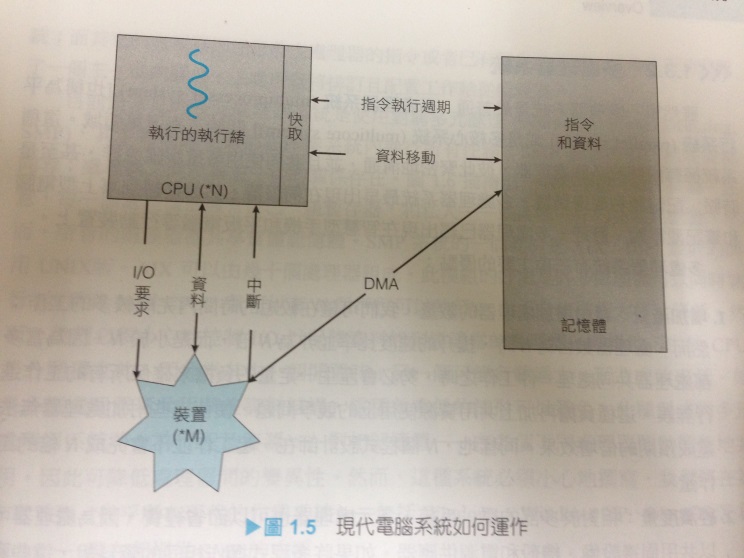
# OS Concept

# Ch1 Introduction:

1. 電腦系統架構：
   1. 電腦啟動 🡺 起始程式(Bootstrap program)
   2. Bootstrap program 通常存儲在ROM 或 EEPROM 又稱為firmware
2. 記憶體：
3. 主記憶體：CPU從記憶體載入程式，大部分來自於可複寫的記憶體
4. RAM Random-Access Memory
5. ROM Read Only Memory
6. EEPROM 電子式可消除程式化唯讀記憶體
7. 輔助記憶體：通常主記憶體太少，不能儲存所有程式，而且是揮發性儲存裝置(Volatile storage)，內容會消失，因此需要搭配非揮發式儲存裝置(nonvolatile storage)當作輔助記憶體
   1. HDD
   2. SSD
   3. CD-ROM
8. I/O:
   1. 每個I/O透過裝置控制器與CPU溝通；在OS內，每個裝置控制器必須有相對應的裝置驅動程式(Device driver)。
   2. 每一個裝置控制器會有緩衝儲存區域，當I/O做動時，控制器會去占存器讀取內容，再將執行動作轉移到緩衝儲存區域
   3. 資料完成轉移，控制器發送中斷訊息給CPU

🡺 但是大量資料移動負擔會變大，因此通常會使用DMA設定緩衝區

* 1. DMA (Direct Memory access): 將I/O緩存區資料直接傳送進入記憶體，每一個區段只中斷一次，提高效能。

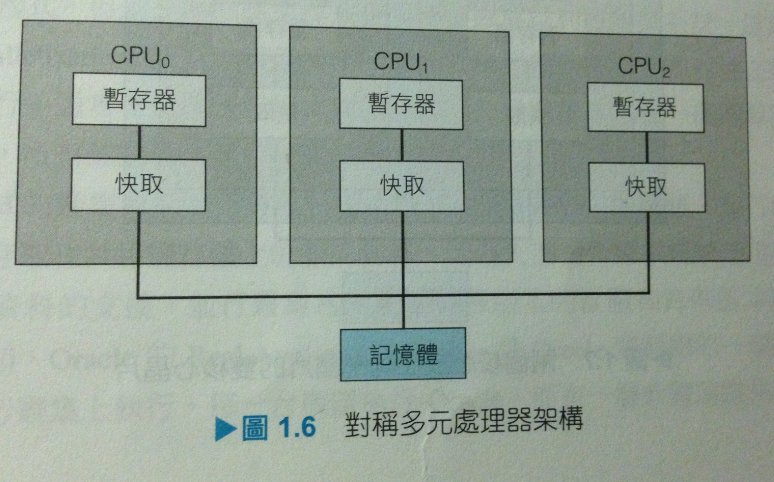
4. 多處理器系統(multiprocessor system A.K.A. parallel system or multicore system):

1. 非對稱多元處理(Asymmetric multiprocessing):

同時處理不同任務，處理器間具有主從關係

2. 非對稱多元處理(Symmetric multiprocessing SMP):

處理器間都是同等，並共享記憶體。



1. 多元處理器的記憶體存取方式：
   1. 一致記憶體存取(Uniform memory access)：

所有處理器取得記憶體資料的花費時間相同

* 1. 非一致記憶體存取(non-Uniform memory access)

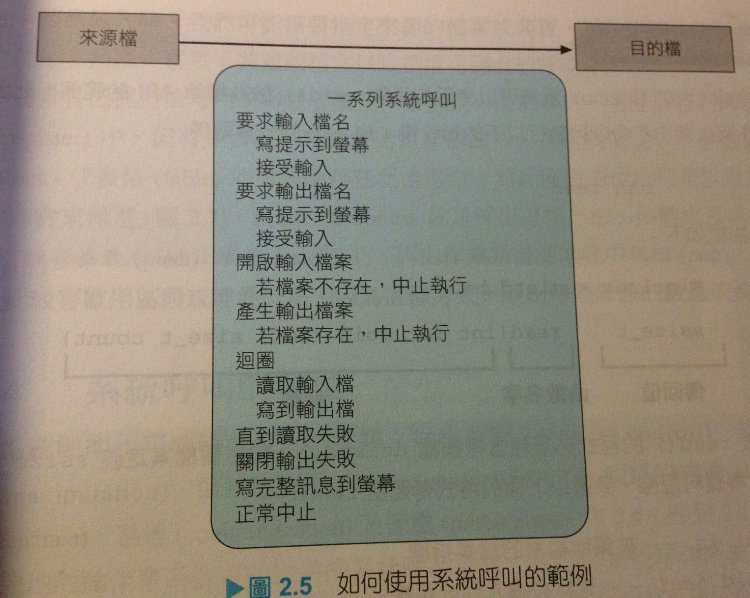
所有處理器取得記憶體資料的花費時間不同

5. 叢集式系統

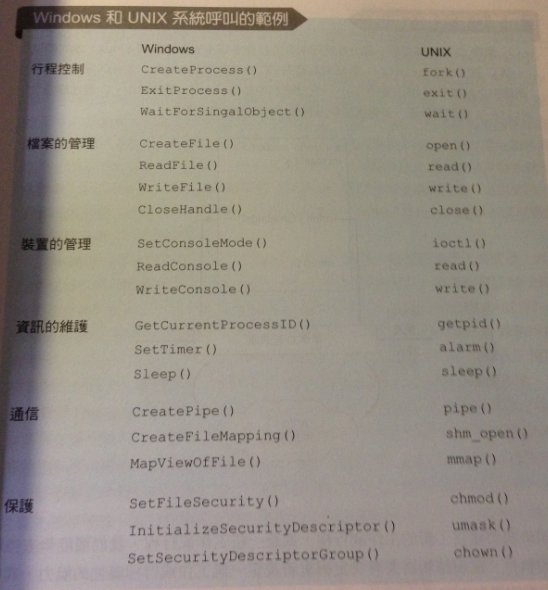
# Ch1 系統結構:

* 1. **使用者介面與作業系統介面：**
     1. 命令直譯程式(Command interpreter):
        1. 命令直譯程式稱為Shells
     2. 圖形使用者介面
  2. **系統呼叫(System call):**

通常是C 或C++寫成的常式，直接呼叫作業系統的提供的服務，如複製檔案，複寫檔案…等。一般是由作業系統之Application Programming Interface(API) 提供system call的函式給使用者使用



* + 1. **系統呼叫的類型：**
       1. 行程控制(Process control)
       2. 檔案管理(File Manipulation)
       3. 裝置管理(Device Manipulation)
       4. 資訊維護(Information Maintenance)
       5. 通信(Communication)
       6. 保護(Protection)



* 1. **系統程式：**

作業系統可以視為系統程式的聚集。

系統程式(System program) 又稱為系統常式(System Utility)，提供程式開發與執行的環境

* 1. 作業系統除錯：
     1. DTrace：可以對執行程式動態增加探測點，觀測程式System call 之間的互動。
     2. DTrace 是由探測點(Probe)、提供者(Provider)和探測點的消費者(Consumer)組成。
     3. DTrace的消費者是對Probe結果感興趣的程式碼，探測啟動時會執行啟動控制區塊(Enabling control block ECB)，若多個消費者都對同一個感測結果感興趣，則會造成多個EBC開始執行。