

CHAPTER 04 作業系統



4-1 作業系統簡介

4-2 各類作業系統

4-3 CPU排班

4-4 記憶體管理

4-5 檔案系統

4-6 熱門作業系統介紹

4-7 行動裝置作業系統



4-1 作業系統簡介

- ▶ 作業系統是一個程式，負責管理電腦裡的硬體及周邊設備，扮演介於使用者與電腦硬體的中間人。
- ▶ 提供使用者一個方便又有效率的環境，使其能夠執行程式。
- ▶ 電腦系統大致可分為：
 - ▶ 硬體(微處理器、記憶體及輸出入設備)
 - ▶ 作業系統
 - ▶ 應用軟體(常用文書處理軟體、電玩、系統程式等)
 - ▶ 使用者(人或其他電腦)



電腦系統的架構





4-1 作業系統簡介

- ▶ 對**大型系統**而言：為了提供諸多使用者同時使用，作業系統必須著眼於效能方面的考量，分配好使用資源。
- ▶ 對**個人電腦**而言：由於大部分的情況都是電腦由單一使用者掌控，因此，除了必須考量效能之外，資源分配就顯得比較不重要。





4-1 作業系統簡介

- ▶ 另一重點在於**容易使用(user friendly)**，功能的使用必須能讓使用者很快上手。
- ▶ 為了提供使用者親切的介面，只好犧牲一點效能來供應**圖形化介面(GUI)**。





4-1 作業系統簡介

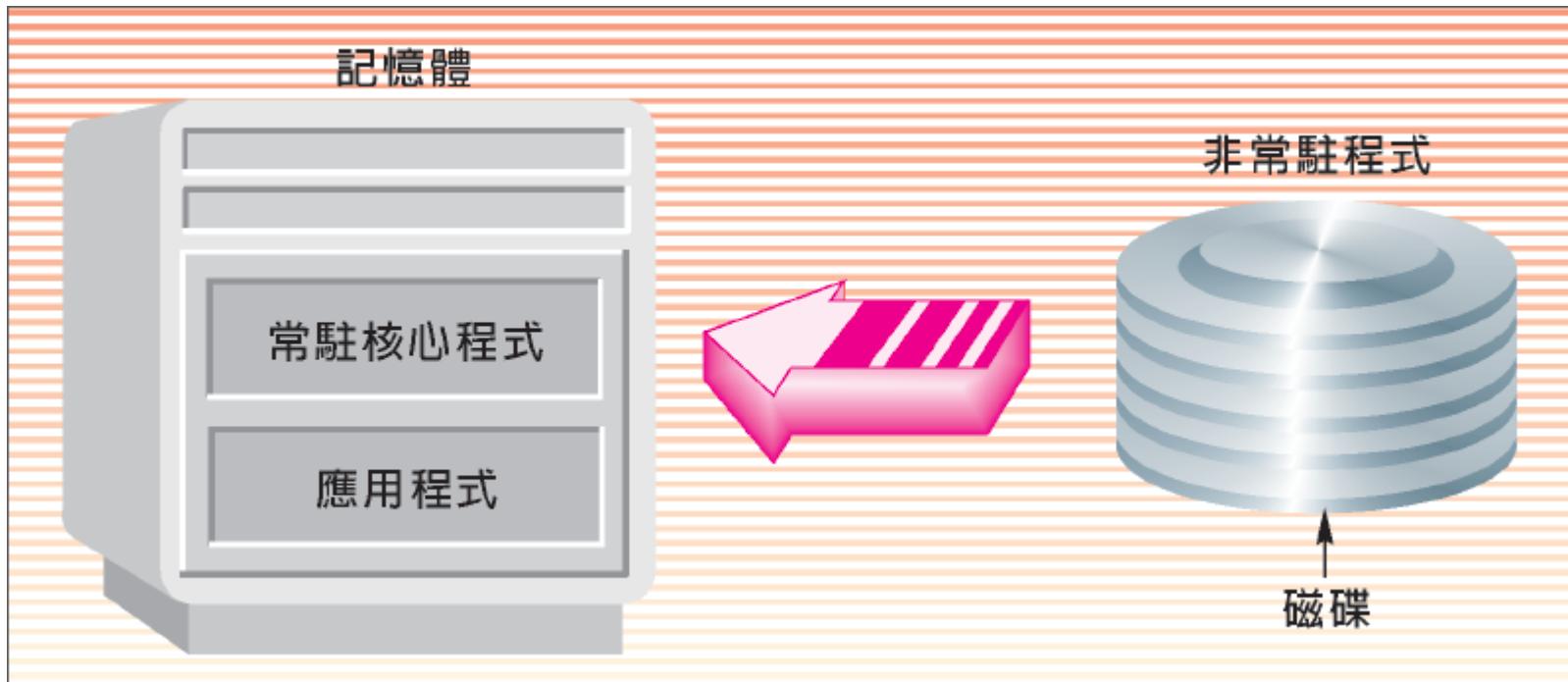
- ▶ 對硬體而言，作業系統是直接接觸的程式，須負責分配資源給有需要的應用程式及使用者。
- ▶ 作業程式應防止因使用者執行程式而導致的錯誤或不正確操作的發生。
- ▶ 不同程式使用到的資源及控制若有交集的部分，應由作業系統來負責。
- ▶ **作業系統**是一個在電腦內部隨時都在執行的核心程式(kernel)，其他則歸類為**應用程式**。



4-1 作業系統簡介

- 整個作業系統是一群程式的集合，其中最重要的部分就是常駐在記憶體的核心程式。
- 核心程式負責把其他部分的作業系統(非常駐程式)在必要時從磁碟中載入至記憶體中。
- 不論使用何種作業系統，當按下電腦電源時，核心程式就負責把其他作業系統載入到記憶體中，此過程稱為**開機**(boot)。





核心程式從磁碟中將非常駐的作業系統載入到記憶體



作業系統負責的工作

- ▶ 作業系統負責的工作
主要有五大項目：





中央處理器管理

- ▶ CPU是整個計算機的核心，當多個程式同時發出CPU需求，作業系統要負責讓CPU充分發揮功能，提高使用效率。
- ▶ 當有一**程序**(process)處於等待狀況，作業系統就讓CPU進行其他程序運算，不能讓CPU閒置。
- ▶ 若有一程序優先順序較目前CPU進行的程序高，作業系統須負責將CPU優先給順序高的程序使用。



記憶體管理

- ▶ 作業系統須將有限的記憶體資源進行合理分配，使每一個程序都能滿足，並提昇效能。
- ▶ 作業系統在分配記憶體時，要讓每個程序都能獨立執行，也要使每個程序共享公共的程序和資料，以避免重複的程式片段和資料佔用，節省記憶體空間。
- ▶ 當程序執行時記憶體空間不夠，作業系統要能做出適當處理，讓目前執行的程序能夠繼續進行。
- ▶ 要防止使用者程序及程序執行時所需要的資料破壞作業系統本身。



檔案管理

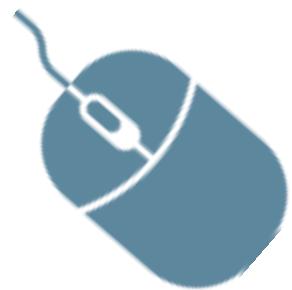
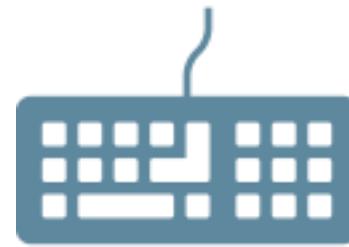
- ▶ 由於主記憶體空間有限，大部分的程序和資料，及作業系統本身都是放在輔助記憶體。
- ▶ 作業系統的檔案管理工作：標示輔助記憶體的資料檔案，有條理地組織訊息，使檔案能夠合理存取與控制，使用者能夠方便而安全地使用。





周邊設備管理

- ▶ 作業系統負責有效管理各種周邊設備，最好支援隨插即用(plug-and-play)，並提供簡易的使用者介面程式，讓使用者即時在對設備不熟悉的情況下，也能循著導引使用設備。





程序管理

- ▶ 每一個程序在作業系統中對應著一個**程序控制表(PCB)**，記載著該程序的相關資訊及狀態。
- ▶ 當程序進入系統後，會被放在**工作佇列(job queue)**中。
- ▶ 就緒並等待執行的程序會被存放在**就緒佇列(ready queue)**，作業系統負責不斷將這些程序交給CPU進行運算。

指標	程序狀態
	程序編碼
	程式計數器
	暫存器
	記憶體限制
	已開放檔案表
	：
	：
	：
程序控制表(PCB)	





4-2 各類作業系統

- ▶ 作業系統須不斷進步，才能使軟體發展跟上硬體的進步速度，以便充分應用新硬體。

主機型
系統

個人電腦
系統

多處理器
系統

分散式系統

即時系統

手持系統





主機型系統

手動操作
階段

批次系統

多程式
規劃系統

分時系統



手動操作階段

- ▶ 早期電腦體積非常龐大，且沒有軟體，也沒有系統負責控制。
- ▶ 輸入裝置可能是讀卡機和磁帶機，輸出裝置則為印表機、打孔機或磁帶機。
- ▶ 使用者獨占計算機的所有資源，並直接使用硬體。

電腦資料打孔紙





手動操作階段

- ▶ 執行程式時，要先將計算機初始化，然後啟動控制開關。
 - ▶ **輸入**：將程式、資料和與工作相關的控制訊息打洞在打孔紙內，依序輸入計算機。
 - ▶ **輸出**：計算機接收到輸入之後，開始執行作業。一段時間之後(可能是數分鐘、數小時、數天)產生輸出。
 - ▶ **除錯**：若程式執行有誤，須檢測記憶體和暫存器裡的值。





批次系統 (batch system)

- ▶ 批次系統是在一個工作到下一個工作時，負責自動控制轉換。由於有了作業之間的轉接自動化，因而縮短了作業間等待CPU的時間，較能發揮計算機的處理能力。
- ▶ 批次系統的作業系統常駐在記憶體中。

作業系統

使用者
程式區

簡單的批次系
統記憶體配置





批次系統 (batch system)

- ▶ 作業員把若干相同或類似的工作集合為一整批 (batched)，讀入主記憶體的緩衝區，再轉到磁帶上。
- ▶ 由系統監控程式接手，負責將磁帶上的每個作業循序讀入主記憶體中，交給CPU去運算處理。
- ▶ 一個作業接著一個作業運作，直到整批全部完成，再把下批作業讀入磁帶，以同樣的方式進行處理。



批次系統 (batch system)

- ▶ 由於I/O緩慢的關係，CPU仍經常處於閒置等待的狀態。
- ▶ 批次系統已經允許把所有工作一起放在磁碟中，若能直接存取多件工作，作業系統就可以進行**工作排班(job scheduling)**，進而增進效率。
- ▶ 批次系統讓使用者不必理解計算機的每個細節，就能進行處理。





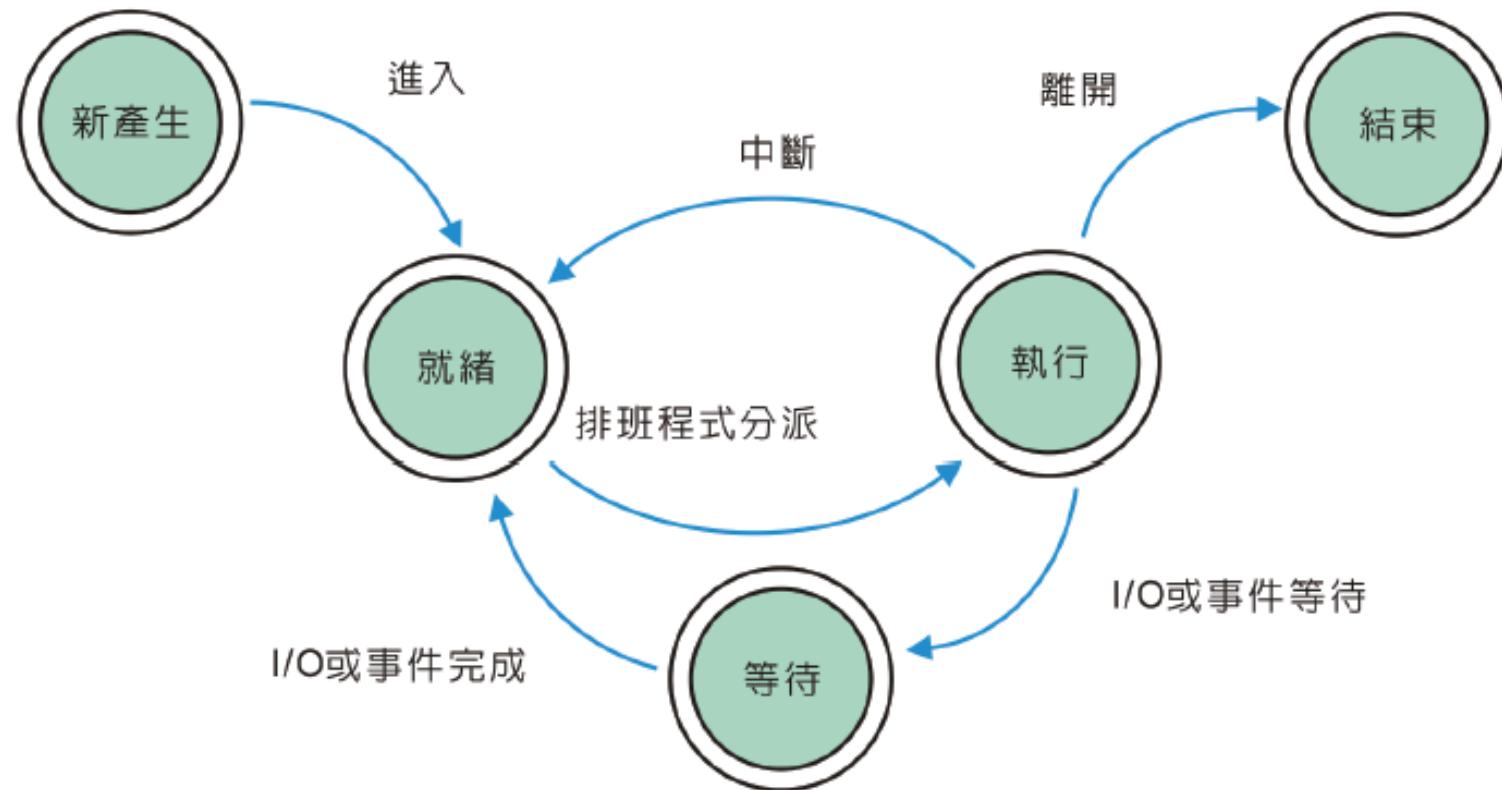
多元程式規劃系統

- ▶ 有了工作排班之後，最重要的就是多元程式規劃 (multiprogramming) 的能力。
- ▶ 多元程式規劃的目的，是要增進CPU的使用率，只要還有程序需要CPU，CPU就不會閒置等待。
- ▶ 一個程式被載入記憶體中並執行時，稱為一個**程序**。





程序的生命週期





程序的生命週期

新產生

- 程序正在產生中。

執行

- 程序得到資源正在執行中。

等待

- 程序在等待某個事件發生(如：等待使用者輸入)。

就緒

- 程序一切已準備就緒，在等待處理器執行。

結束

- 程序已完成。





多元程式規劃系統

- 只要程序進入等待狀態，CPU就轉移到其他工作運作，若原本在等待的程序已經變成就緒，就有機會再重新得到CPU資源。
- 所有工作進入系統後，都放在**工作池(job pool)**，多元程式規劃必須為使用者決定執行優先順序將多個程序放置在記憶體中。

記憶體配置

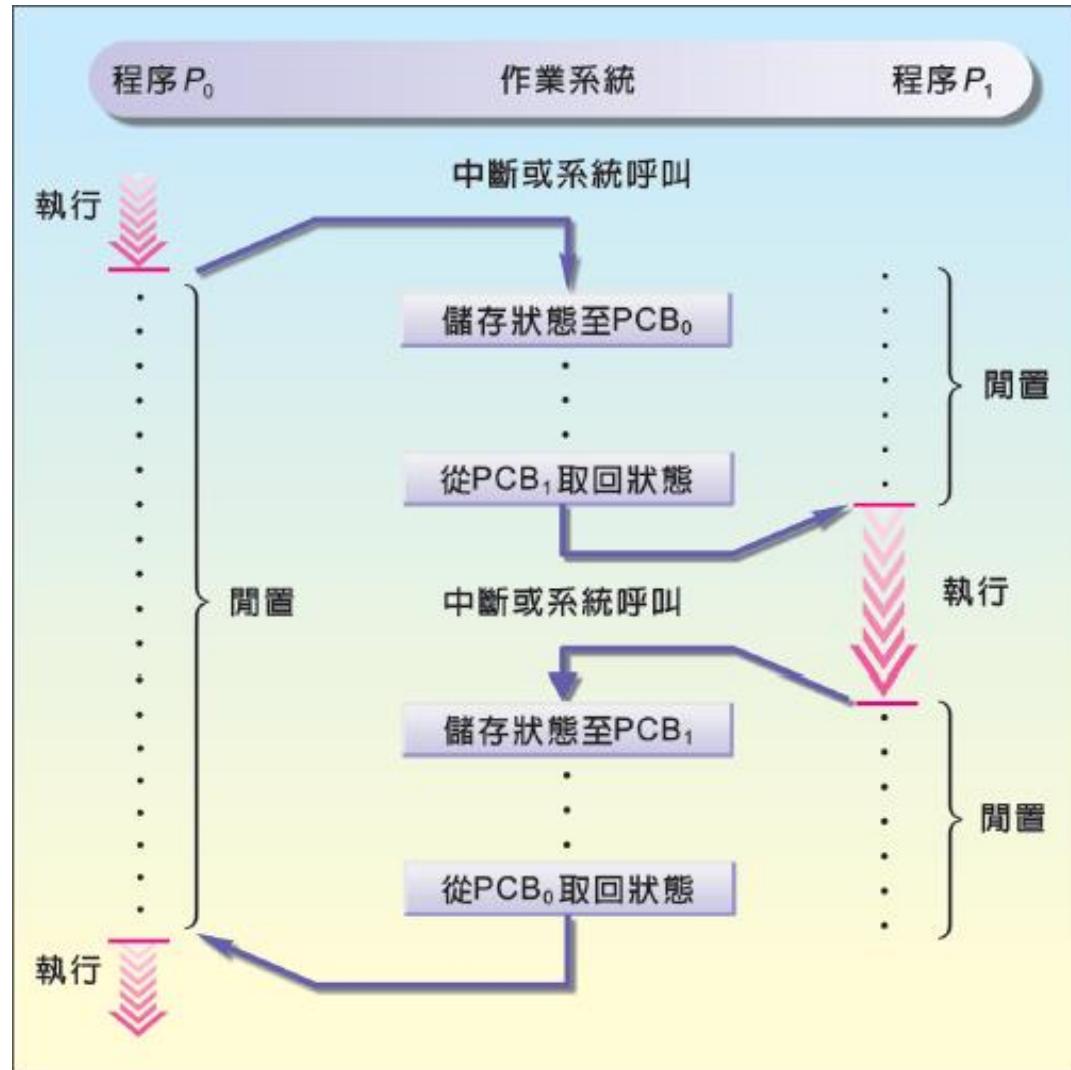




多元程式規劃的排班

- ▶ 當多個程序需要執行時，會先放到大型儲存裝置的工作池中，等候被執行：
 1. **工作排班(job scheduling)**：負責把程序從工作池中選出，載入到記憶體內以便執行。
 2. **處理器排班(CPU scheduling)**：程序進入記憶體後仍須等待，此時使用CPU排班，從記憶體中挑選已就緒的程序，將CPU分配給它使用。
- ▶ 這兩種排班的主要差別是執行次數，CPU排班的執行頻率非常頻繁。





CPU在程序之間來回運轉狀況。

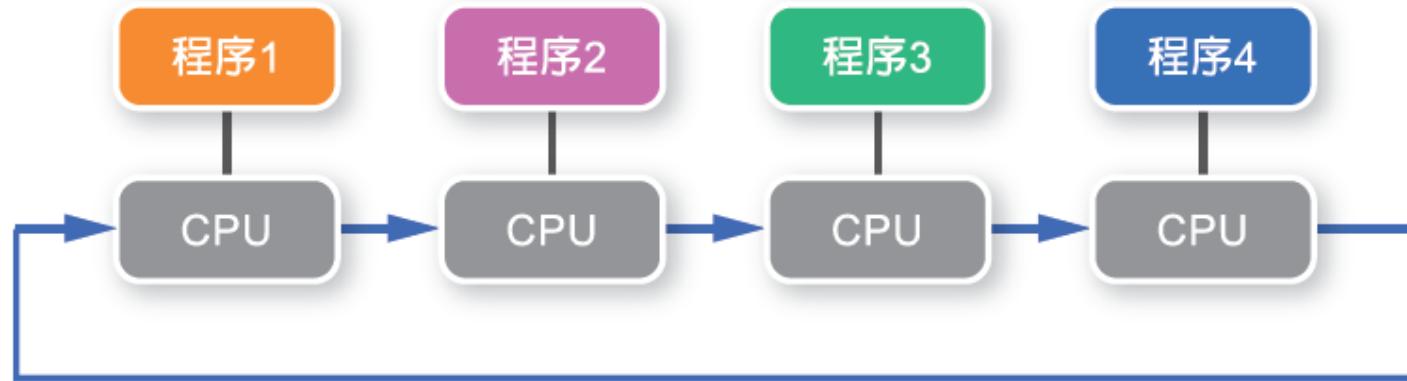
CPU首先為 P_0 進行運算， P_0 進入等待狀況時，作業系統提出中斷或者系統呼叫，接著CPU改為 P_1 進行運算，如此來回轉換。





分時系統

- ▶ 分時系統將CPU的時間切割成很多小時段，CPU在各程序或使用者間不停切換執行。頻繁的切換使每個使用者都以為自己和CPU是持續運作的。



時間一到，CPU就為下一個程序進行運算，如此不停來回處理多個程序





分時系統

- ▶ 數位處理器的運算速度非常快，而人類在輸入或處理資料時，對電腦而言相對太慢，因而有「分時」的概念產生。
- ▶ 連上工作站就是一個分時系統。同一時間點可能有很多使用者同時登入，同時在機器上進行不同的工作，就是拜分時系統之賜。





分時系統的特色

同時性

- 可同時有若干使用者連結到同一台計算機上進行運算。

獨立性

- 不同的使用者之間不會相互干擾。

即時性

- 每一個使用者都可即時得到計算機的回應。





分時系統

- ▶ 分時作業的概念，讓多個使用者共用一部電腦；也讓只用一個處理器的個人電腦，可同時開多個視窗來完成不同任務。
- ▶ 有效且公平的排程是作業系統設計的重點。
- ▶ 多元程式規劃系統倚靠**事件觸發(eventdriven)**；分時系統則是倚靠**時間觸發(time-driven)**。





個人電腦系統

- ▶ 個人系統的目標在於增進使用者操作方便，並提昇CPU回應速度，避免使用者等待。





個人電腦常見的作業系統

DOS

磁碟作業系統

是微軟最早的工作系統產品，用在IBM相容個人電腦上，使用者以指令方式操控電腦。

Mac OS

是第一個圖形化介面的個人電腦工作系統，適用於蘋果電腦上。

Windows

微軟推出的Windows 1.0是PC上的第一個圖形化工作系統。

Linux

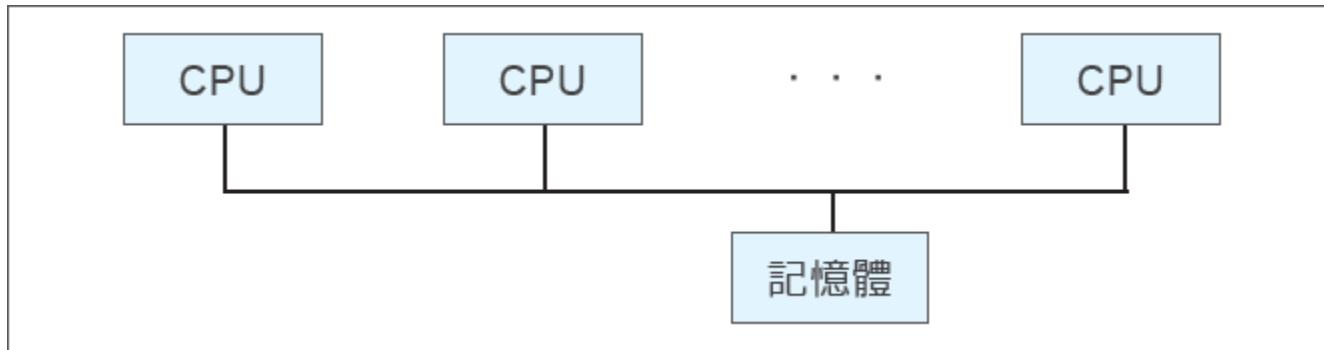
根據大型電腦工作系統UNIX所發展，早期只有命令列模式，現在也有了圖形化使用者介面。





多處理器系統

- ▶ **多處理器系統(multiprocessor system)**擁有一個以上的處理器，彼此間緊密溝通合作、共享資源、共用時脈。



多處理器系統中，多個應用程式共用記憶體等資源



多處理器系統

- ▶ 多處理器系統的最大好處：進行大量運算時，各處理器可以一起分擔工作。
- ▶ 但假設另添加四顆CPU為原來的五倍，但效能不一定就會變成原來的五倍。
- ▶ 因為多處理器同時運作時，要多費功夫保持處理器間的溝通合作正常。
- ▶ 雖然無法呈現等倍數成長，但還是能增進效能。



多處理器系統

- ▶ 想要增進效能，也可採用生產線的模式，使用多台單一處理器系統。但從經濟層面考量，多處理器系統較符合經濟利益。
- ▶ 多處理器一起運作可達到**容錯(fault tolerance)**的目的。
- ▶ 多處理器系統下的容錯，是指各個工作適當分配在不同的處理器上，若其中一個處理器發生問題，系統也不會因此中止，只是速度會變得較慢。



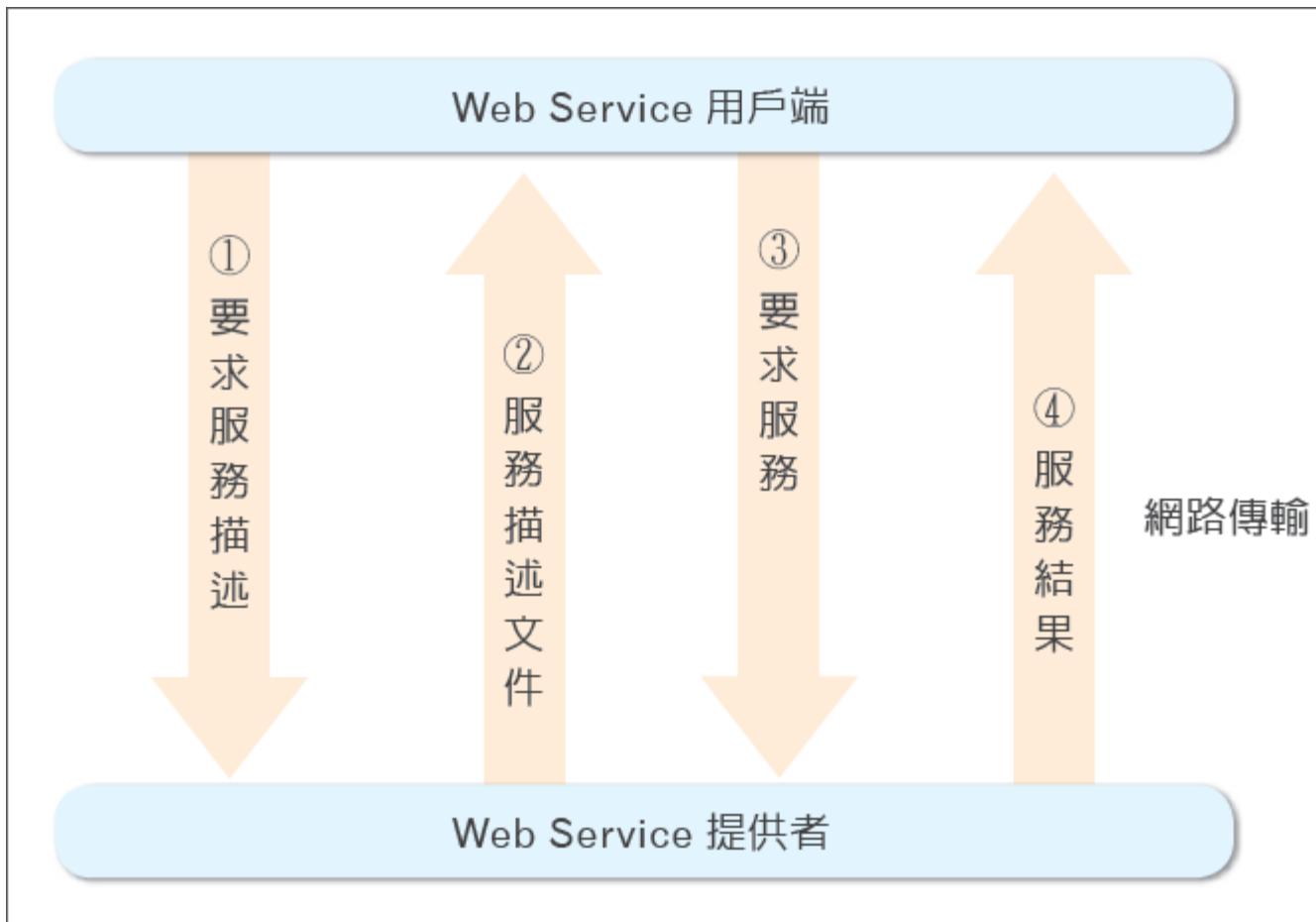
分散式系統

- ▶ 分散式系統是指電腦是分散的，電腦間並不共享資源或時脈，每一個電腦都有自己的記憶體及作業系統，彼此間依靠網路傳輸交換資料。
- ▶ 雖然電腦會透過網路連線使用其他電腦的資料或元件，但是使用者並不會有任何感覺，像是只對單一電腦交談一樣。



Web Service

- ▶ 高速網路環境使分散式作業系統的使用環境越來越成熟。
- ▶ Web Service是將傳統分散式系統演變成網際網路上的標準化服務。
- ▶ 透過網際網路通訊協定及資料格式的開放式標準(例如：HTTP、XML及SOAP等)來為其他應用程式提供服務，達到分散式的效果。



分散式應用的Web Service · 透過網路提供服務



Web Service

- ▶ Web Service能夠提供企業許多便利性。
- ▶ 假設要成立一個旅遊網站，可採用航空公司提供的訂票訂房Web Service，就不需費心理解內部運作，只要設計好介面。程式運作時，就會透過網路連線，使用分散在網路其他位址的服務元件。
- ▶ 由於分散式系統的資料在網路上相互流傳，因此資料的安全性將是一大考量。



即時系統

- ▶ 在即時系統中，計算機要能即時回應外部事件的要求，並於規定時間內完成對該事件的處理，還要控制所有的即時設備與工作能夠協調執行。
- ▶ 即時系統適用於使用者對處理器操作或資料傳輸，有嚴謹的時間限制。
- ▶ 通常用在專門應用範圍的控制裝置，感應器將資料傳送給電腦，電腦必須將資料加以分析，並且控制感應器。



即時系統

- ▶ 使用即時系統可能是用來控制科學實驗、醫學影像系統、或者是工業工程控制。
- ▶ 這些情況下都必須有嚴謹的時間控制，譬如造飛機的時候，生產線上每一個環節的時間點都必須剛剛好，總不能機頭都已經往前推進了，噴槍才開始噴漆，或者更嚴重的情況可能是機器手臂動作太慢，無法跟其他工作協調導致傷害到飛機，這些情況都是不允許發生的。



設計即時系統的重點

即時時鐘管理

- 用以完成定時工作或者延遲工作，以利與其他工作協調。

過載保護

- 當工作太多導致計算機來不及處理時，須有完善的服務策略(例如：使用緩衝區)。

高度可靠性

- 須保障計算機發生問題時，系統仍然正常工作，通常使用的解決方式就是多處理器。



手持系統

- ▶ 隨著技術的進步，個人數位助理(PDA)幾乎都有手機功能，稱為**PDA手機**(PDA phone)；手機也演進到擁有PDA的功能，稱為**智慧型手機**(smart phone)。
- ▶ 兩者目前沒有嚴謹的定義區隔。





主流高階手機作業系統



Android



iOS



Windows
Mobile





手持式系統的三項特點

記憶體容量小

- 所以作業系統與應用程式必須更有效管理記憶體。

處理器運算緩慢

- 運算越快的電腦相對越耗電，若配置運算能力快的處理器，就須使用較大顆的電池。

顯示螢幕小

- 使用者介面設計須格外留意，讓使用者盡可能看到電子郵件或網頁內容。

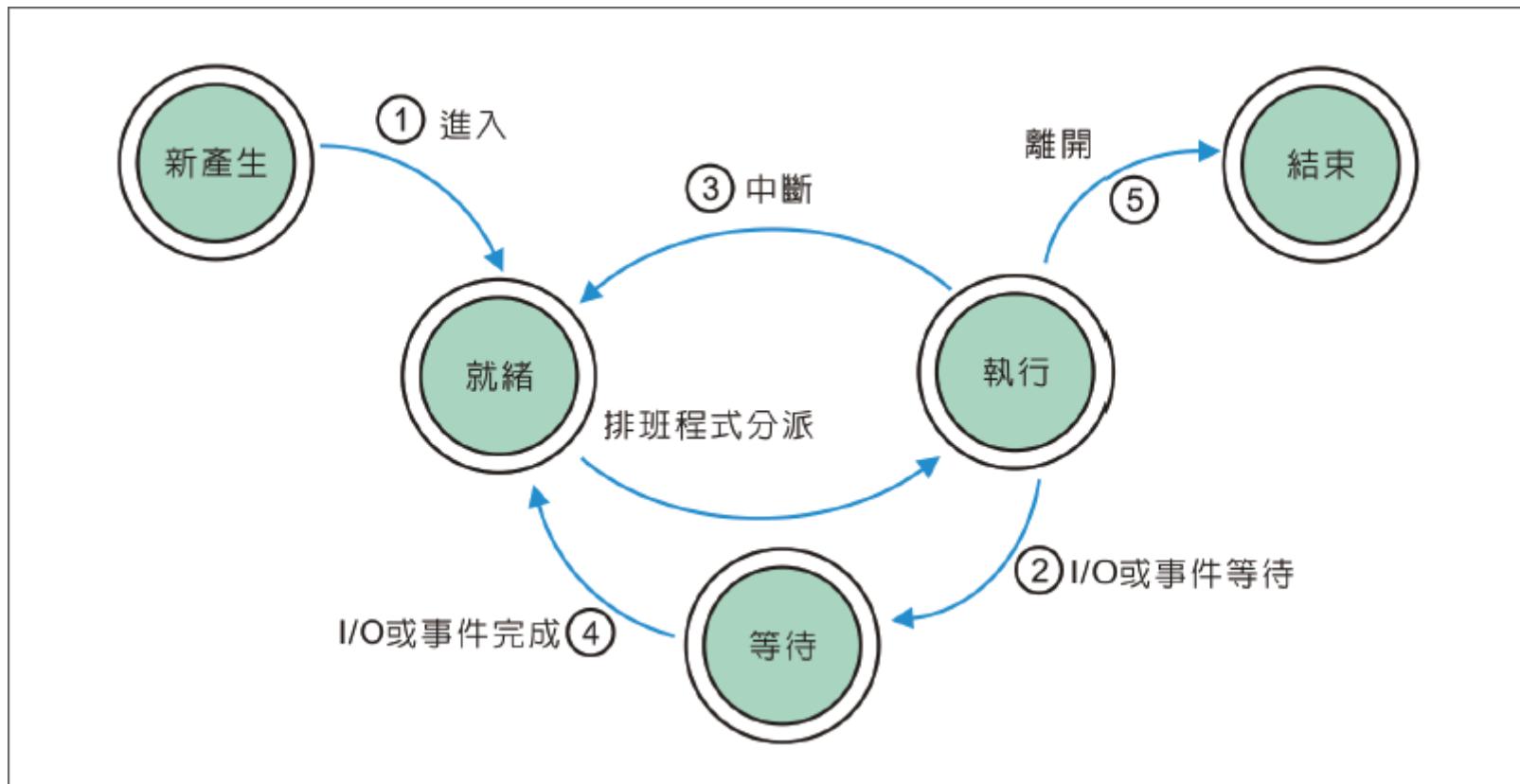


4-3 CPU排班

- ▶ 多元程式規劃作業系統是藉著**CPU排班**來保持隨時都有一個程序在執行，以提高CPU使用率。
- ▶ 藉由CPU在不同的程序中轉換，各個程序就可以盡可能地榨乾CPU的運算能力。
- ▶ 單一處理器系統中，一次只能有一個程序被執行，其他的都得依照CPU排班在旁等待。



CPU排班的決策時間點





CPU排班的決策時間點

- 1 • 程序新產生時。
- 2 • 程序從執行狀態變成等待狀態(譬如I/O要求)。
- 3 • 程序從執行狀態變成就緒狀態(譬如中斷發生時)。
- 4 • 程序從等待狀態變成就緒狀態(譬如I/O要求得到回應)。
- 5 • 程序中止時。



4-3 CPU排班

- ▶ **不可搶先排班(nonpreemptive)**：排班方式確保已享有CPU資源的程序能夠一直用，不管其他程序的狀態，直到程序自己跳到其他非執行的狀態，才進行排班。
- ▶ **可搶先排班(preemptive)**：時時刻刻都要注意所有程序的狀態，若有程序進入就緒狀態，就要相互比較已就緒程序與正在使用CPU程序的優先順序。



CPU排班演算法的評估標準

- ▶ **CPU使用率**：理論上CPU使用率最好達100%，但實際應用上，CPU不可能到達此極限。
- ▶ **產能(throughput)**：CPU越忙碌，產能就越高。產能是衡量單位時間CPU可完成多少程序
- ▶ **回復時間(turnaround time)**：指一個程序從進入電腦開始至CPU執行所花費的單位時間。
- ▶ **等待時間**：算計程序會花多久的時間在就緒佇列中等待。



CPU排班演算法

先到
先處理

最短工作
先處理

優先權
排班

依序循環
排班



先到先處理

- ▶ 先到先處理(FCFS)是最簡單的方式，就是把CPU分給第一個需要CPU資源的程序。
- ▶ 可用有先進先出(first in first out)特性的佇列來排置程序。CPU會先運算佇列中的第一個程序，並將它從就緒佇列中踢除；當有新程序需CPU資源時，就被串接到就緒佇列的末端。
- ▶ FCFS排班方式是不可搶先的。





先到先處理

- 假設程序的資訊如下所示，時間以毫秒為單位：

程序	抵達順序	所需時間（毫秒）
P ₁	1	15
P ₂	2	3
P ₃	3	3

- 使用FCFS方式，CPU工作表會如下甘特圖(gantt chart)所示：





先到先處理

- 各程序的等待時間如下表所示：

程序	等待時間（毫秒）	平均等待時間（毫秒）
P ₁	0	
P ₂	15	
P ₃	18	11





先到先處理

► 假設三個程序來的順序不同，就有不同結果：

程序	抵達順序	所需時間（毫秒）
P ₁	3	15
P ₂	1	3
P ₃	2	3

► 這種情況下的甘特圖如下所示：





先到先處理

- 各程序的等待時間則為：

程序	等待時間（毫秒）	平均等待時間（毫秒）
P_1	6	
P_2	0	
P_3	3	3

- 同樣三個程序，因抵達順序不同，平均等待時間就相差很大。因此雖然先到先處理方式在實作上很容易完成，卻不是最理想的排班方式。





最短工作先處理

- ▶ **最短工作先處理(SJF)**是指當CPU排班在挑選下一個執行程序時，會檢視所有需要CPU資源的程序各會花多少時間完成工作。最短的程序就能先得到CPU資源，其餘則等下一次挑選時再行比較。
- ▶ 如果有兩個以上的程序所需時間相同，則採用先到先服務的方式挑選。

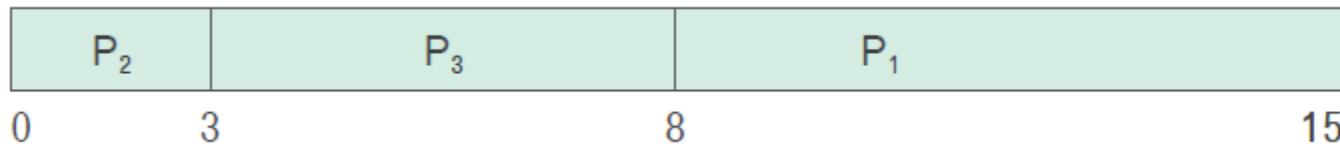


最短工作先處理

► 假設程序的資訊如下所示，時間以毫秒為單位：

程序	所需時間（毫秒）
P_1	7
P_2	3
P_3	5

► 這個例子的甘特圖如下所示：





最短工作先處理

► 各程序的等待時間則為：

程序	等待時間（毫秒）	平均等待時間（毫秒）
P_1	8	
P_2	0	
P_3	3	3.67





最短工作先處理

- 使用FCFS的程序先後順序及其平均等待時間如下：

程序先後順序	平均等待時間
$P_1 \rightarrow P_2 \rightarrow P_3$	5.67
$P_1 \rightarrow P_3 \rightarrow P_2$	6.33
$P_2 \rightarrow P_1 \rightarrow P_3$	4.33
$P_2 \rightarrow P_3 \rightarrow P_1$	3.67
$P_3 \rightarrow P_1 \rightarrow P_2$	5.67
$P_3 \rightarrow P_2 \rightarrow P_1$	4.33

- 除了 $P_2 \rightarrow P_3 \rightarrow P_1$ 剛好與最短工作先處理的順序相同，其餘的平均等待時間都比SJF來得長。





最短工作先處理

► 最短工作先處理的方式可分為下列兩種：

CPU執行某工作時，若有新程序產生...

可搶先

- 比較新程序的所需時間。若較目前執行程序的剩餘時間短，新程序則可搶走CPU。

不可搶先

- 新程序須等目前正在執行的程序完成。





最短工作先處理

- ▶ 若單純考慮使平均等待時間降到最低，最短工作先處理的方式是最理想的。
- ▶ 雖然將一個短程序排到前面，會使長程序擠到後面而增加其等待時間，然而卻可以大大減低短程序的等待時間。
- ▶ 難處：實際執行時，很難精準預測每一個程序將花多少時間來完成工作。





優先權排班

- ▶ 優先權(Priority Scheduling)是指為程序們階級化，標上優先順序。
- ▶ 先到先處理的方式並無優先權概念，而最短工作先處理的方式，則屬優先權的特例。

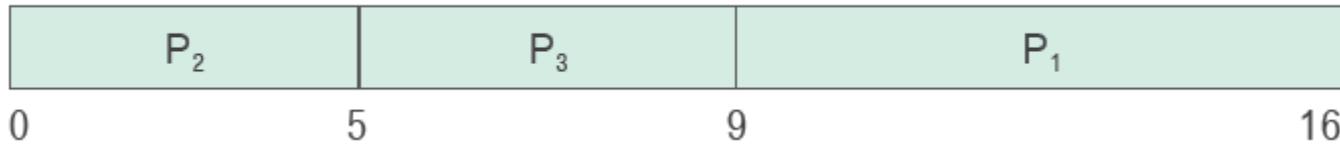


優先權排班

► 假設有如下的程序及其優先權：

程序	優先權	所需時間（毫秒）
P_1	3	7
P_2	1	5
P_3	2	4

► 使用優先權為排班依據，則甘特圖如下：





優先權排班

- ▶ 優先權可從多面向算計，譬如可計算：
 - ▶ 所需時間
 - ▶ 記憶體需求
 - ▶ 程序大小
 - ▶ 程序重要性
- ▶ 優先權排班也可分成可搶先及不可搶先兩種。



優先權排班

- ▶ 優先權排班會因程序始終得不到CPU資源，而變成**飢餓(starvation)**狀況。
- ▶ 工作繁重的電腦系統可能會有一連串高優先權的程序不斷進入就緒佇列，而導致低優先權的程序永遠癱瘓等待。
- ▶ 解決方式：將等待時間也列入優先權設定的考量。
設定等待超過一定時間，就提升該程序的優先權。





依序循環排班

- ▶ 依序循環排班(Round Robin Scheduling)方式是為了分時系統而設計。
- ▶ 基本精神與先到先處理相同，但在使用時先設定好間隔時間(time slice)。
- ▶ 所有程序放在先進先出的佇列裡，首先CPU排班從佇列裡挑第一個程序執行，然後開始進行倒數，時間到時，就讓CPU處理佇列裡下一個程序。





依序循環排班

- ▶ 程序所需時間 < 間隔時間：程序一執行完畢，就自動交出CPU，執行下一個程序。
- ▶ 程序所需時間 > 間隔時間：倒數完畢時，作業系統須將目前執行的內容及狀態記錄下來，寫進**程序控制表(PCB)**中，然後把該程序放到佇列的尾端，從佇列開頭拿取下一個程序執行。



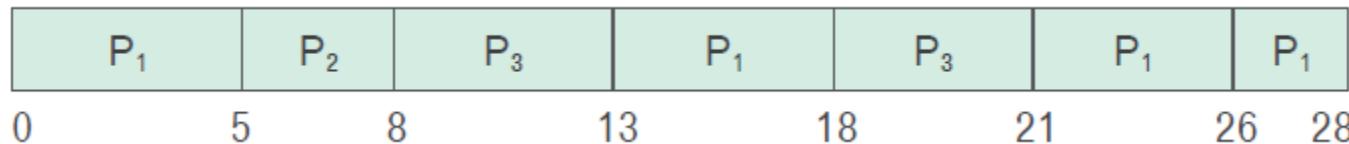


依序循環排班

► 假設設定時間間隔為5毫秒，且有如下程序：

程序	所需時間（毫秒）	抵達順序
P ₁	17	1
P ₂	3	2
P ₃	8	3

► 使用依序循環排班方式之甘特圖如下：





依序循環排班

► 依序循環排班方式的關鍵在於時間間隔的選擇：

程序所需時間：8毫秒	時間間隔(毫秒)	執行次數
	10	1
	5	2
	3	3





依序循環排班

- ▶ 執行次數越多，代表程序轉進轉出CPU的次數越多，就會花較多時間讀取及寫入程序控制表。
- ▶ 若時間間隔設定太小，會導致執行次數太多；若時間間隔設定太大，排班方式會與先進先出相同。
- ▶ 通常使用「**二八原則**」：讓20%的程序所需時間大於時間間隔，而80%的程序所需時間會小於時間間隔。





IT 魔法小百科

什麼是二八原則？二八原則是一個神奇的數字，可以廣泛的應用在經濟學或者是社會學上，意指我們可以用2:8來作為分割對比，這個神奇數字在很多場合都不約而同的能夠適切的形容狀況。譬如我們「20%的人享有地球80%的財富」、「20%的客戶為企業帶來80%的收益」、「20%的人主宰(影響)80%人的命運」、「80%的考題出自課本的20%」，這些都是二八原則的例子，也許下次你可以多留心社會或生活週遭的狀態，會發現有更多二八原則的應用喔！



4-3 CPU排班

- ▶ 無論採用何種排班方式，當程序需要資源時，都會循著以下順序：

請求資源



使用資源



釋放資源



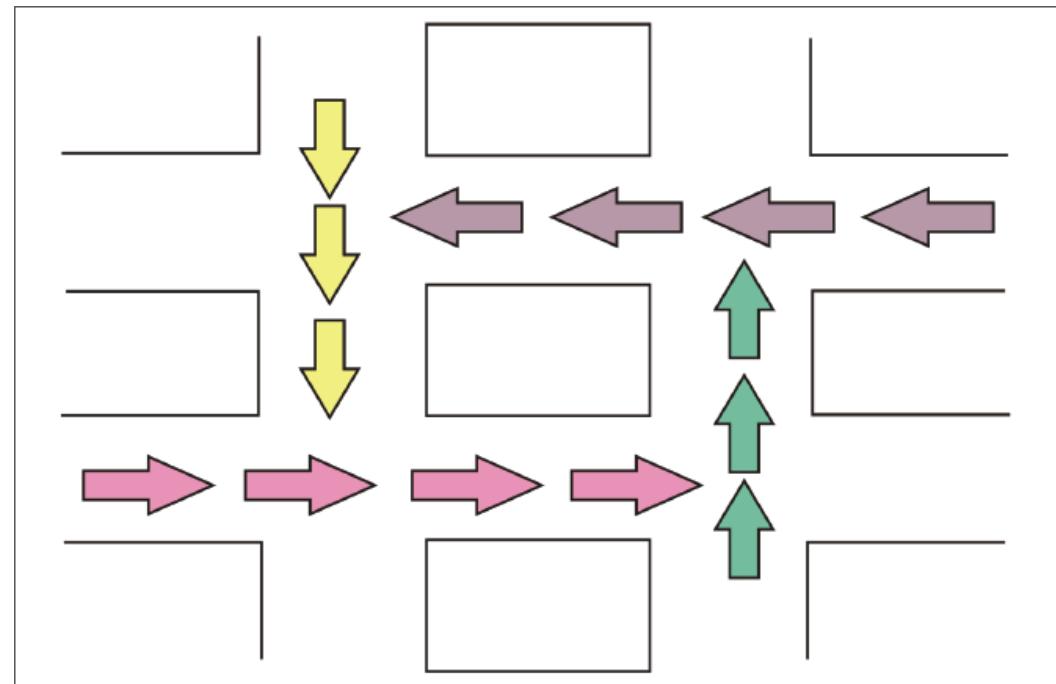
死結

- ▶ 在**多元程式規劃(multiprogramming)**中，程序進到「等待」狀態後，因其所請求的資源，可能被其他同樣在等待狀態中的程序佔用，從此陷入等待狀態無法跳脫，稱為**死結(Deadlock)**。
- ▶ 在死結中的程序，由於每個程序都在等待其他程序觸發事件，但卻沒有程序可以先觸發事件，導致所有程序癱瘓，最終呈現**餓死(starvation)**的狀態。



死結

- 十字路口的車輛都在等待別的車輛空出通道，便可能發生死結。





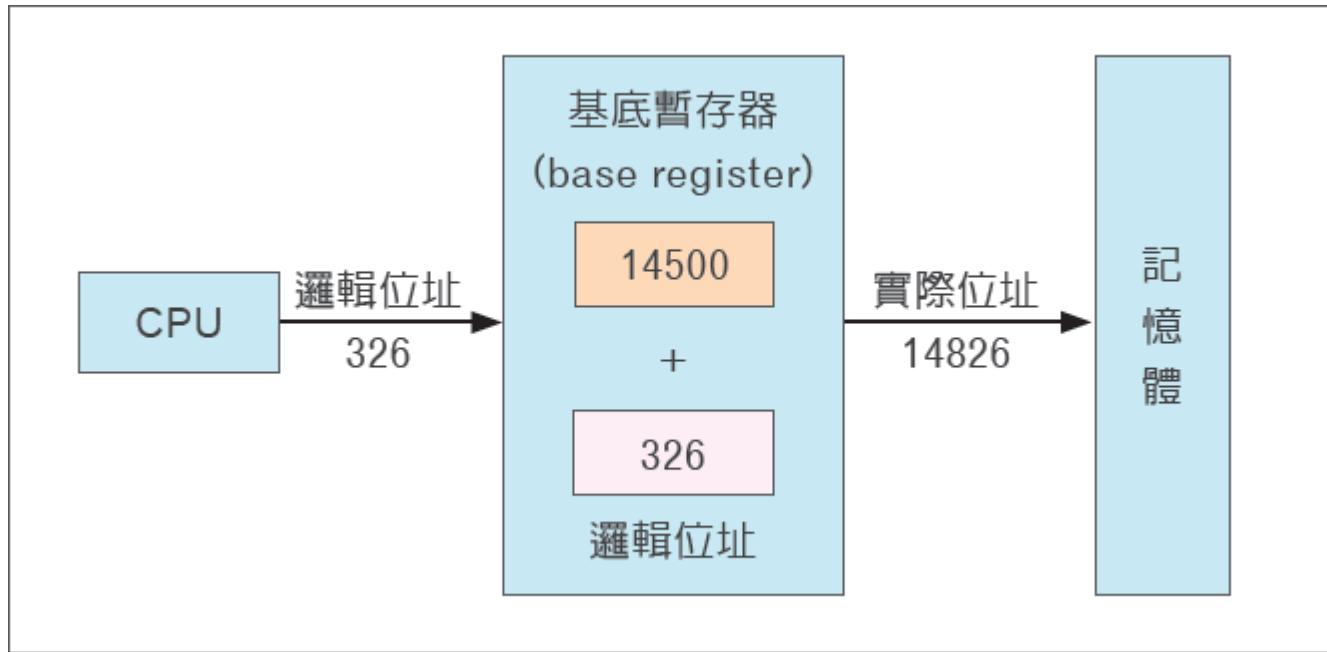
4-4 記憶體管理

- ▶ 記憶體管理主要是讓記憶體能夠充分被利用。
- ▶ 為了使CPU的使用效率增加，我們必須把多個程序放在佇列裡等待，也就是說，我們必須把許多程序放在記憶體中，因此各個程序必須共用記憶體。



記憶體管理的功能

- ▶ 記憶體管理：要把記憶體分割(partition)成各個區塊，分配給各程序，甚至是各使用者使用。
- ▶ 記憶體位址定位：負責把程式所使用的邏輯位址(logical address)與記憶體裡的實際位址(physical address)做映射(mapping)工作。
- ▶ 記憶體保護與共享：要保護好作業系統所使用的記憶體區塊不會被程序破壞。

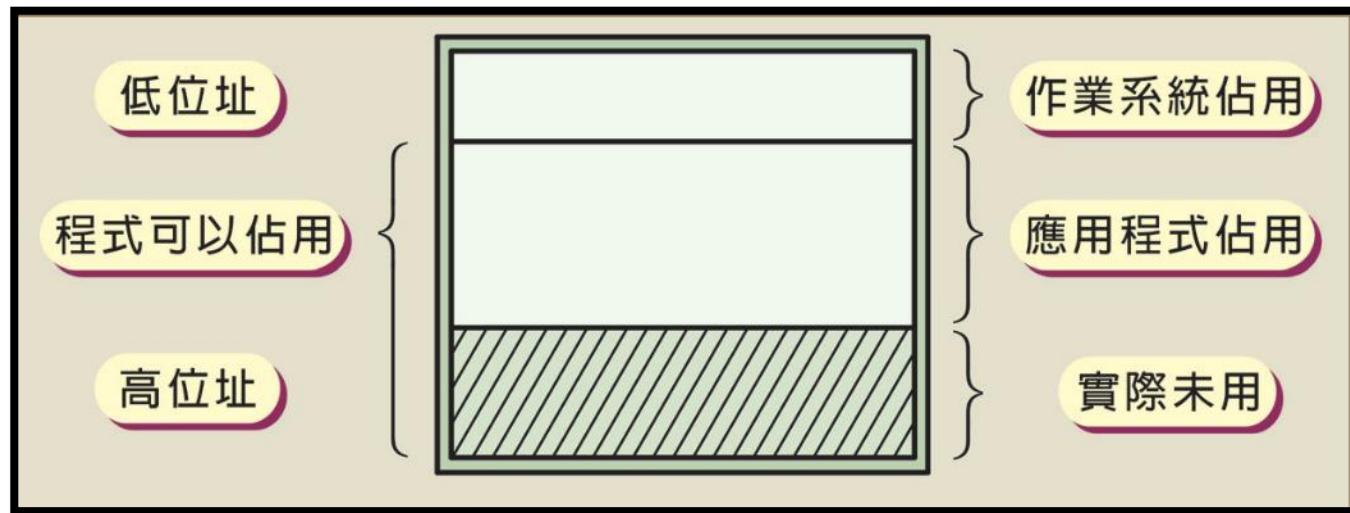


邏輯位址轉為實體位址：透過基底暫存器進行轉換



單一連續記憶體配置方式

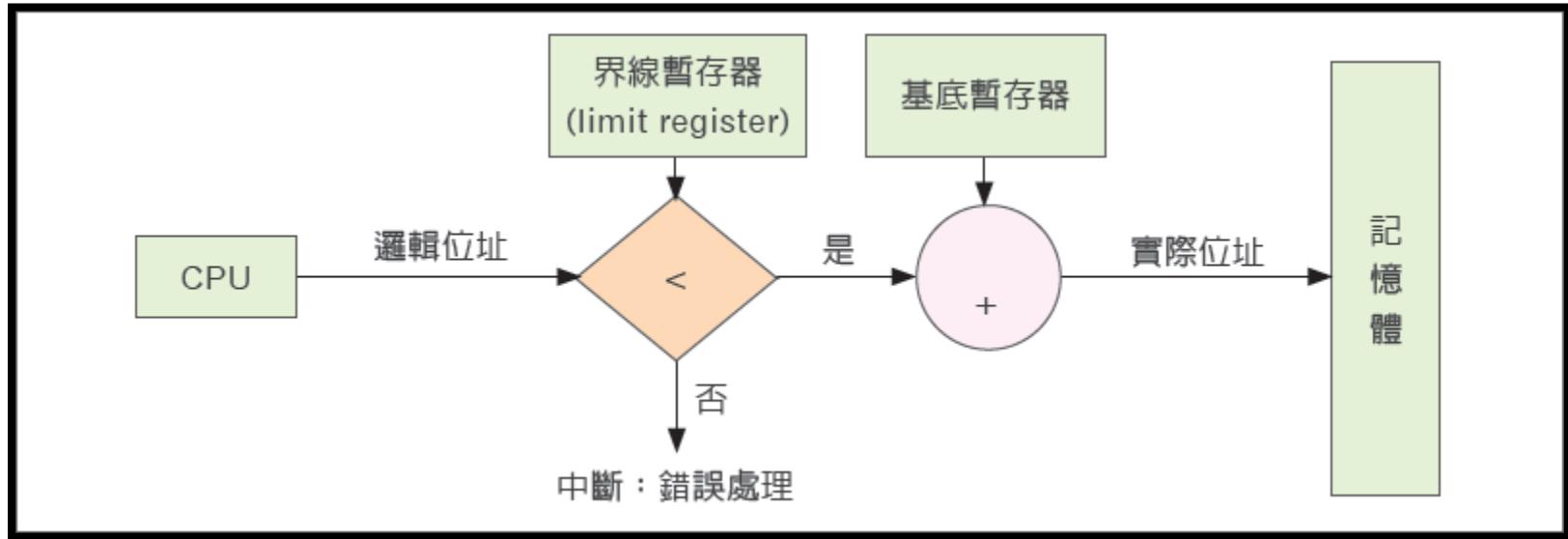
- 記憶體被分為三個連續區塊：最底層是作業系統存放的區塊，其他部分則為隨著工作負荷被動態的應用程式佔用的區塊及未使用區塊。





界線位址保護法

- ▶ 界線位址保護法是靠**界線暫存器**及**基底暫存器**來防止使用者破壞作業系統的保護機制。
- ▶ 若作業系統放置在高位址區域，界線暫存器就用來保護作業系統區塊最下界。
- ▶ 當CPU存取應用程式區塊時，會先檢查存取的邏輯位址是否小於界線暫存器。若存取位址比界線暫存器內的值還高，則發出錯誤中斷訊息，要求作業系統做出錯誤處理；若沒問題，則再加上基底暫存器的值形成實際位址。



利用界線暫存器和基底暫存器來提供記憶體保護



動態載入

- ▶ 動態載入是指**常式(routine)**只有在被呼叫的時候才會載入，也就是說，將所有的常式以可重定位載入的方式存放在磁碟空間內。
- ▶ 主程式存放在記憶體中，執行時，若需呼叫其他常式，先查看是否已經存在記憶體內。如果沒有，則呼叫重疊定位連結程式，將需要的程式或資料載入記憶體中，以便執行。



覆蓋

- ▶ 覆蓋能讓大小超過記憶體容量的程式執行。
- ▶ 是指最主要的部分會存放在記憶體中，而只有在特定時候才需要用到的指令或資料，則只有被用到時，才會放在記憶體中，其他時候就被覆蓋掉。
- ▶ 藉著舊的指令被覆蓋，才有空間存放新的指令。



覆蓋

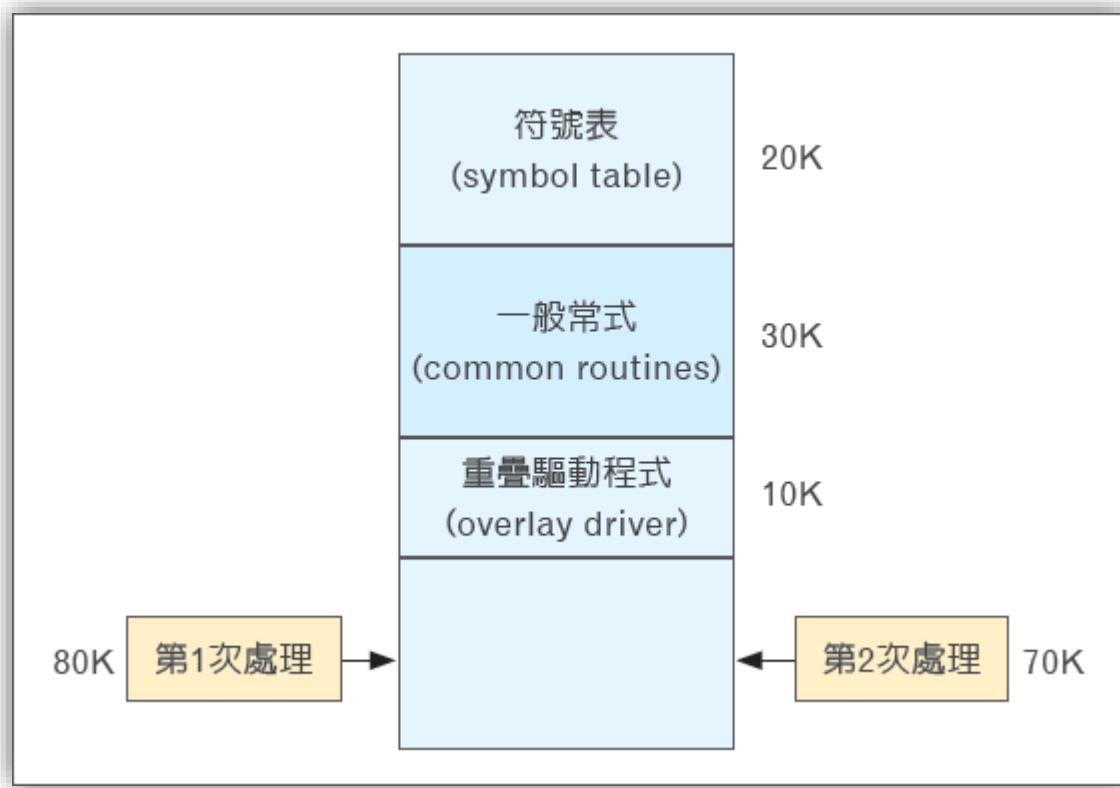
- ▶ 假設一編譯器在處理程式時：
 - ▶ 第1次處理的程式碼有80KB
 - ▶ 第2次處理的程式碼有70KB
 - ▶ 所建立的符號表有20KB
 - ▶ 會被呼叫到的常式有30KB
- ▶ 一共有200KB。假設記憶體只有150KB，可採用覆蓋技術。



覆蓋

除了符號表和常式會一直用到之外，第1次處理的程式碼跟第2次處理的程式碼分別獨立使用一次，因此：

- ▶ 第1次處理：第1次載入符號表、常式、第1次處理的程式碼、再加上覆蓋所使用到的覆蓋驅動程式(10KB)，一共是140KB。
- ▶ 第2次處理：跳到重疊驅動程式，把第2次處理程式碼覆蓋在第1次處理程式碼的區塊上，然後控制權交給第2次處理程式碼，第2次的程式大小也只有130KB。



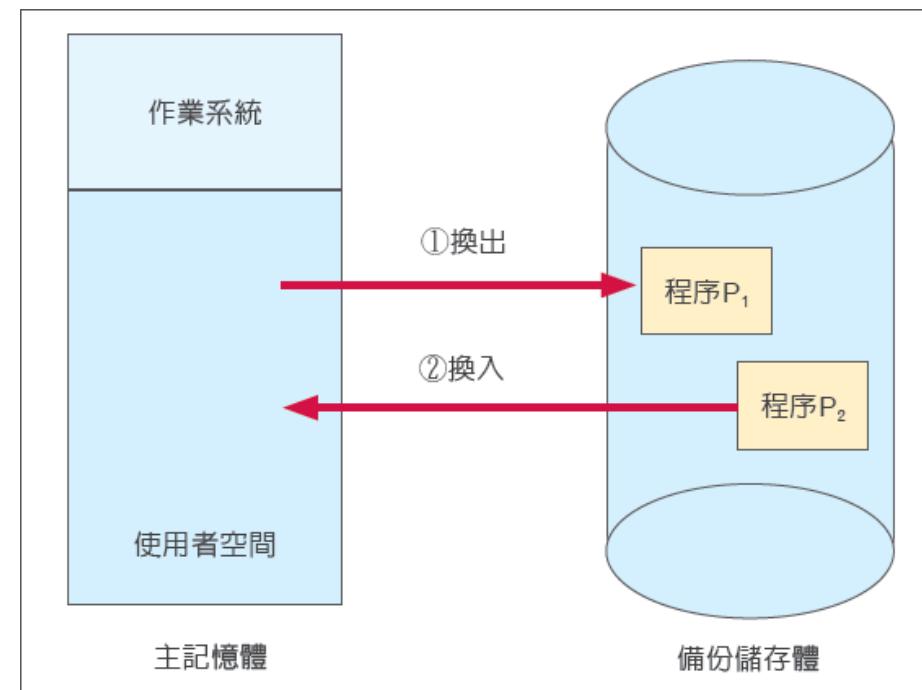
利用覆蓋的方式使大小大於記憶體的程式得以執行





置換

- 程序可能會在記憶體與磁碟之間置換(swapping)備份儲存。



利用磁碟當作備份的儲存體，
用以置換兩個程序



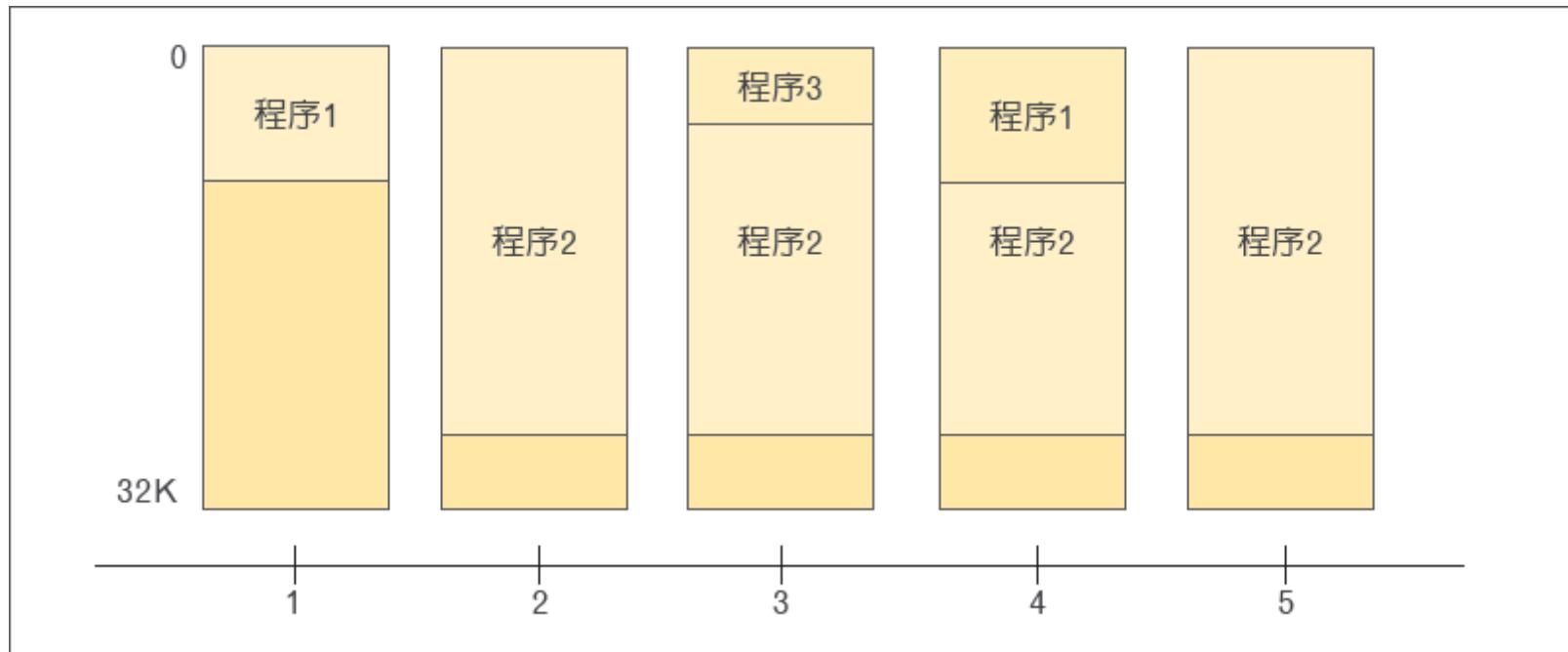
置換

- ▶ 可搭配優先順序排班來使用，當較高優先順序程序到來，低優先順序的程序就先置換到磁碟，等高優先順序結束執行，再置換低優先順序程序進來。
- ▶ 由於程序在記憶體上進進出出，因此也被稱作「**轉進轉出**(roll-in-roll-out)」方式。
- ▶ 原則上，任一時間點應該都要有一程序在記憶體中被執行。



置換

► 置換方式中使用洋蔥皮演算法：





4-5 檔案系統

- ▶ 檔案系統，是指負責存取和管理檔案資料的軟體。
- ▶ 對使用者而言，檔案系統應該要具有操作簡單、安全可靠、能夠共享或是保密等特性。
- ▶ 對作業系統而言，檔案系統要能管好檔案使用空間，使存取更有效率等。
- ▶ 檔案系統中，最重要的兩個部分就是**檔案**與**目錄結構**。



檔案屬性

名稱

符號式的檔名是給人辨識不同的檔案。

識別符號

獨一無二的標籤，用來標示檔案系統內的檔案，通常是以數字來表示，是給作業存取所用，並非給使用者辨識。

型態

提供使用者有關這個檔案的類型。



檔案屬性

位置

標示出這個檔案所在的磁碟裝置及目錄位置。

大小

顯示該檔案目前的大小。

時間日期

顯示檔案建立日期、修改日期、最後開啟日期等資訊。





4-5 檔案系統

- ▶ 所有的檔案屬性等資訊都存在目錄結構中。
- ▶ 目錄結構包含**檔名**和**識別符號**，識別符號再指到其他檔案屬性。
- ▶ 一個檔案的屬性大概需要1KB來記錄。



檔案的基本操作

- ▶ **建立檔案**：首先，在磁碟中找到空間存放這個檔案；接著，在目錄中增添一個新的檔案項目。
- ▶ **寫入檔案**：做系統呼叫，給予參數設定。在目錄中會儲存指標指到檔案結尾，寫檔時依照指標目前位置把新資料加在後頭，最後儲存最新的指標位置。
- ▶ **讀取檔案**：做系統呼叫，指定要讀的檔案及其位置。目錄裡也存放指標指向下次將被讀的位置，若有資料又被讀出，再更新指標位置。
- ▶ **刪除檔案**：搜尋目錄找到檔案所在位置，釋放檔案佔用空間，並將登記在目錄裡的檔案項目刪除。



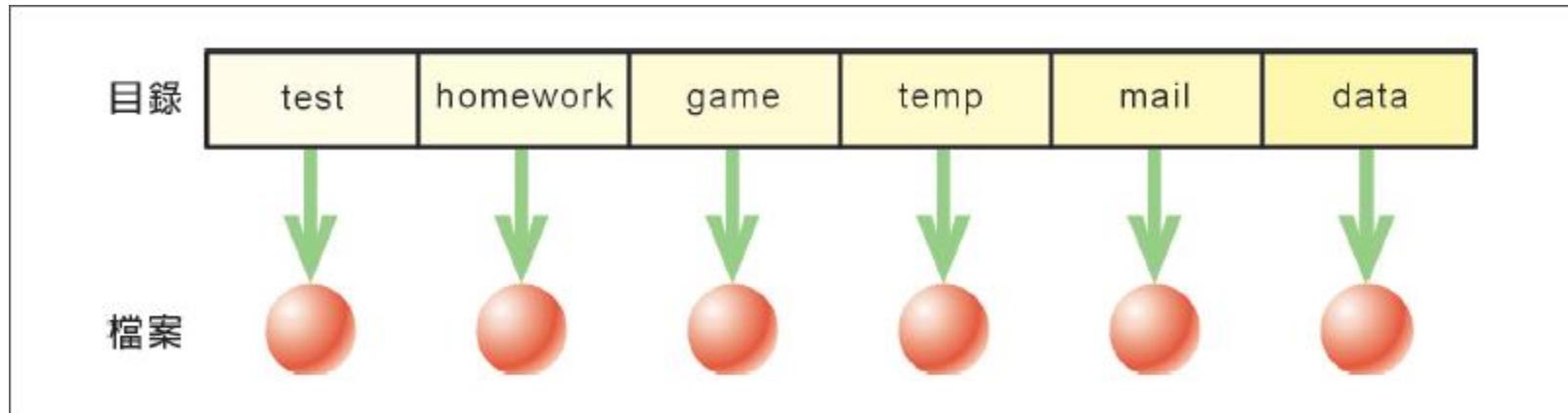
目錄結構的操作功能

- ▶ **搜尋**：由於檔案有不同的屬性，因此可根據某一屬性特色進行檔案的搜尋。
- ▶ **建檔**：當有新檔案被建立時，目錄必須要能夠為其增加一個檔案項目。
- ▶ **刪除檔案**：當檔案被刪除時，必須要把該檔案相關的資訊刪除。
- ▶ **更改檔名**：當檔案名稱被使用者改變時，目錄結構裡所記載的檔案資訊也必須一起改變。



單層目錄

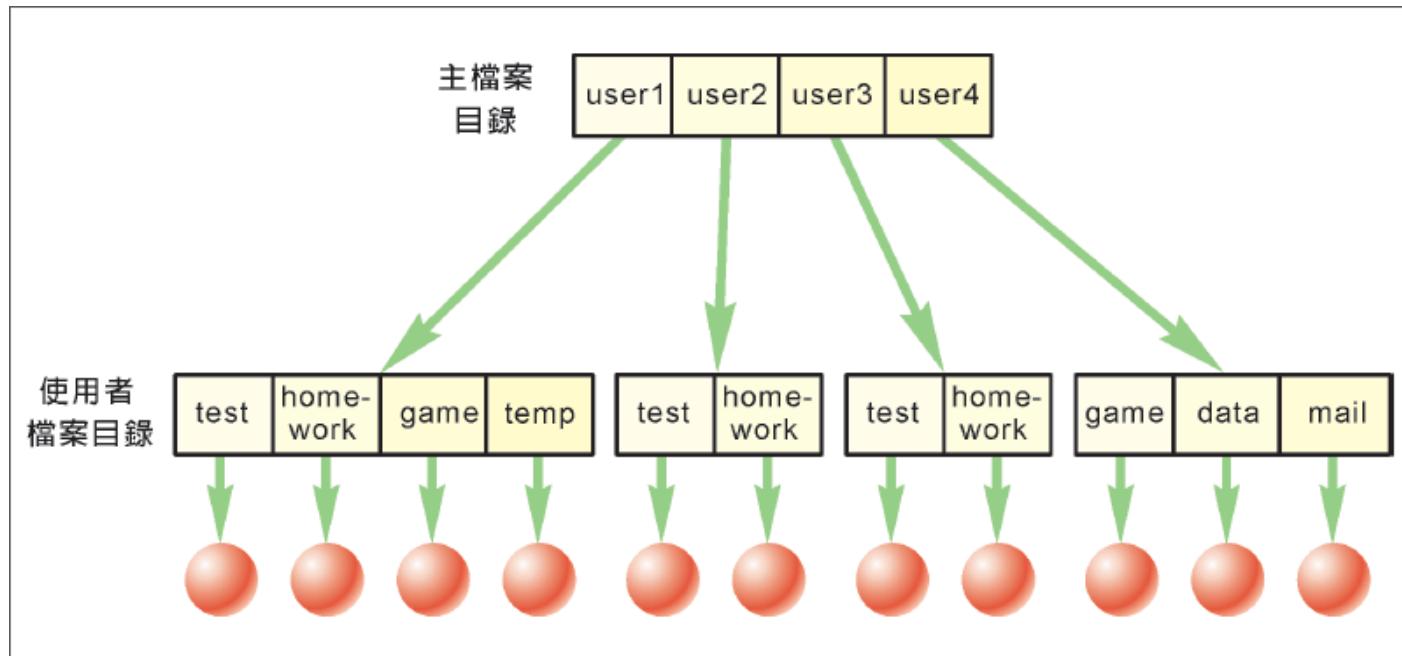
- ▶ 所有檔案都裝在同一層目錄中，是目錄結構中最簡單的架構。
- ▶ 所有使用者檔案都放在一起，當檔案數目變多，就不夠實用。





雙層目錄

- 最簡單的作法是為每一使用者建立一個目錄夾，每個使用者的目錄夾裡的結構類似。



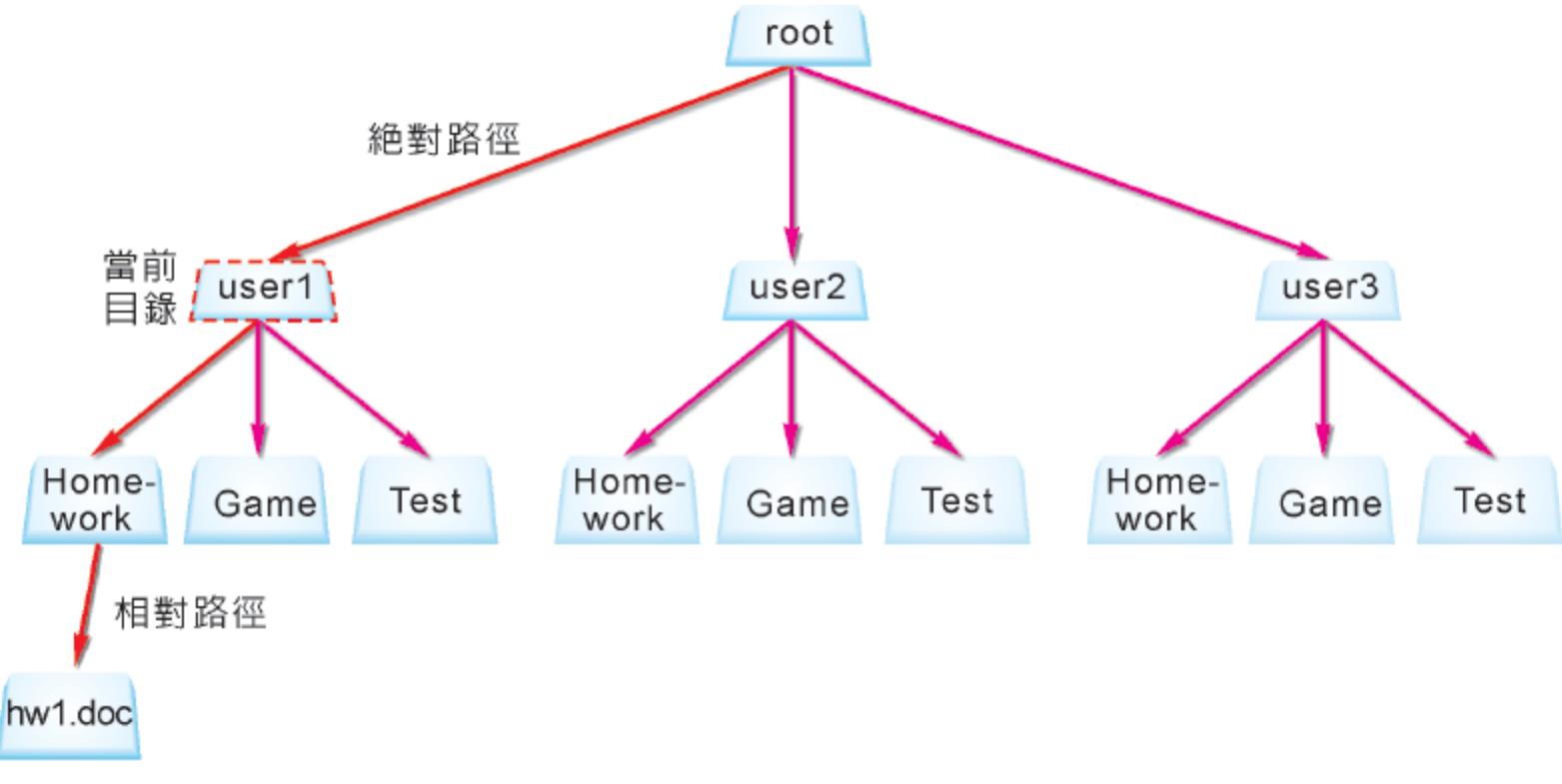


絕對路徑和相對路徑



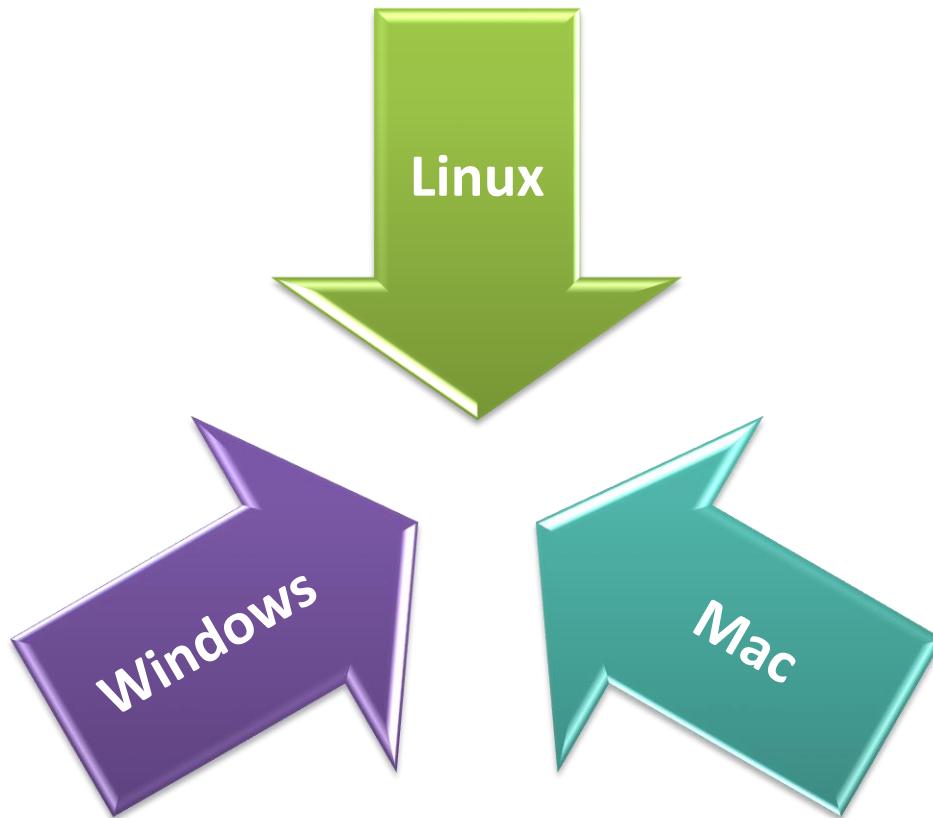


絕對路徑與相對路徑





4-6 熱門作業系統介紹





Linux

- ▶ Linux 的創始人**托瓦茲(Linus Torvalds)**有一句名言：「軟體就像性，免費的比較好。」
- ▶ 托瓦茲高喊**反版權(copyleft)**，認為軟體應該透過分享的概念，才能更快速傳遍世界，讓更多人受惠。

Linux的象徵圖案





Linux

- ▶ 1991年，還在芬蘭大學唸書的托瓦茲將發展尚未成熟的作業系統Linux 0.02版本放上學校的FTP站。
- ▶ 托瓦茲當時是想在386個人電腦上執行類似Unix的分時多工作業系統，結果受到廣大迴響。
- ▶ 所採用的策略是**史托曼(Richard Stallman)**所提倡的**通用公共授權(GPL)**，讓每個人都可在GPL授權下，免費使用或修改Linux程式碼。



Linux的特點

- ▶ 真正意義上的多工、多用戶作業系統。
- ▶ 支援數十種檔案系統格式。
- ▶ 提供先進的網路支援。
- ▶ 採用先進的記憶體管理機制，更有效地利用實體記憶體。
- ▶ 開放原始碼，用戶可以自己對系統進行改進。





Mac

- 除了外型優美之外，Mac OS使用者介面也是美得令人驚艷。





Mac

- ▶ Mac OS的使用者介面，不論是按鈕、顯示圖示，或是邊框設計，都是經過精心設計。
- ▶ 使用者介面的設計，是指研究人類的行為與操作電腦之間的關係，讓使用者能夠盡可能方便地找到如何使用某功能，或者讓使用者很快捷地選取常用的軟體。





Mac OS



Windows 1.0

- ▶ 1985年，Windows推出Windows 1.0版本。
- ▶ 在Windows之前，使用者用的是MSDOS命令列模式，只能靠著在命令列中以文字模式操作電腦。
- ▶ 有了Windows之後，使用者開始能用滑鼠來點選執行工作。此外，Windows 1.0已提供時間、月曆、小算盤等桌面應用程式。

請點

整個Windows是個龐大的家族





Windows 3.0

- ▶ 1990年推出Windows 3.0，微軟改善了效能，並讓圖形擁有16色顯示，支援386處理器。
- ▶ 提供程式管理員、檔案管理員、印表機管理員等。
- ▶ **軟體開發工具**(Software Development Kit；SDK)的發行，使軟體工程師能夠開發不同的應用軟體，Windows 3.0採納多個來自軟體獨立開發者的應用程式，而逐漸成長受歡迎。



Windows NT

- ▶ 自1993年發行Windows NT 3.1後，Windows 的發展分為DOS及Windows NT兩個支脈。
- ▶ Windows NT是微軟從80年代晚期開始專注研發的焦點，而NT兩個字代表的則是「New Technology」，企圖幫助企業使用更好的作業系統做更方便的運算。
- ▶ NT系列有桌上型版本與伺服器版本(Windows NT Advanced Server)。



Windows

- ▶ 95年推出Windows 95。
- ▶ 96年有Windows NT Workstation 4.0。
- ▶ 98年推出Windows 98。
- ▶ 2000年推出Windows ME(MillEnnium)，隨即又推出Windows 2000 Professional版本。





Windows XP

- ▶ 2001年Windows XP(代表eXPerience)問世，帶來汽球式對話方塊提醒使用者各種訊息、使用者介面學習Mac OS變得更為柔美、功能的設定更方便等。
- ▶ Windows XP有Professional及Home Edition兩個版本，其中Home Edition的市場瞄準一般家庭使用，針對影音有較佳的整合，並有各式精靈幫忙設定協定。





Windows XP Tablet PC Edition

- ▶ 2002年微軟開始大量主打以筆操控的平板電腦 (Tablet PC)，配合平板電腦推出從Windows XP Professional改版而來的Windows XP Tablet PC Edition。
- ▶ 作業系統的改變，使人們可將電腦當作電子書一般輕巧攜帶，輸入資料時也可擺脫鍵盤與滑鼠，直接用筆輸入。





Windows Vista

- ▶ 2007年推出Windows Vista(於發展階段時被稱為Longhorn)作業系統。
- ▶ 標榜讓使用者「更方便、更安全、更多樂趣」。讓使用者以更方便輕鬆的方式，尋找、使用、管理、共用並享用資訊。
- ▶ Windows Aero介面讓使用者可清楚看到正從事的工作，而 Windows Flip 3D 能協助使用者迅速在視窗與工作間切換。



Windows Flip 3D 能協助使用者迅速地在視窗與工作間切換





Windows Vista

- ▶ Vista的「立即搜尋」可直接輸入關鍵字來尋找任何檔案、文件、相片、電子郵件訊息、歌曲、影片或程式。
- ▶ Vista將維護電腦健全狀況及安全性的流程自動化，若遇重大問題，使用者可使用「完整電腦備份及還原功能」，輕鬆復原整台電腦。





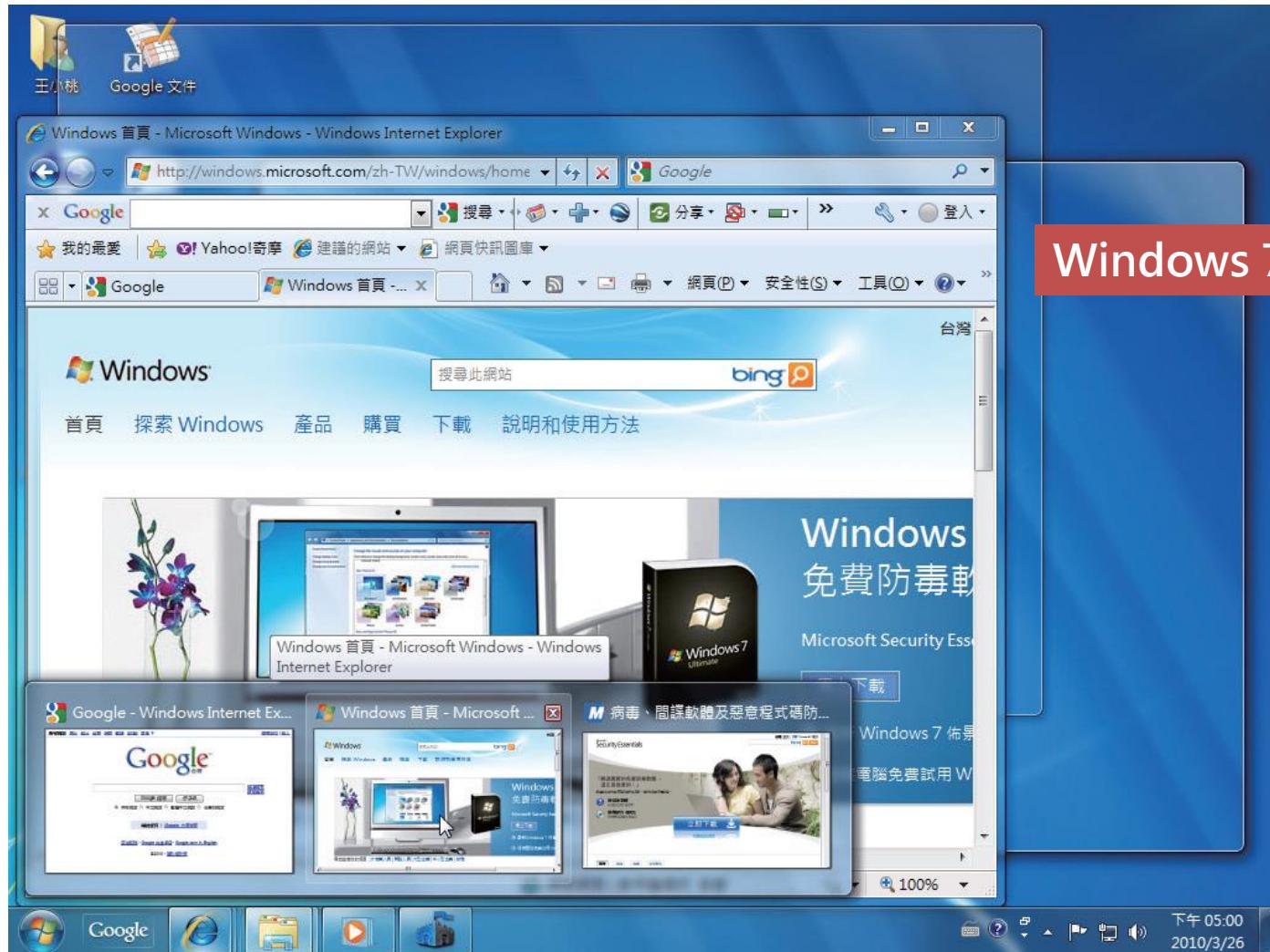
Windows Vista

- ▶ IE7中的「動態安全性保護功能」可協助防護電腦及隱私權，不受惡意軟體、詐騙網站及線上網路釣魚的侵害。
- ▶ Windows Defender技術能有效阻擋由間諜軟體及其他侵入性程式所帶來的快顯視窗、效能低落及安全性威脅等侵擾。
- ▶ Windows Media Center讓使用者以劇院投影片放映方式檢視相片、依照封面瀏覽音樂收藏、輕鬆播放DVD、觀賞並錄製電視節目、下載電影。



Windows 7

- ▶ 2009年10月推出Windows 7，供個人電腦、筆記型電腦、平板電腦及多媒體中心等使用。
- ▶ 速度快、硬體需求低、使用者介面操作簡單、與各類軟硬體相容性佳、介面漂亮等特點，是Windows 7最常被讚許的特點。
- ▶ Windows 7支援螢幕**多重觸控**(multitouch)和手寫辨識。
- ▶ Windows 7企圖切入Netbook這類低價裝置。





Windows 8

- ▶ 2012年10月推出Windows 8作業系統，特別強化螢幕的觸控功能。
- ▶ 取消傳統的「開始」功能表按鈕，推出**Metro(動態磚)**操作界面，無法讓使用者接受。
- ▶ 隔年推出Windows 8.1系統，便把「開始」功能表按鈕加回去。

Windows 8的使用者介面





Windows 10

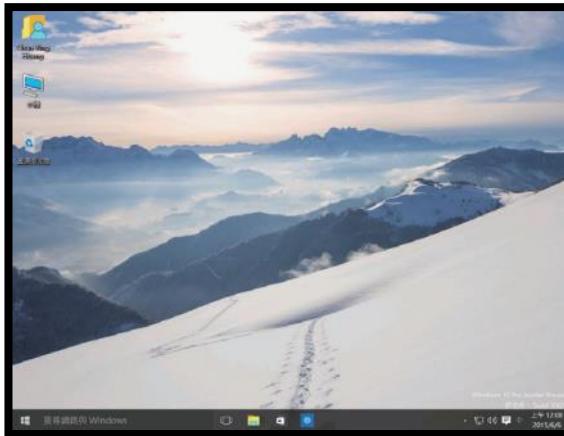
- ▶ 2015年7月推出Windows 10作業系統。
- ▶ 微軟提供免費升級Windows 10作業系統。
- ▶ Windows 10根據使用者意見改良設計，包括結合傳統和動態磚設計的新式開始功能表、更簡潔的界面設計、新的軟體市集、語音助手、通知中心、虛擬桌面，內建的地圖應用程式、還有取代IE的全新瀏覽器Microsoft Edge等。



Windows 10作業系統的操作界面



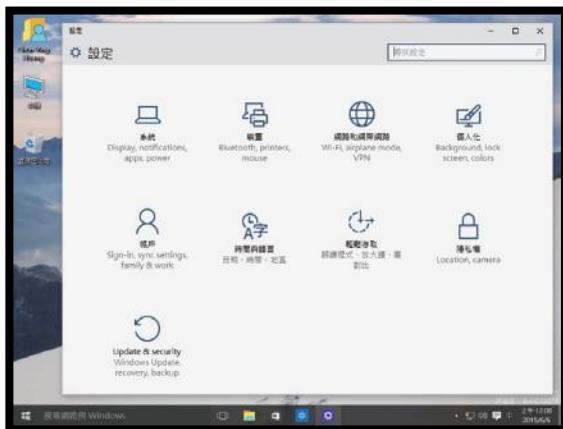
(a) 登入畫面



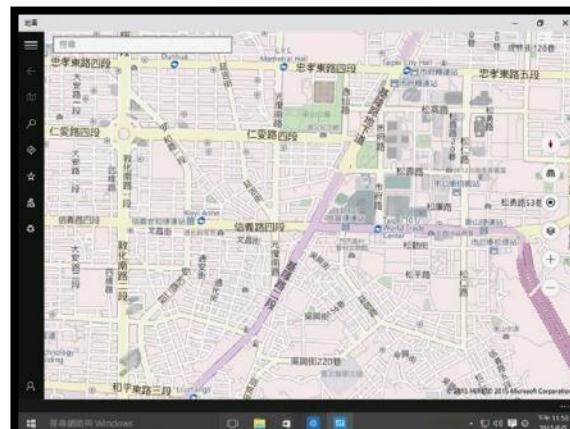
(b) 桌面



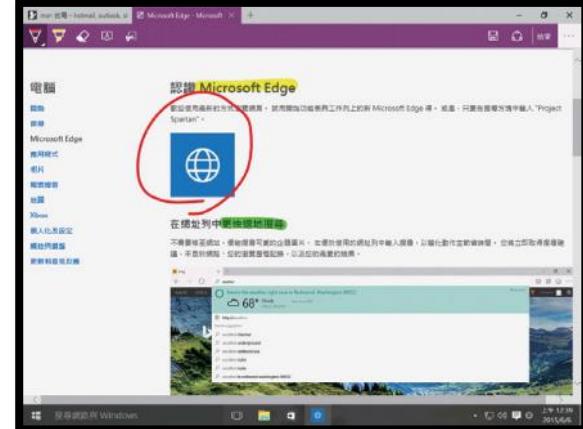
(c) 開始功能表



(d) 設定畫面



(e) 地圖應用程式



(f) Microsoft Edge瀏覽器

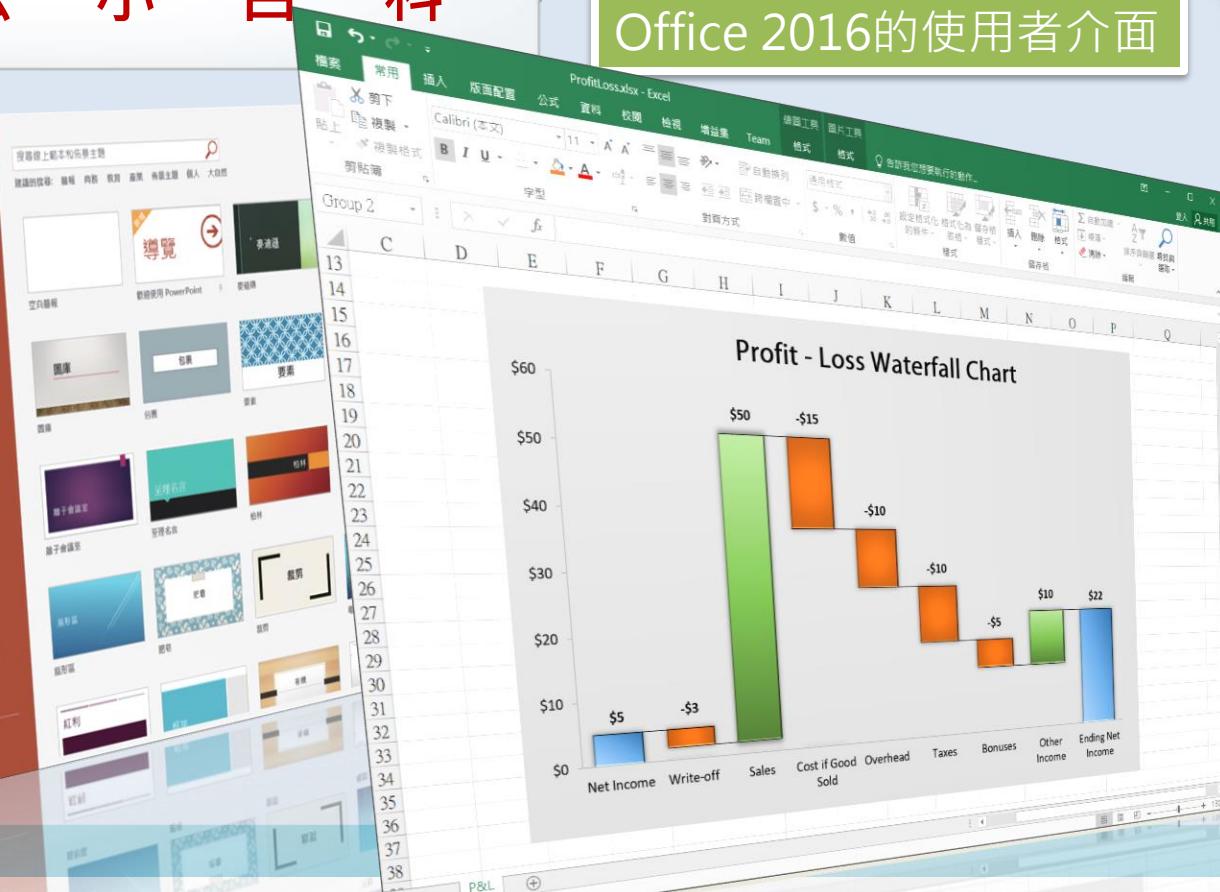
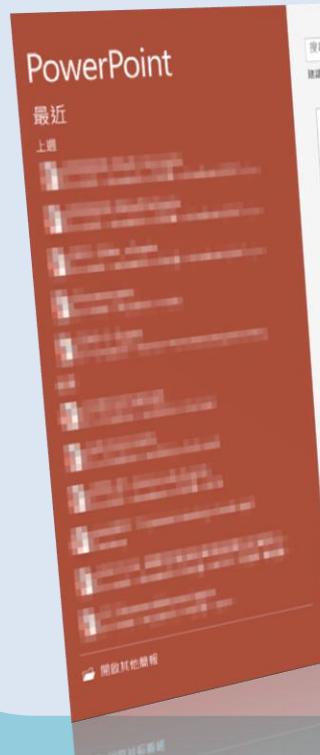




Office軟體自版本2007以來，最顯著的改變是使用者介面的設計，然而更值得一提的是，在預設情況下儲存Word、Excel、PowerPoint檔案時，會使用新的XML檔案格式，稱為 **Microsoft Office Open XML Formats**。Office XML因採用ZIP壓縮技術，因此能縮小各種Office的檔案，並且也因為重新調整過檔案架構，因此檔案存取、檔案內容的管理都更有效率。此外，因為Office XML格式是開放標準，因此其他軟體開發商或開發者，都能開發出存取Office XML檔案格式的程式。Office 2010是第一個有32位元及64位元版本的Office。



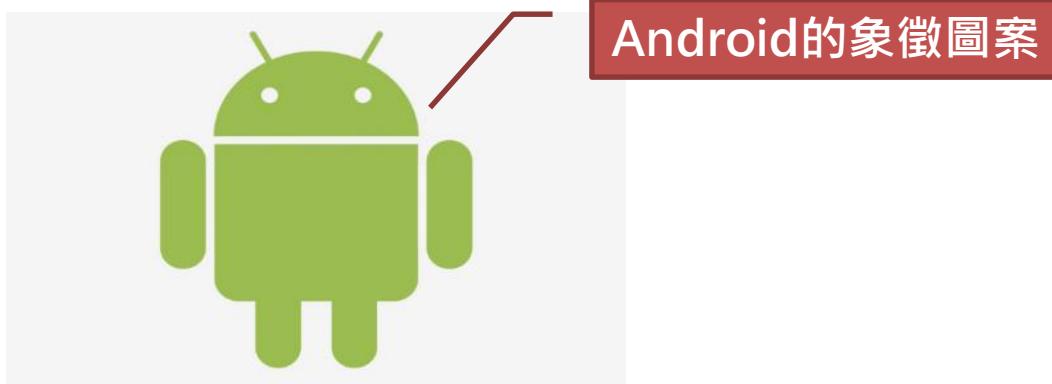
Office 2016的使用者界面





4-7 行動裝置作業系統

- ▶ 行動裝置從早期的傳統手機演進到現在的智慧型手機和平板，其功能也愈來愈強大複雜。
- ▶ 目前市佔率最高的作業系統，就是Google的Android和Apple的iOS作業系統。





Android

- ▶ 隨著科技的演進，手持式裝置的硬體規格和功能愈來愈強大，為了讓手持式裝置得以運作順暢，必須有足以支持複雜系統運作的作業系統。
- ▶ Android就是一個專門為嵌入式和手持式裝置開放的作業系統。
- ▶ Android一開始是由一家叫做Android的公司自2003年起開始進行研發的作業系統。原本是以像數位相機這樣的小型嵌入式系統為目標。



Android

2005年

- Google買下Android，且仍以Android為名稱，繼續進行相關研發。

2007年11月5日

- Google正式宣佈將Android作業系統用於手機，同時宣稱Android將會是一個免費且開放原始碼的作業系統。

2008年9月

- 第一個搭配Android 1.0的手機，是以美國電信商T-Mobile之名推出型號為G1的手機。



Android

- ▶ 自Android 1.5版本起，所有主要發行版本都有對應的甜點別名！
- ▶ 隨著Android新版本的推出，其開發工具的API等級也一直提升。Android軟體開發人員需小心挑選開發工具的API等級，以支援最多的使用者。



Android版本列表

版本	發行時間	別名	API等級
Android 1.5	2009年4月	紙杯蛋糕(cupcake)	3
Android 1.6	2009年9月	甜甜圈(donut)	4
Android 2.0	2009年10月	閃電泡芙(eclair)	5
Android 2.2	2010年6月	霜凍優格(froyo)	8
Android 2.3	2010年12月	薑餅(gingerbread)	9
Android 3.0	2011年2月	蜂巢(honeycomb)	11
Android 4.0	2011年5月	冰淇淋三明治(ice cream sandwich)	14
Android 4.1	2012年7月	雷根糖(jelly bean)	16
Android 4.4	2013年10月	奇巧(kit kat)	19
Android 5.0	2014年11月	棒棒糖(lollipop)	21
Android 6.0	2015年10月	棉花糖(Marshmallow)	23
Android 7.0	2016年8月	牛軋糖(Nougat)	24
Android 8.0	2017年8月	奧利奧(Oreo)	26



Android

- ▶ Android是一個基於Linux核心開發的作業系統，但它和一般的GNU/Linux作業系統的差異很大。
- ▶ Android系統使用的視窗圖形界面系統及系統函式庫，與大部份POSIX函式相容，但並不是完全相同。因此某些GNU/Linux環境下的工具不易被移植到Android系統裡。
- ▶ Android應用程式大都使用Java編寫，執行時是透過**Dalvik**的類Java虛擬機器來執行。



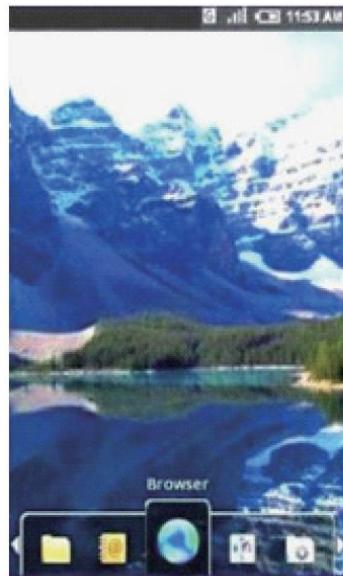
Android

- ▶ Android有一套自己的系統資源權限管理機制，所有程式都是在獨立的**沙盒(sandbox)**內執行。
- ▶ 自Android 4.4版起，Google提出Android ART(Android Runtime)，嘗試取代舊有效率較差的Dalvik執行環境。
- ▶ 自Android 5.0，ART則成為內建且唯一的執行環境。



Android操作界面畫面

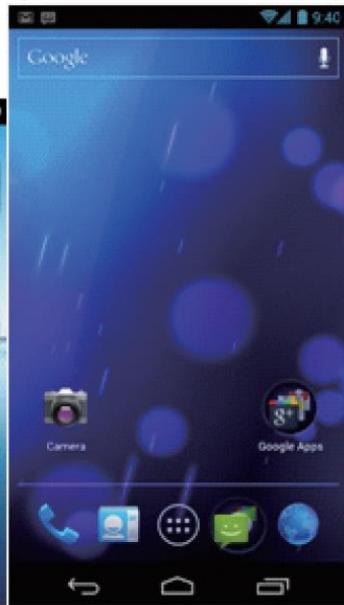
Android 1.1



Android 1.1



Android 4.0



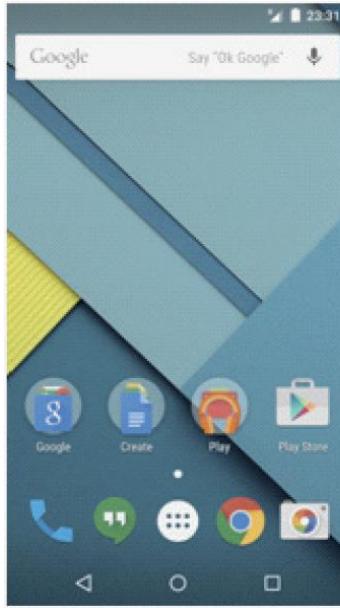
Android 4.4





Android操作界面畫面

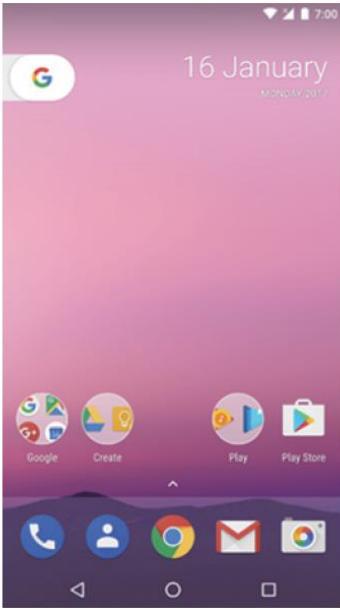
Android 5.0



Android 6.0



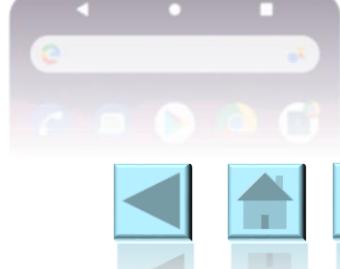
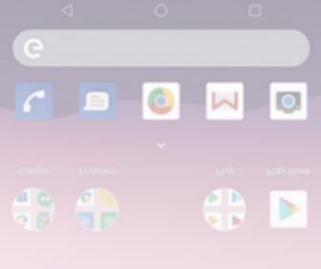
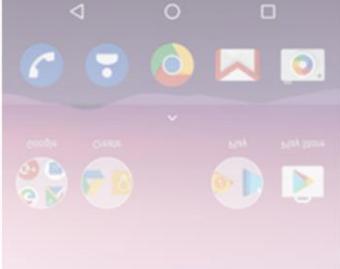
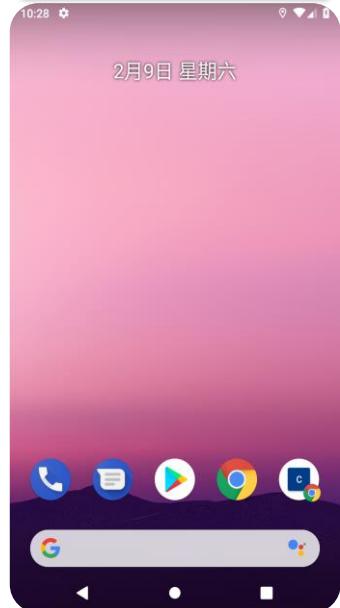
Android 7.0



Android 8.1



Android 9.0





iOS

- ▶ 蘋果電腦作業系統是Mac OS X，iPhone手機的作業系統則是iOS。雖然兩者外觀和界面有所不同，但作業系統核心是同一套：都是基於Darwin核心所建構起來的作業系統。
- ▶ iOS最早版本是在2007年發行的iOS 1.0版，最早用在第一代的iPhone手機和iPod Touch裝置上。目前已到11.x系列版本。
- ▶ iOS作業系統只能在蘋果智慧型手機上才能使用。



iOS作業系統主頁畫面截圖





Android v.s. iOS



- ▶ iOS系統是比較封閉的系統。
- ▶ 二者官方網站均提供「軟體市集」服務供使用者下載和安裝應用程式。但iOS是不允許使用者從第三方網站安裝未經蘋果公司檢驗的軟體。
- ▶ 蘋果公司對於軟體市集上架的軟體檢驗時間較長、標準也較嚴格。
- ▶ 雖然系統較封閉，但iOS環境可能因此較安全。



比較熱門的行動裝置作業系統：Google Android VS Apple iOS



	Android	iOS
系統本質	開放、多元、有彈性	簡潔、易用、穩定
系統核心	Linux	OS X(Darwin)
軟體開發語言	C/C++/Java	C/C++/Objective-C/Swift
系統客製化	容易	非常有限，除非越獄
可使用記憶卡	依裝置廠牌型號而有所不同	無法使用



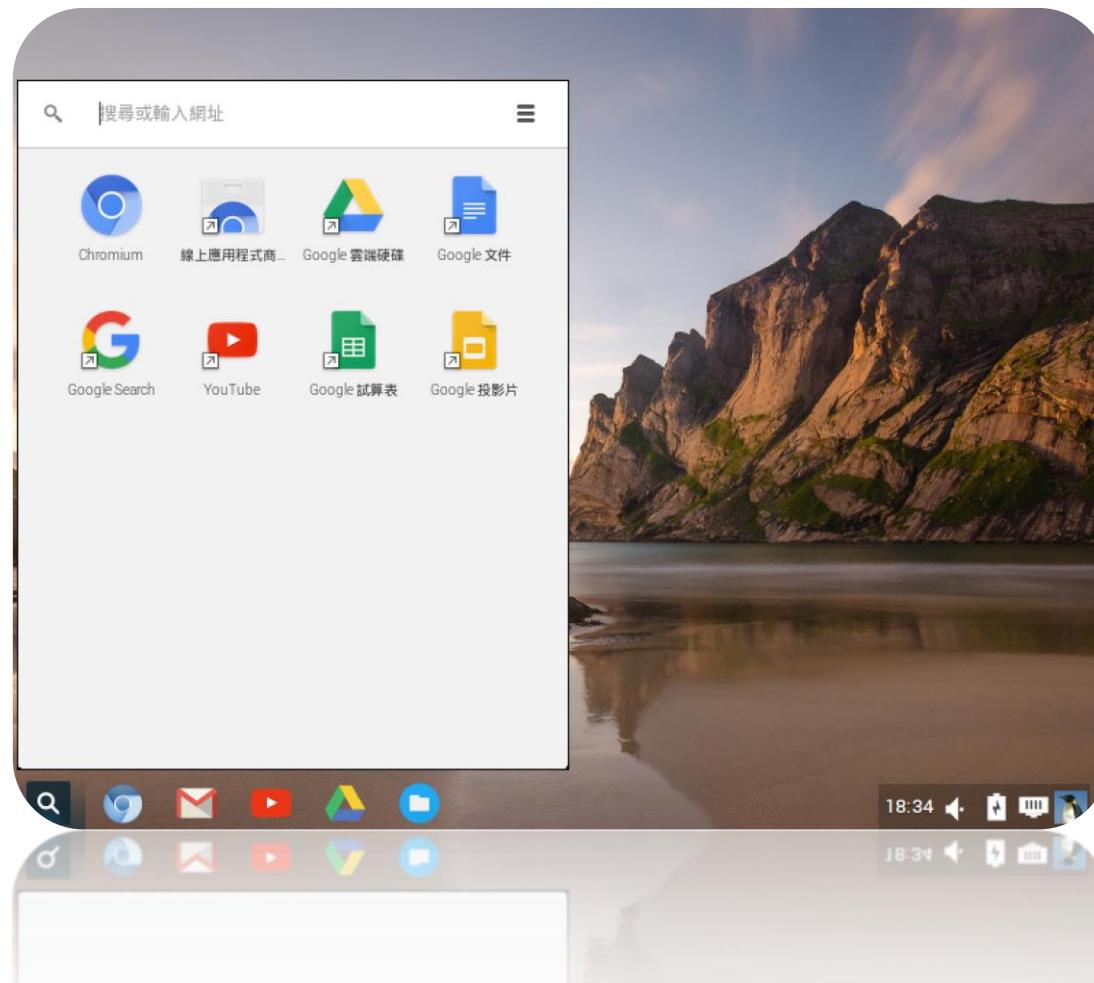


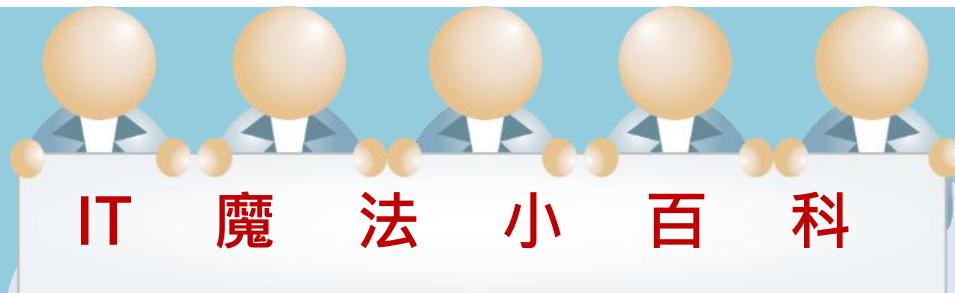
Chrome OS

- ▶ Chrome OS 是由Google針對上網裝置推出的輕型作業系統。
- ▶ Chrome OS的理念就是：使用者不需要安裝任何應用程式，只要有網頁瀏覽器，所有的應用程式都可以在雲端裡執行。也因此，Chrome OS其實沒辦法安裝一般的應用程式。
- ▶ 而雲端上的應用程式（如Google的各式服務）在 Chrome OS裡執行時，也包裝得好像就是本機的應用程式一樣。此外，使用者必需要有Google帳號，才得以登入和使用Chrome OS 。



Chrome OS 的桌面截圖





「越獄」(jailbreak)這一詞在iOS作業系統上很常見。由於iOS的限制較多，駭客便嘗試破解作業系統，以取得更多系統權限，安裝客製化的軟體。

而在Android系統上，雖然客製化已經非常容易了，但不滿足的使用者也會嘗試取得所謂的「root」系統管理員權限，以更進一步地控制自己的行動裝置。

然而，讀者若對越獄或是取得「root」權限有興趣，可能得非常小心才行！首先，越獄程式通常來自不明的出處，越獄後安裝的第三方軟體可能藏有後門；另外，如果不小心沒有操作好的話，可能還會發生無法回覆的破壞或是影響保固呢！