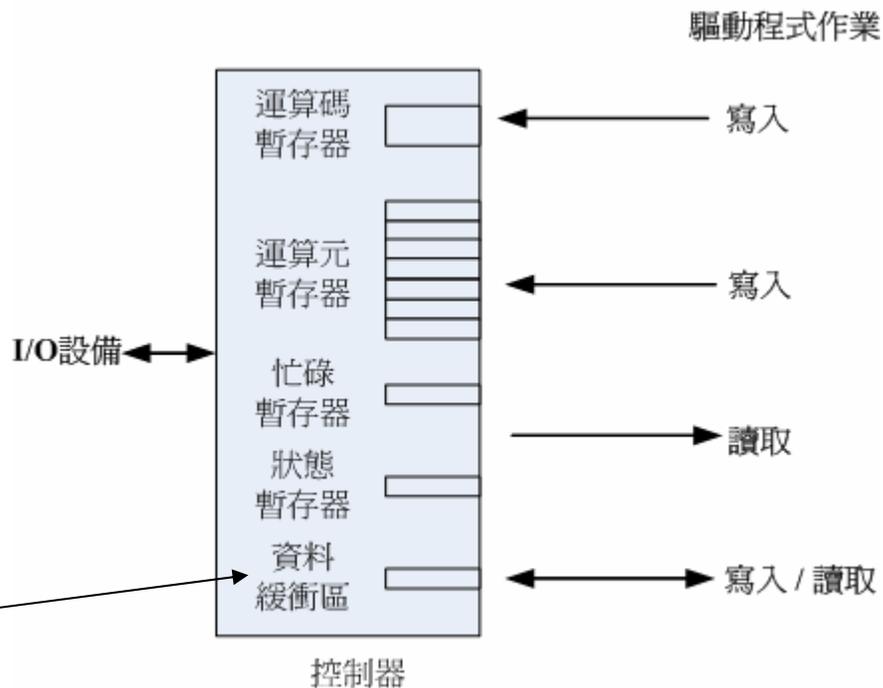
特性	藍芽	IEEE 802.11g	HomeRF
速度	1Mbps	11Mbps	1、2、10Mbps
實體層 (Physical Layer)	FHSS ^[1]	HR/DSSS ^[2]	FHSS
頻帶 (Frequency Band)	2.4GHz	2.4GHz	2.4GHz
適用距離 (Range)	30 fet	50 – 300 fet	150 fet

裝置驅動程式

- 裝置的驅動程式主要是要從I/O系統的較高階程序接受指令，並且和裝置間進行互動。
- 裝置控制器示意圖

重點在資料緩衝區，因為CPU所需要寫入/讀出的資料都放置於緩衝區中

那CPU如何讀取資料緩衝區當中的資料？如何避免被不同的程序所影響？

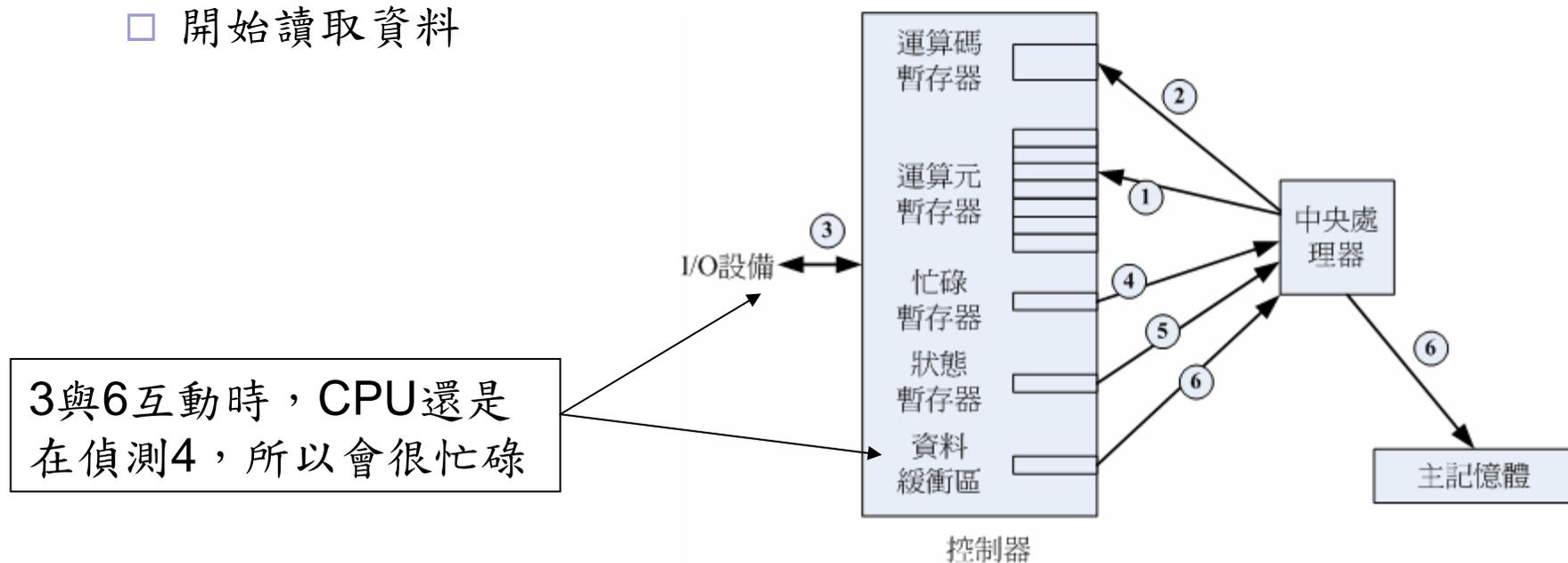


I/O裝置使用的問題

- CPU與裝置(驅動程式)互動的情況
 - 作業系統與應用程式存在著多層次轉換
 - 這些轉換會造成CPU的額外負擔
 - 環境切換的問題
 - 應用程式傳送資料給驅動程式的資料複製
 - I/O裝置的中斷處理
 - 問題：CPU如何處理這些資料的轉換

CPU與裝置的互動：輪詢方式

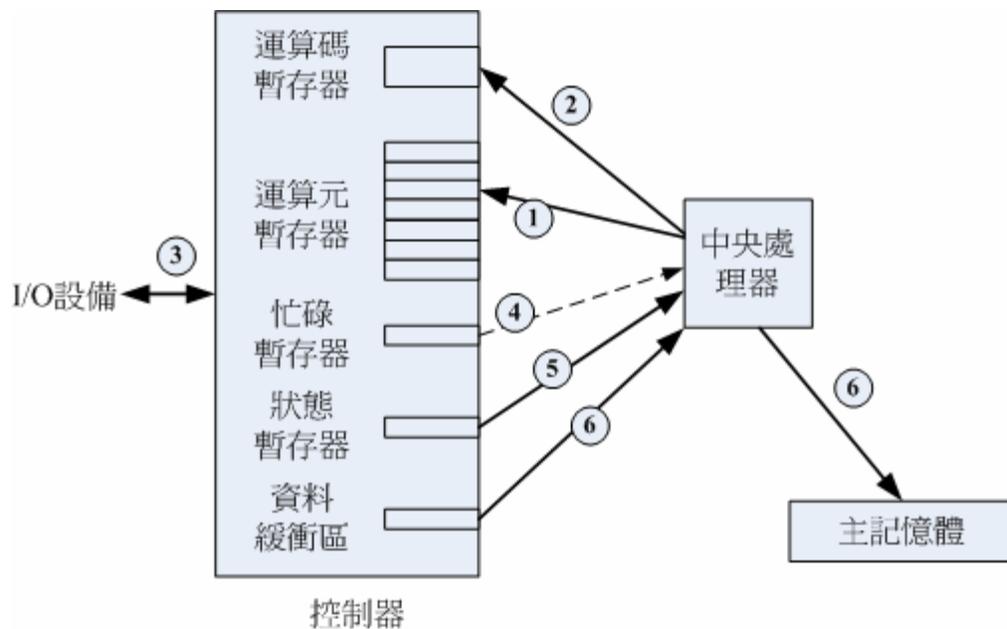
- CPU持續不斷的偵測I/O裝置的活動
- 一直在控制器與主記憶體間偵測
- 處理器需要持續偵測狀態暫存器，故會一直處於忙碌的狀態
- 等上述暫存器為false
 - 開始讀取資料



CPU與裝置的互動：中斷方式

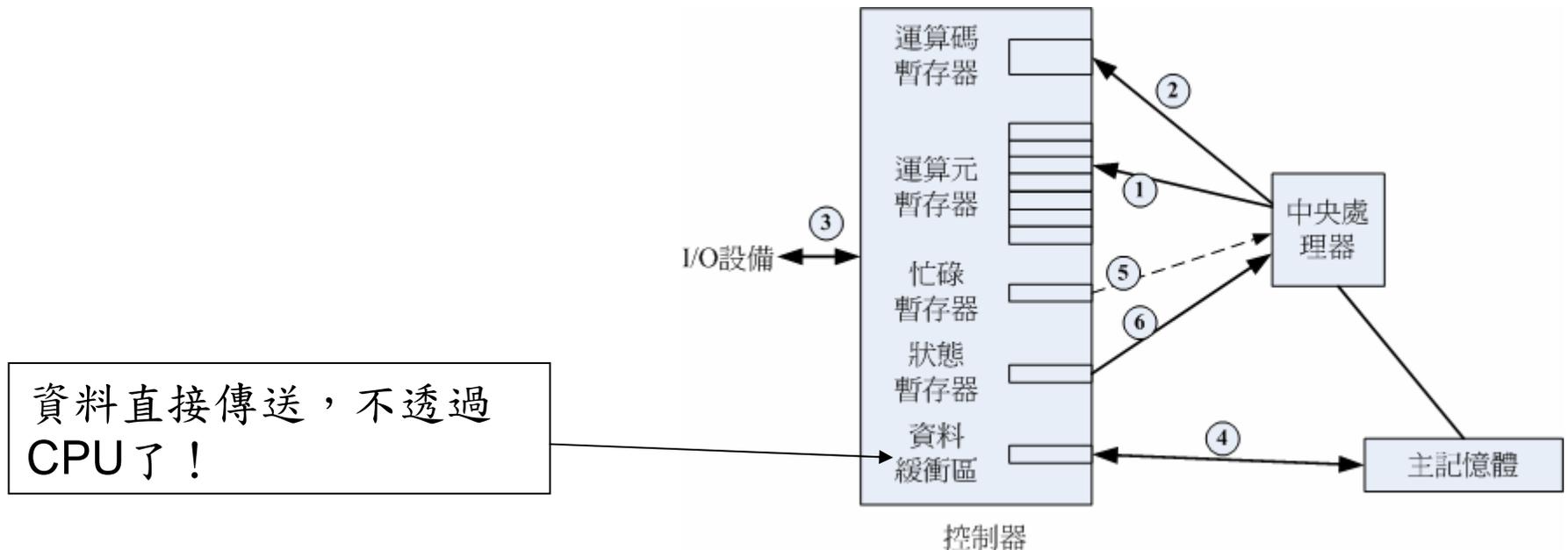
- 透過中斷訊號，讓CPU知道I/O設備正在運作，就不會持續偵測了！
- 等到設備運作完畢，在發送一個中斷給CPU，讓CPU來讀取緩衝區的資料即可。

當3與6互動時，裝置會發出中斷給CPU，所以CPU就不會持續偵測4，因此CPU會有比較多的時間去處理其他任務。



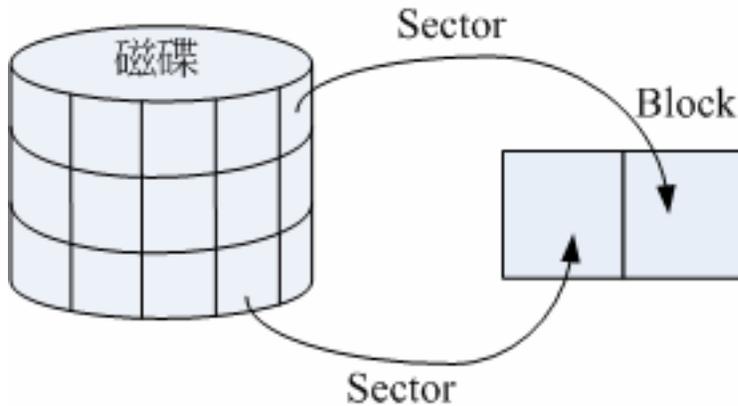
CPU與裝置的互動：DMA方式

- 前面兩種方式中，資料的傳輸都是透過：
 - 資料緩衝區→CPU→主記憶體
- 透過DMA (Direct Memory Access)直接將資料丟給記憶體，等待資料上傳完畢才通知CPU，更節省系統資源了。



Linux I/O系統實例

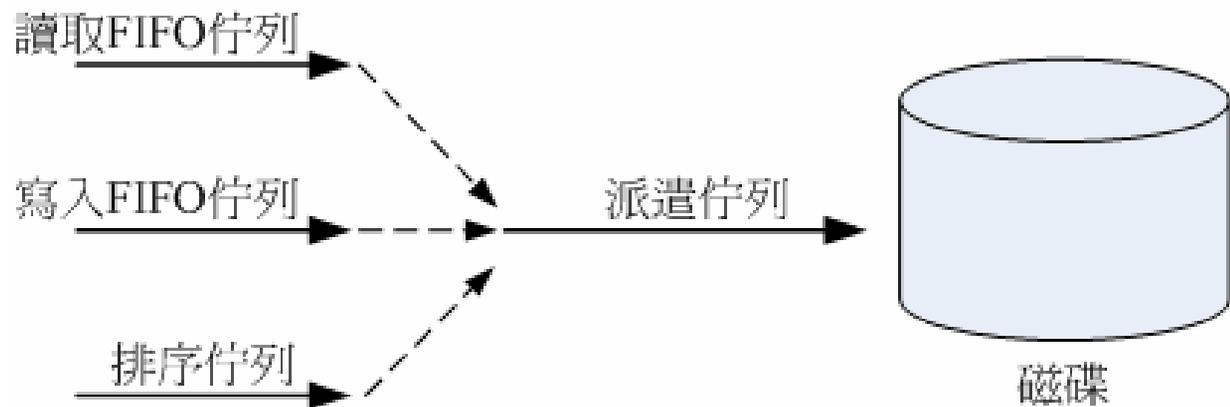
■ 磁區與檔案系統Block關係



■ Linux核心上的I/O結構設計：Buffer Head與Bio結構

■ Linux核心中所支援的排程方式

- The Linus Elevator
- 限期式I/O排程系統
- 預測式I/O排程系統
- 公平佇列式I/O排程系統
- 無行為I/O排程系統



本章重點回顧

- 了解基本的裝置管理問題。
- 了解輸入／輸出系統的階層式架構。
- 了解常見的輸入／輸出裝置與其特性。
- 了解裝置驅動程式於輸入／輸出系統中扮演的角色。
- 將會透過Linux作業系統的輸入／輸出系統實例了解整體的運作模式。

