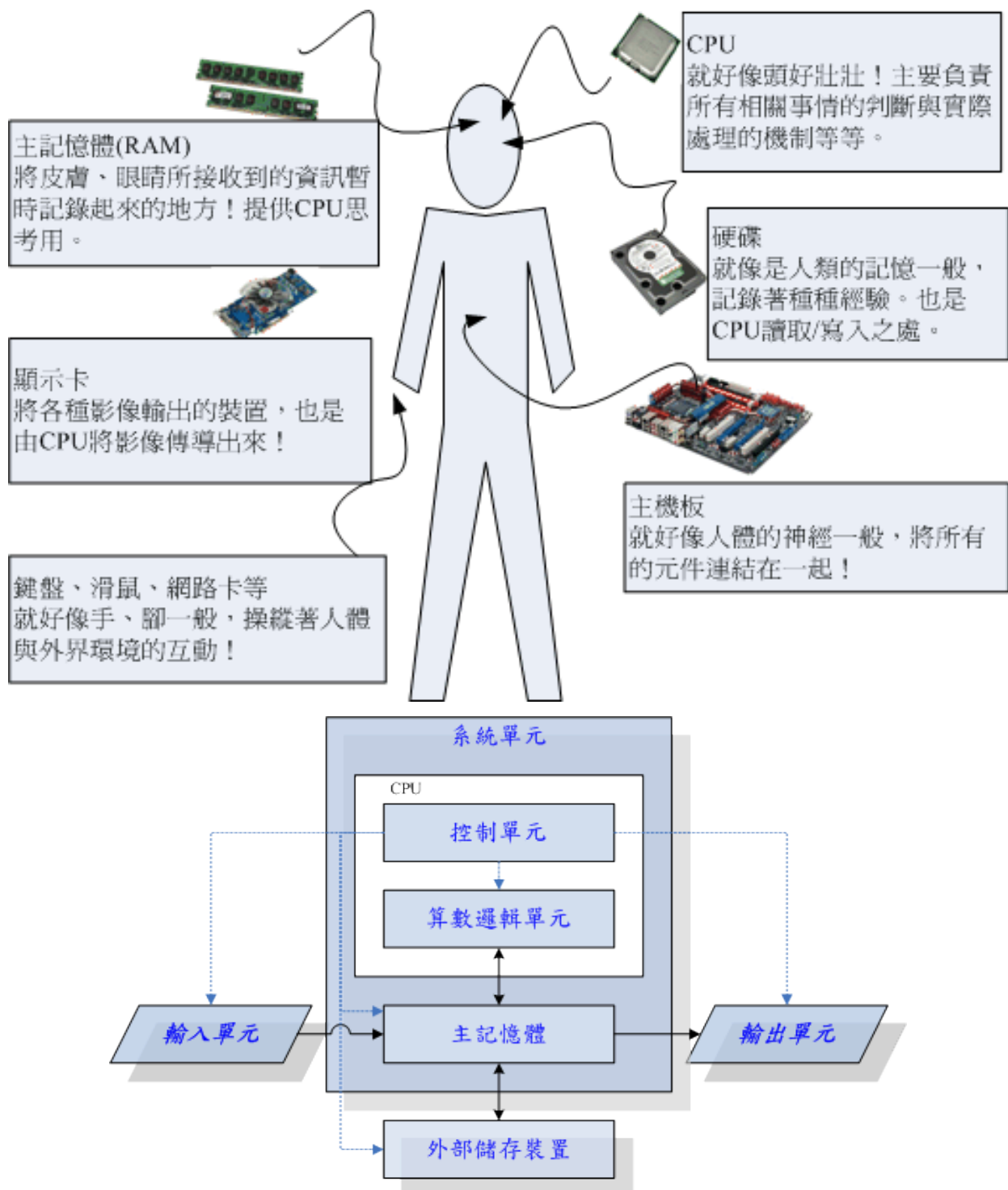


計算機概論

Note / Erebuszz

- 接受使用者輸入指令與資料，經由中央處理器的數學與邏輯單元運算處理後，以產生或儲存成有用的資訊



- 以電腦的複雜度與運算能力進行分類

(由上到下的複雜度、運算能力、價格分別由高到低)

- 超級電腦(Supercomputer)

1. 用於需要高速計算的計畫中
2. 例如：國防軍事、氣象預測、太空科技
3. 國家高速網路與計算中心、全世界最快速的前 500 大超級電腦

- 大型電腦(Mainframe Computer)

1. 用來處理大量資料與複雜的運算
2. 例如大型企業的主機、資料庫伺服器、全國性的證券交易所等

- 迷你電腦(Minicomputer)

1. 同時支援多使用者
2. 主機可以放在一般作業場所，不必像前兩個大型電腦需要特殊的空調場所
3. 通常用來作為科學研究、工程分析與工廠的流程管理等

- 工作站(Workstation)

1. 針對特殊用途而設計的電腦
2. 學術研究與工程分析方面相當常見

- 微電腦(Microcomputer)

1. 個人電腦就屬於這部份的電腦分類
2. 大致又可分為桌上型、筆記型等等
3. 兩大主流 x86 開發商 (Intel, AMD)

註：光以效能來說，目前的個人電腦效能已經夠快了，甚至已經比工作站等級以上的電腦運算速度還要快！但是工作站電腦強調的是穩定不當機，並且運算過程要完全正確

- 電腦上面常用的計算單位

- 容量單位

1. 電腦對資料的判斷主要依據有沒有通電來記錄資訊
2. 檔案容量使用的是二進位的方式
3. 1 Byte = 8 bits (0/1：二進位的的單位)

註：1 GBytes 的檔案大小實際上為：1024x1024x1024 Bytes

- 速度單位

1. CPU 的運算速度常使用 MHz 或者是 GHz 之類的單位 (Hz=次/秒)
2. 網路傳輸方面，由於網路使用的是 bit 為單位，因此網路常使用的單位為 Mbps 是 Mbits per second

ex:

20M/5M => 理論最大傳輸值：每秒 2.5Mbyte/ 每秒 625Kbyte 的下載/上傳速度

- 個人電腦架構與相關設備元件

1. 早期的晶片組通常分為兩個橋接器來控制主機板上各元件的溝通：

- 北橋

- 1) 負責連結速度較快的 CPU、主記憶體與顯示卡界面等元件

- 南橋

- 2) 負責連接速度較慢的裝置介面，包括硬碟、USB、網路卡等等

∴ 北橋最重要的就是 CPU 與主記憶體之間的橋接

→ 目前的主流架構中，大多將北橋記憶體控制器整合到 CPU 封裝中 (Intel 部份使用 QPI (Quick Path Interconnect) 與 DMI 技術，而 AMD 部份則使用 Hyper Transport)

- 中央處理器 (Central Processing Unit, CPU)

1. 一個具有特定功能的晶片，裡頭含有微指令集

2. 兩個主要的單元：

- (1) 算數邏輯單元

主要負責程式運算與邏輯判斷

- (2) 控制單元

主要在協調各周邊元件與各單元間的工作

3. 讀取的資料都是從主記憶體來的，處理完畢的資料也必須要先寫回主記憶體中

註：基本上資料都是流經過主記憶體再轉出去的，因此記憶體的容量如果太小，資料快取就不足，影響效能相當大

4. 不同的 CPU 型號大多具有不同的腳位，能夠搭配的主機板晶片組也不同 → 當你想要將你的主機升級時，不能只考慮 CPU，你還得要留意你的主機板上面所支援的 CPU 型號

- 微指令集

- 1) 我們所使用的軟體都要經過 CPU 內部的微指令集來達成

- 2) 設計理念、目前世界上常見到的兩種主要 CPU 架構：

- 精簡指令集 (Reduced Instruction Set Computer, RISC)

a) 每個指令的執行時間都很短，完成的動作也很單純，指令的執行效能較佳

b)

I. 甲骨文 (Oracle) 公司的 SPARC 系列

→ 學術領域的大型工作站、銀行金融體系的主要伺服器

II. IBM 公司的 Power Architecture (包括 PowerPC) 系列

→ 例如新力(Sony)公司出產的 Play Station 3(PS3)

III. 安謀公司 (ARM Holdings) 的 ARM CPU 系列等

→ 各廠牌手機、PDA、導航系統、網路設備(交換器、路由器等)等，目前世界上使用範圍最廣的 CPU

■ 複雜指令集(Complex Instruction Set Computer, CISC)

a) 每個小指令可以執行一些較低階的硬體操作，指令數目多而且複雜，每條指令的長度並不相同

b) 因為指令執行較為複雜所以每條指令花費的時間較長，但每條個別指令可以處理的工作較為豐富

c) AMD、Intel、VIA 等的 x86 架構的 CPU

I. x86 架構 CPU 被大量使用於個人電腦用途上面

II. 最早的那顆 Intel 發展出來的 CPU 代號稱為 8086，後來依此架構又開發出 80286, 80386...，因此這種架構的 CPU 就被稱為 x86 架構了

○ 時脈

1) 時脈就是 CPU 每秒鐘可以進行的工作次數

■ 外頻

a) CPU 與外部元件進行資料傳輸時的速度

- 倍頻

a) CPU 內部用來加速工作效能的一個倍數

→ 兩者相乘才是 CPU 的時脈速度：內頻

2) 現在 Intel 的 CPU 會主動幫你超頻，透過的是 Intel 的 turbo 技術。
同時，如果你沒有大量的運算需求，CPU 時脈會往下降，藉此達到
節能省電的目的

註：所謂的超頻指的是：將 CPU 的倍頻或者是外頻透過主機板的設定功能更
改成較高頻率的一種方式。但因為 CPU 的倍頻通常在出廠時已經被鎖定而無
法修改，因此較常被超頻的為外頻

- 32 位元與 64 位元的 CPU 與匯流排『寬度』

- 系統匯流排

- a) CPU 與主記憶體、顯示卡溝通的匯流排

- 輸入輸出 (I/O) 匯流排

- a) 南橋，主要在聯繫硬碟、USB、網路卡等周邊設備

- 前端匯流排 (Front Side Bus, FSB)

- a) CPU 內的記憶體控制晶片與主記憶體間的傳輸通道

- 匯流排『寬度』

- a) 每次時脈能夠傳輸的資料量

- 頻寬

- a) FSB 頻率 * 匯流排寬度

- 字組大小 (word size)

- a) CPU 每次能夠處理的資料量

b) 依據 CPU 的設計而有 32 位元與 64 位元

c) 我們現在所稱的電腦是 32 或 64 位元的主要依據

○ CPU 等級

∴ x86 架構的 CPU 在 Intel 的 Pentium 系列 (1993 年) 後就有不統一的腳位與設計 → 將不同種類的 CPU 規範等級

1) Intel Pentium MMX 與 AMD K6 年代的 CPU 稱為 i586 等級

2) Intel Celeron 與 AMD Athlon(K7)年代之後的 32 位元 CPU 就稱為 i686 等級

3) 目前的 64 位元 CPU 則統稱為 x86_64 等級

4) 向下相容

○ 超執行緒 (Hyper-Threading, HT)

∴ 現在的 CPU 運算速度都太快了，運算核心經常處於閒置狀態下

→ 現在的系統大多都是多工的系統，CPU 可以假象的同時執行兩個程序，讓系統效能增加

1) 如何達成？

在每一個 CPU 內部將重要的暫存器 (register) 分成兩群，而讓程序分別使用這兩群暫存器。也就是說，可以有兩個程序『同時競爭 CPU 的運算單元』，而非透過作業系統的多工切換！這一過程就會讓 CPU 好像『同時有兩個核心』的模樣

2) case by case，而且使用的軟體影響很大

補充：CPU-Z

1) CPU 規格檢查程式

2) CPU 的詳細規格、L1、L2 快取大小、主機板型號、記憶體大小及規格、記憶體插槽 (SPD)、顯示卡資訊、Windows 及 DirectX 版本...等等

- 記憶體

- 動態隨機存取記憶體 (Dynamic Random Access Memory, DRAM)

- 1. 個人電腦的主記憶體主要元件

- 2. 隨機存取記憶體 (RAM) 只有在通電時才能記錄與使用，斷電後資料就消失了

- 又稱為揮發性記憶體

- 3. 以伺服器來說，主記憶體的容量有時比 CPU 的速度還要來的重要的

- SDRAM

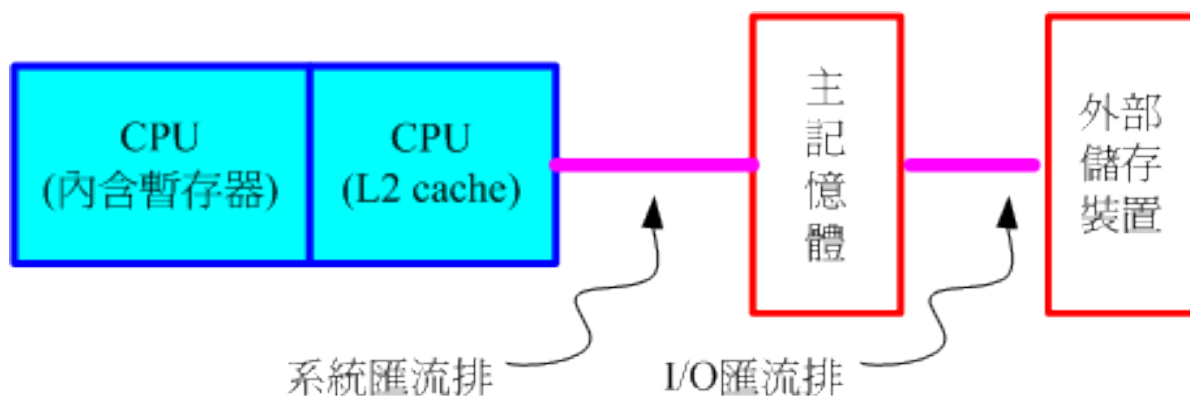
- DDR SDRAM

- 1. DDR 是所謂的雙倍資料傳送速度(Double Data Rate)，他可以在一次工作週期中進行兩次資料的傳送，感覺上就好像是 CPU 的倍頻

- 2. 又依據技術的發展，有 DDR, DDR2, DDR3, DDR4 等等

註：DDR3L 為了節省更多的電力，新的製程中降低了主記憶體的操作電壓，因此 DDR3 標準電壓為 1.5V，但 DDR3L 則僅須 1.35V，通常可以用在耗電量需求更低的筆電中

- 靜態隨機存取記憶體(Static Random Access Memory, SRAM)



1. 在設計上使用的電晶體數量較多，價格較高，且不易做成大容量，不過由於其速度快 → 整合到 CPU 內成為快取記憶體以加快資料的存取

ex: 第二層快取 (L2 cache)

○ 唯讀記憶體 (Read Only Memory, ROM)

1. 一種非揮發性的記憶體，一旦儲存資料就無法再將之改變或刪除，且內容不會因為電源關閉而消失

○ 多通道設計

1. 設計理念：

主記憶體的資料寬度當然是越大越好。但傳統的匯流排寬度一般大約僅達 64 位元，為了要加大這個寬度，因此晶片組廠商就將兩個(或以上，當用於中階以上伺服器時)主記憶體彙整在一起

2. 這幾支記憶體除了容量大小要一致之外，型號也最好相同

→ 因為啟動多通道記憶體功能時，資料是同步寫入/讀出這幾支主記憶體中，如此才能夠提升整體的頻寬

● 顯示卡

1. 又稱為 VGA (Video Graphics Array)

2. 一般對於圖形影像的顯示重點在於解析度與色彩深度 (受顯示卡記憶體容量影響)

- GPU

- (1) 顯示卡上面嵌入的一個 3D 加速的晶片

- (2) 緣由：3D 的運算早期是交給 CPU 去運作的，但是 CPU 並非完全針對這些 3D 來進行設計的

3. 透過 CPU 的控制晶片來與 CPU、主記憶體等溝通

→ 顯示卡的規格 (資料的傳輸頻寬，適用於其他界面卡插槽)：

PCI → AGP → PCI-Express

註：PCIe (PCI-Express) 使用的是類似管線的概念來處理，管線越多(通常設計到 x16 管線)則總頻寬越高 (主機板上以插槽的長度來看)

4. 目前主要的連接介面

- D-Sub (VGA 端子)

- (1) 15 針的連接、為類比訊號的傳輸、設計是針對傳統映像管螢幕而來

- DVI

- (1) 四種以上的接頭、常見於液晶螢幕的連結

- (2) 台灣市面上比較常見的為僅提供數位訊號的 DVI-D，以及整合數位與類比訊號的 DVI-I 兩種

- HDMI

- (1) 可以同時傳送影像與聲音 → 被廣泛的使用於電視(電腦)螢幕中

- Display port

- (1) 與 HDMI 相似，目前在台灣比較少螢幕的支援

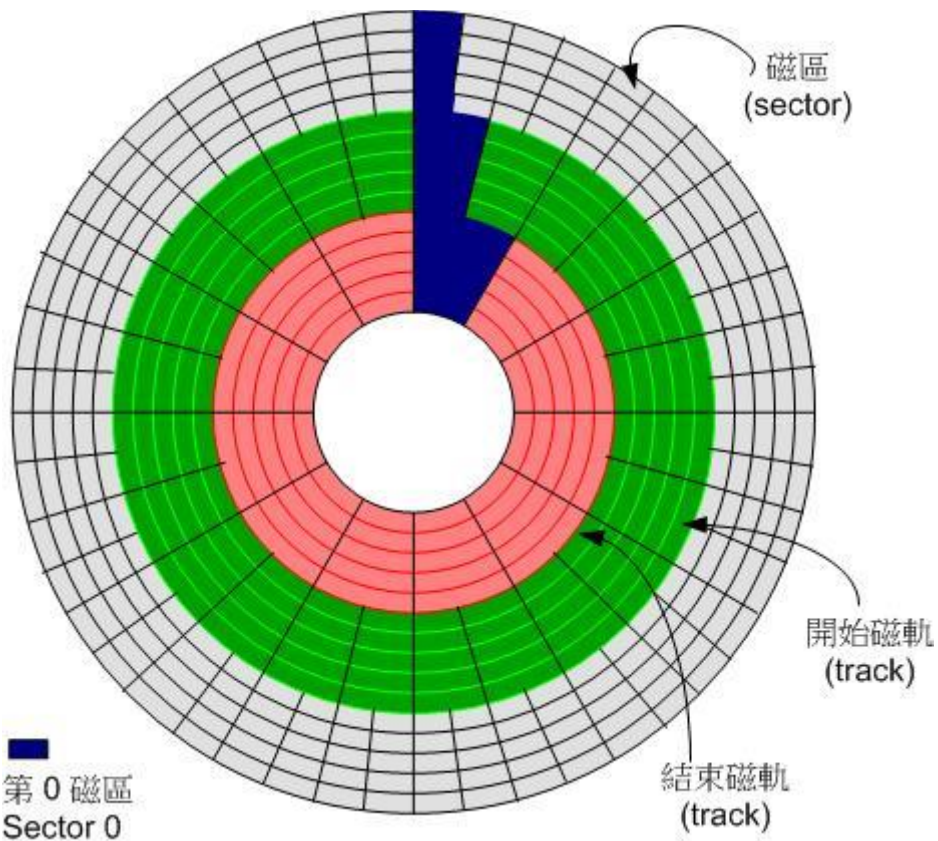
- 硬碟與儲存設備

- 硬碟的物理組成



註：由於單一磁碟盤的容量有限，因此有的硬碟內部會有兩個以上的磁碟盤

○ 磁碟盤上的資料



1. 磁碟盤要轉動才能夠讓機器手臂讀寫

→ 通常資料寫入當然就是以圓圈轉圈的方式讀寫

■ 磁區 (sector)

(1) 磁碟的最小物理儲存單位

■ 磁軌 (track)

(1) 同一個同心圓的磁區組合成的圓

- 磁柱 (cylinder)

(1) 可能會有多個磁碟盤

→ 在所有磁碟盤上面的同一個磁軌可以組合成的柱狀

2. 外圈的磁區數量比較多

→ 通常資料的讀寫預設會由外圈開始往內寫

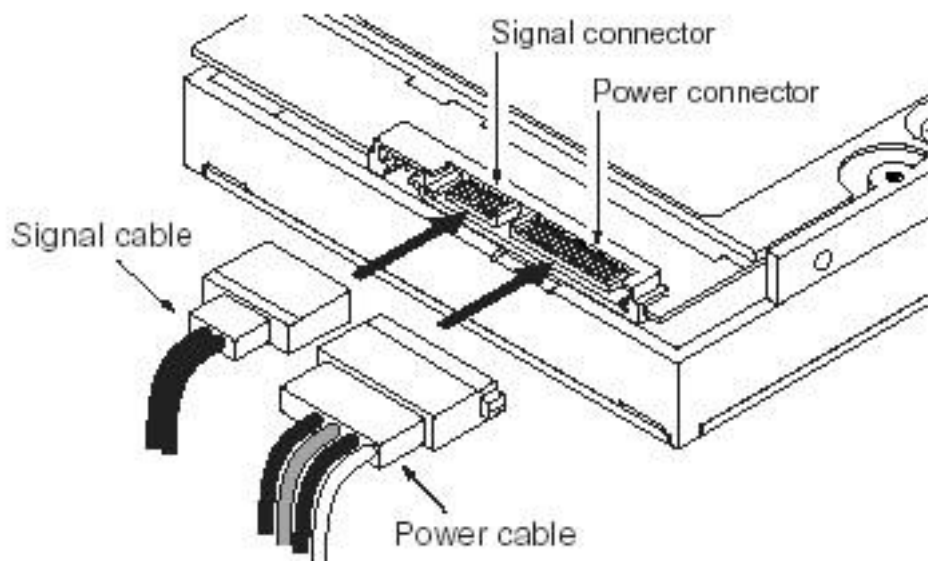
註：原本硬碟的磁區都是設計成 512byte 的容量，但因為近期以來硬碟的容量越來越大，為了減少資料量的拆解，所以新的高容量硬碟已經有 4Kbyte 的磁區設計

→ 在磁碟的分割方面，目前有舊式的 MSDOS 相容模式，以及較新的 GPT 模式

- 傳輸界面

1. 目前 IDE 已經被 SATA 取代，而 SCSI 則被 SAS 取代

- SATA 介面



SATA cabling with separate power and signal attachments

(1) 插槽所使用的排線比較窄小

→ 對於安裝與機殼內的通風都比較好

(2) 每個裝置需要使用掉一條 SATA 線

(3) 每一代都可以向下相容

(4) 透過的資料演算法的關係，當傳輸 10 位元編碼時，僅有 8 位元為資料，其餘 2 位元為檢驗之用

→ 頻寬的計算上面，使用的換算 (bit 轉 byte) 為 1:10 而不是 1byte = 8bits

■ SAS 介面

(1) SCSI：早期工作站或大型大腦上面，為了讀寫速度與穩定度

(2) 串列式 SCSI (Serial Attached SCSI, SAS)

(3) 磁碟盤轉速與傳輸的速度都比 SATA 硬碟好、支援例如熱拔插等功能、作為磁碟陣列卡的連接插槽

(4) 貴 → 一般個人電腦的主機板上面通常沒有內建

■ USB 介面

(1) 磁碟是外接式的介面，很可能跟主機板連結的就是這種介面

(2) 跟 SATA 介面一樣，不是理論速度到達該數值，實際上就可以跑到這麼高

○ 固態硬碟 (Solid State Disk, SSD)

1. 跟傳統磁碟 (Hard Disk Drive, HDD) 不同，傳統的

(1) 需要驅動馬達去轉動磁碟盤，造成很嚴重的磁碟讀取延遲

(知道資料在哪個磁區上 → 命令馬達開始轉 → 讓讀取頭去讀取正確的資料)

2. 以快閃記憶體製作成的高容量設備，也是透過 SATA 或 SAS

3. 透過記憶體直接讀寫的特性

→ 沒資料延遲、快速、省電

4. 早期有『寫入次數的限制』

註：每秒讀寫操作次數 (Input / Output Operations Per Second, IOPS)：

測試磁碟的效能，數值越大，代表可操作次數較高

○ 選購與運轉須知

■ HDD 或 SSD

(1) 目前大多是 SSD 作為系統碟 (系統運作快速)、資料儲存放在 HDD 上面 (資料儲存量大)

■ 容量

(1) HDD 高達 2T、SSD 大約 128~256GB (2015 年)

■ 緩衝記憶體

(1) 將硬碟內常使用的資料快取起來，以加速系統的讀取效能

■ 轉速

(1) 主流的桌上型電腦硬碟為每分鐘 7200 轉，筆記型電腦則是 5400 轉

■ 運轉須知

(1) 電腦通電之後，就絕對不要移動主機

→ 抖動或髒污在磁頭與硬碟盤之間會造成資料的損毀或者是實體硬碟整個損毀

(2) 不要隨便將插頭拔掉就以為是順利關機

→ 機械手臂必須要歸回原位，使用作業系統的正常關機

- 擴充卡與界面

1. PCIe 介面有不同的通道數，基本上常見的就是 x1, x4, x8, x16
2. 個人電腦主機板常見是 x16 的，一般中階伺服器則大多有多個 x8 的界面
 - 多通道卡 (例如 x8) 安裝在少通道插槽 (例如 x4) 的可用性
 - (1) 因為要讓所有的擴充卡都可以安插在主機板上，所以在比較中高階的主機板上，都會做出 x16 的插槽，但是該插槽內其實只有 x8 或 x4 的通道有用！其他的都是空的沒有金手指 (電路)
 - (2) 可以在 x8 或 x4 上插上 x16 的卡，但是效能只會有原本的 $8 / 16 = 1 / 2$ 或 $4 / 16 = 1 / 4$

- 主機板

- 發揮擴充卡效能須考慮的插槽位置
 1. 跟 CPU 直接連線的那幾個插槽，效能最佳
 2. 插槽與南橋連接時，擴充卡資料需要先進入南橋跟大家搶頻寬，之後要傳向 CPU 時，還得要透過 CPU 與南橋的溝通管道，那條管道稱為 DMI 2.0
 3. 詳細閱讀您主機板上面的邏輯圖示，尤其 CPU 與南橋溝通的頻寬方面
- 設備 I/O 位址與 IRQ 中斷通道
 - I/O 位址
 - (1) 類似每個裝置的門牌號碼
 - (2) 不能有兩個裝置使用同一個 I/O 位址
 - IRQ 中斷 (Interrupt)

(1) 可以想成是各個門牌連接到郵件中心(CPU)的專門路徑

(2) 透過 IRQ 中斷通道來告知 CPU 該裝置的工作情況

○ CMOS 與 BIOS

■ CMOS

(1) 一種積體電路的設計製程，早期的唯讀記憶體主要以這種電路製作，∴ 當時電腦系統的 BIOS 程式和參數信息都保存在 ROM 中

→ 在很多情況下，當人們提到「CMOS」時，實際上指的是電腦系統之中的 BIOS 單元

(2) 主要的功能為記錄主機板上面的重要參數，包括系統時間、CPU 電壓與頻率、各項設備的 I/O 位址與 IRQ 等

(3) ∴ 這些資料的記錄要花費電力

→ 主機板上面有電池

■ BIOS (Basic Input Output System)

(1) 一個韌體 (韌體就是綁在硬體上面的控制軟體)

(2) 系統在開機的時候首先會去讀取的一個小程式

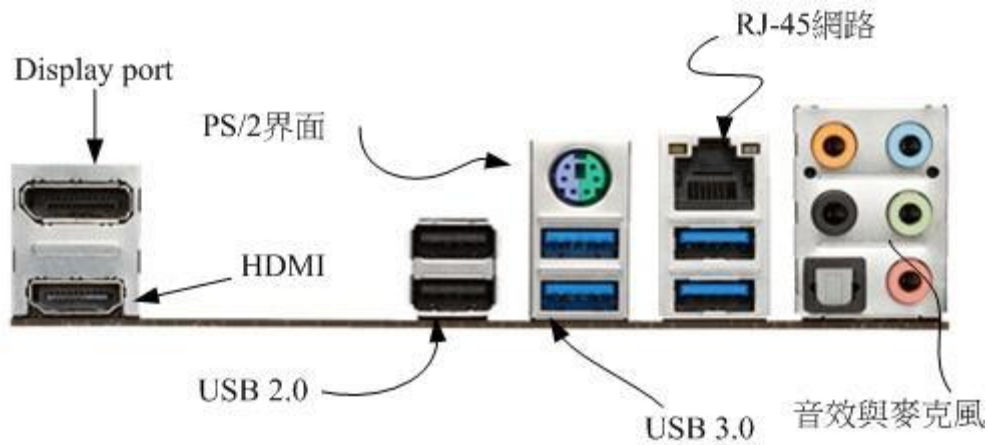
(3) 載入 CMOS 當中的參數，並嘗試呼叫儲存裝置中的開機程式，進一步進入作業系統當中

(4)

∴ 電腦發展的速度太快，BIOS 程式碼也需要作適度的修改！但是 BIOS 原本使用的是無法改寫的 ROM

→ 現在的 BIOS 通常是寫入類似快閃記憶體 (flash) 或 EEPROM

○ 連接周邊設備的介面



■ PS/2 介面

(1) 原本是常見的鍵盤與滑鼠的介面，目前漸漸被 USB 介面取代

■ USB 介面

(1) 通常只剩下 USB 2.0 與 USB 3.0 (藍色插槽)

■ 聲音輸出、輸入與麥克風

(1) 主機板上需內建音效晶片

■ RJ-45 網路頭

(1) 主機板上需內建網路晶片，內部有八蕊線

(2) 接上網路線後在這個接頭上要有燈號亮起才對

■ HDMI

(1) 主機板上需內建顯示晶片

● 電源供應器 (Power)

1. 用料不同，電源供應的穩定度也會差很多

○ 能源轉換率

(1) 電源供應器本身也會吃掉一部份的電力

(2) 輸出的功率 / 輸入的功率

→ 數值越高表示被電源供應器『玩掉』的電力越少

- 選購須知

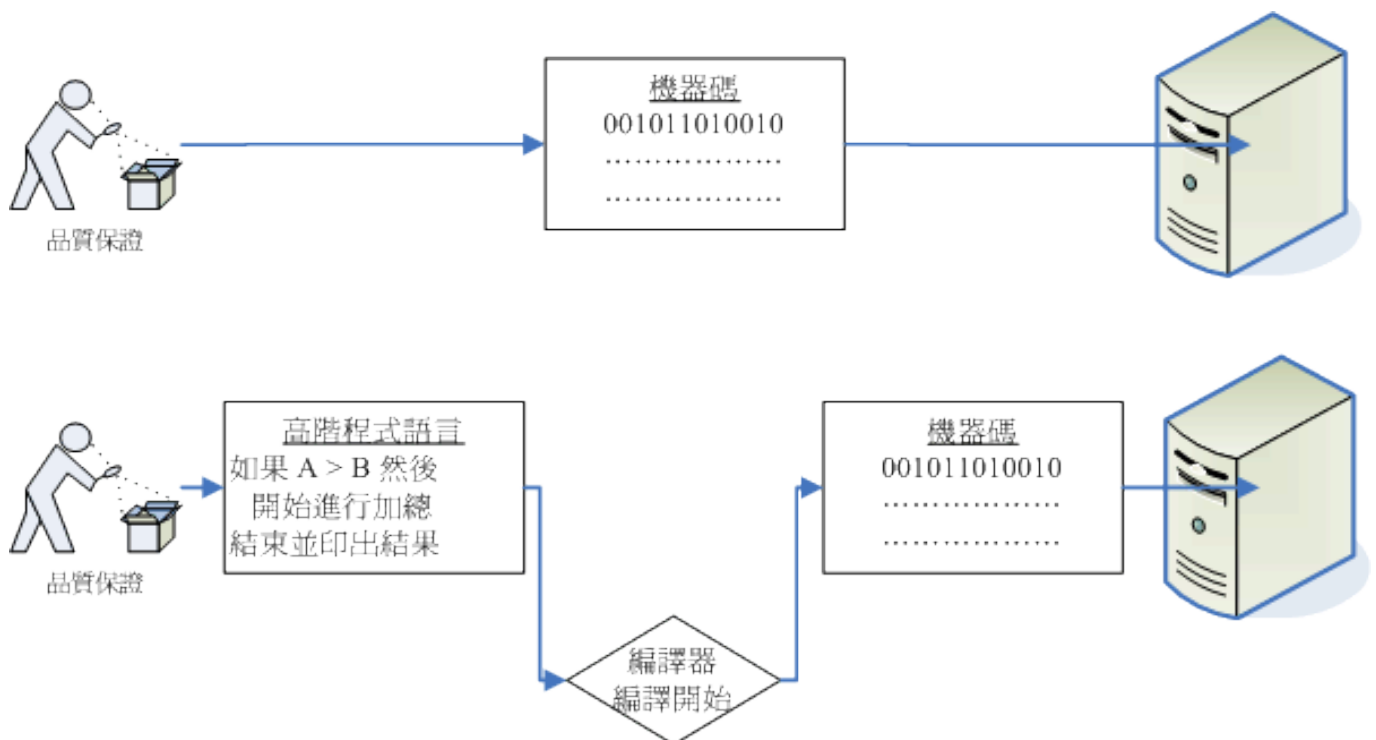
1. 效能並非僅考量 CPU 的能力而已，速度的快慢與『整體系統的最慢的那個設備有關！』

- 系統不穩定的可能原因

- (1) 系統超頻
- (2) 電源供應器不穩
- (3) 記憶體無法負荷
- (4) 系統過熱

- 軟體程式運作

- 機器程式與編譯程式



1. 機器程式缺點：

(1) 需要了解機器語言

∴ 機器只認識 0 與 1

(2) 需要了解所有硬體的相關功能係數

→ 參考機器本身的功能，然後針對該功能去撰寫程式碼

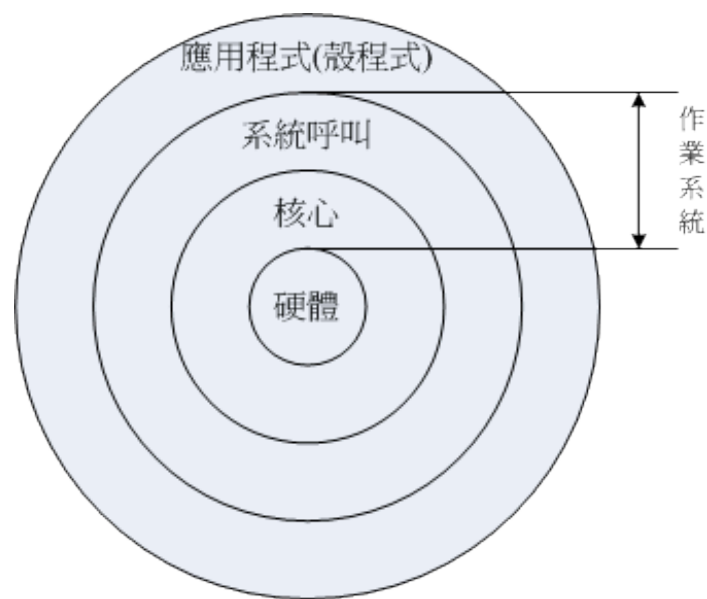
(3) 程式不具有可攜性

→ 為 A 電腦寫的程式，理論上是沒有辦法在 B 電腦上面運作的

(4) 程式具有專一性

→ 每寫一個新的程式，都需從頭再參考硬體的功能函數來繼續撰寫

○ 作業系統 (Operating System, OS)



1. 將所有的硬體都驅動，並且提供一個發展軟體的參考介面來給工程師開發軟體

2. 如果沒有其他的應用程式輔助，作業系統只能讓電腦主機準備妥當 (Ready)而已

3. 應用程式的開發都是參考作業系統提供的開發介面，所以該應用程式只能在該作業系統上面運作而已

■ 作業系統核心 (Kernel)

(1) 其實也是一組程式，重點在於管理電腦的所有活動以及驅動系統中的所有硬體

(2) 核心程式所放置到記憶體當中的區塊是受保護的！並且開機後就一直常駐在記憶體當中

(3) 直接參考硬體規格寫成

→ 同一個作業系統程式不能夠在不一樣的硬體架構下運作

■ 系統呼叫(System Call)

(1) 為了保護核心，並且讓程式設計師比較容易開發軟體，因此作業系統通常會提供的一整組開發介面

(2) 軟體開發工程師只要遵循公認的系統呼叫參數來開發軟體，該軟體就能夠在該核心上頭運作

■ 核心功能

➤ 系統呼叫介面 (System call interface)

➤ 程序管理 (Process control)

1) 『多工環境』：讓 CPU 的資源作有效的分配

2) 良好的 CPU 排程機制：就是 CPU 先運作那個工作的排列順序

➤ 記憶體管理 (Memory management)

➤ 檔案系統管理 (Filesystem management)

1) 資料的輸入輸出 (I/O) 等等的工作

2) 不同檔案格式的支援

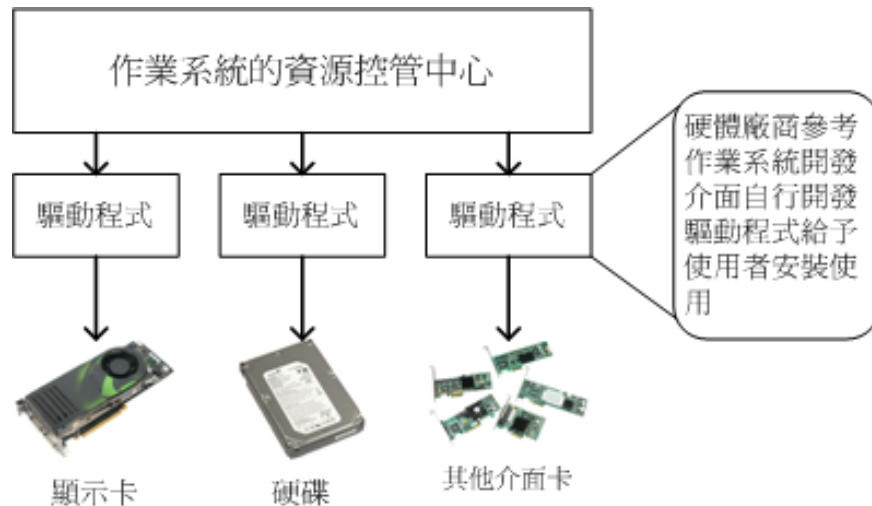
➤ 裝置的驅動 (Device drivers)

1) 目前都有所謂的『可載入模組』功能，可以將驅動程式編輯成模組，就不需要重新的編譯核心

註：硬體廠商要推出硬體時，應該要自行參考作業系統的驅動程式開發介面，開發完畢後將該驅動程式連同硬體一同販賣給使用者

■ 作業系統與驅動程式

- 作業系統必須要能夠驅動硬體，應用程式才能夠使用該硬體功能
- 作業系統會提供開發介面，讓開發商製作他們的驅動程式
- 驅動程式是由廠商提供的，與作業系統開發者無關



○ 應用程式

1. 參考作業系統提供的開發介面所開發出來的軟體
2. 與作業系統有關係

→ 想要購買新軟體，請務必參考軟體上面的說明，看看該軟體是否能夠支援你的作業系統

● 參考資料：

鳥哥的 Linux 私房菜 -- 第零章、計算機概論