

04 作業系統

CHAPTER

- 4-1 作業系統簡介
- 4-2 CPU排班
- 4-3 記憶體管理
- 4-4 檔案系統
- 4-5 熱門作業系統介紹

4-1 作業系統簡介

- 電腦系統：硬體、作業系統、應用軟體、使用者
- 作業系統：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



作業系統的主要工作

中央處理器管理

把處理器有效地安排給各個程序使用

記憶體管理

妥善分配記憶體給各個程序使用

檔案管理

讓使用者安全存取及控制檔案

周邊設備管理

管理各項周邊系統，提供簡易使用者介面程式

程序管理

依據程序控制表安排資源

作業系統的演進：主機型系統

- 手動操作階段
 - 使用者必須熟悉硬體操作，運作麻煩並且容易出錯
- 批次系統
 - 把相同或相似的工作集合成為一整批(batched)
 - 各個作業之間的轉接由程式自動操作，縮短手動操作緩慢所造成的CPU等待時間
 - 簡單的批次系統記憶體配置

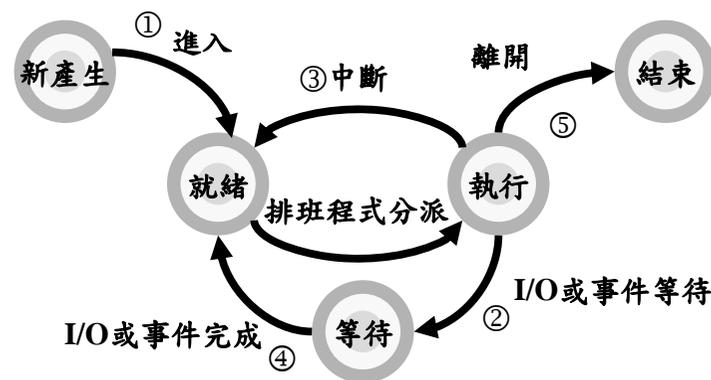


作業系統的演進：主機型系統(cont)

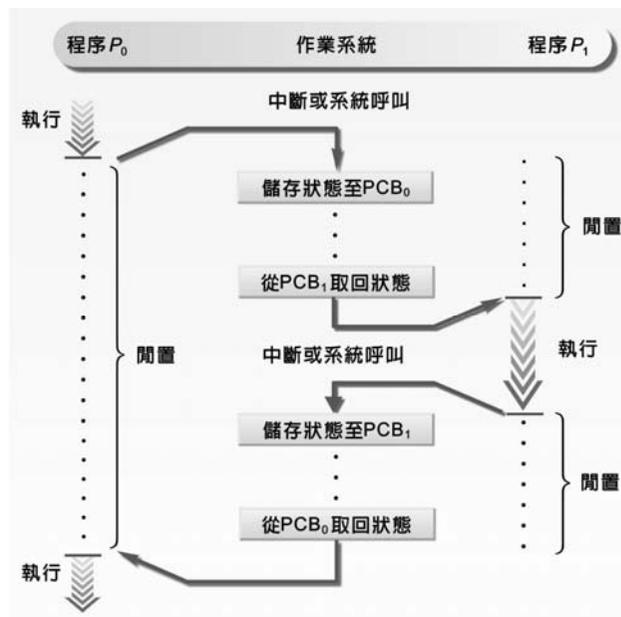
- 多元程式規劃系統
 - 利用多元程式規劃增加CPU使用率
 - 程序的狀態：

新產生	程序正在產生中
執行	程序得到資源正在執行
等待	程序等待某個事件發生
就緒	程序一切已準備就緒
結束	程序已完成

程序狀態關係圖

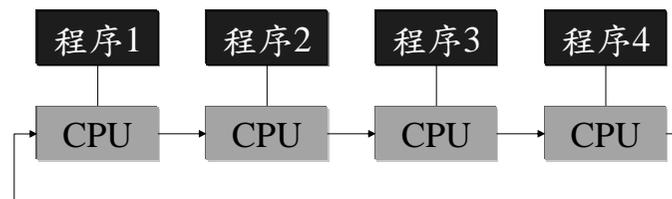


處理器排班



作業系統的演進：主機型系統(cont)

- 分時系統
 - 採用時間觸發，CPU輪流計算各個程序，時間一到就把CPU交給下一個程序使用
 - 分時系統的特點：
 - 同時性：可同時有若干個使用者連結到同一計算機進行運算
 - 獨立性：不同使用者之間不會相互干擾
 - 即時性：每一個使用者都可以即時得到計算機的回應



作業系統的演進：個人電腦系統

- 個人電腦設計方向：增進使用者操作方便，並且提升CPU的回應速度避免使用者等待
- 個人電腦系統的演進：

早期：DOS文字指令

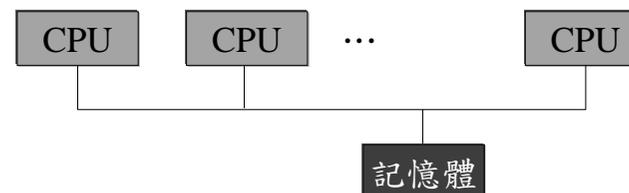
第一個圖形化介面：Mac OS

最多人使用：Windows

免付費作業系統：Linux

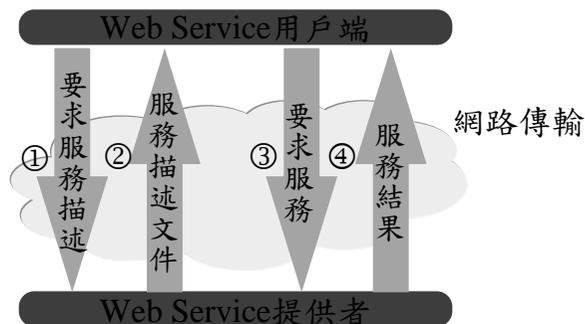
作業系統的演進：多處理器系統

- 一個以上的處理器，彼此之間緊密地溝通合作，共享資源，共用時脈。
- 多處理器系統可以增進效能，也因共用周邊設備，共享電腦資源，所以較省空間。
- 多處理器系統中，多個應用程式共用記憶體等資源。
- 多處理器系統擁有較好的容錯能力。



作業系統的演進：分散式系統

- 每一台電腦是各自獨立的，並不共享資源或時脈，彼此之間依靠網路傳輸交換資料
- 分散式系統的應用：Web Service



作業系統的演進：即時系統

- 即時系統：
 - 計算機要能即時回應外部事件的要求，於規定的時間內完成對該事件的處理，並且還能控制所有的即時設備和工作能夠協調一致地執行
- 即時系統的重點：

即時時鐘管理

定時工作或延長工作以利與其他工作協調。

過載保護

發生過載時，使用緩衝區來應急。

高度可靠性

計算機發生問題時，系統有容錯能力能繼續運作。

作業系統的演進：手持系統

- 手持系統：個人數位助理，較手提式電腦輕薄短小。
- 手持式系統的特點：



記憶體容量小

必須有較好的記憶體管理方式。

處理器運算緩慢

為使電池使用時間較長，故運算速度不可能太快，因此必須巧妙設計作業系統或應用程式。

顯示螢幕小

使用者介面設計必須格外留意好讓使用者看到較多畫面。



4-2 CPU排班

- CPU排班：保持隨時都有一個程序在執行，以提高CPU的使用率
- CPU排班的五個決策時間點：

程序新產生時

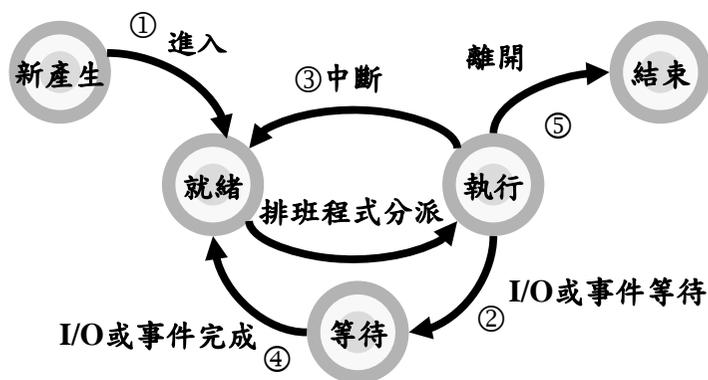
程序從執行狀態變等待狀態(譬如有I/O要求)

程序從執行狀態變就緒狀態(譬如有中斷發生時)

程序從等待狀態變就緒狀態(譬如I/O要求得到回應)

程序終止結束

五個必須CPU排班的時間點



評量CPU排班演算法好壞的標準

CPU使用率 | 使用率越高表示演算法越好。

產能 | 計算單位時間內完成的程序數目，但若是執行大程序則可能導致產能比執行很多小程序小。

回復時間 | 程序經歷多少時間才完成其計算。

等待時間 | 一個程序花多少時間在就緒佇列中等待。

最短工作先處理

- 舉例：

程序	所需時間(毫秒)
P ₁	7
P ₂	3
P ₃	5

- 甘特圖如下所示：



最短工作先處理 (cont)

- 各程序的等待時間

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 (cont)

- 不同的抵達順序及其平均等待時間

程序先後順序	平均等待時間
P ₁ → P ₂ → P ₃	5.67
P ₁ → P ₃ → P ₂	6.44
P ₂ → P ₁ → P ₃	4.33
P ₂ → P ₃ → P ₁	3.67
P ₃ → P ₁ → P ₂	5.67
P ₃ → P ₂ → P ₁	4.33

優先權排班

- 優先權如下：

程序	優先權	所需時間(毫秒)
P ₁	3	7
P ₂	1	5
P ₃	2	4

- 甘特圖如下：



依序循環排班

- 依序循環排班方式在使用時，先預設好經過多少時間CPU就該切換執行下一個程序，也就是設定好間隔時間(time slice)。所有的程序放在先進先出的佇列裡面，

首先 CPU排班從佇列裡挑第一個程序執行。

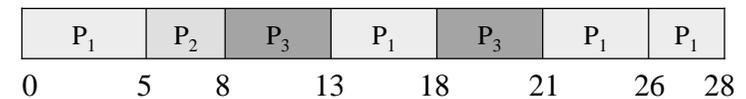
然後 開始進行倒數，時間到的時候就得讓CPU處理佇列裡下一個程序。

依序循環排班 (cont)

- 範例：

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

- 甘特圖：



4-3 記憶體管理

- 記憶體管理

記憶體管理

把記憶體分割成各個區塊，以供各程序或各使用者使用。

記憶體位址定位

把程序所使用的邏輯位址與記憶體的實際位址作映射。

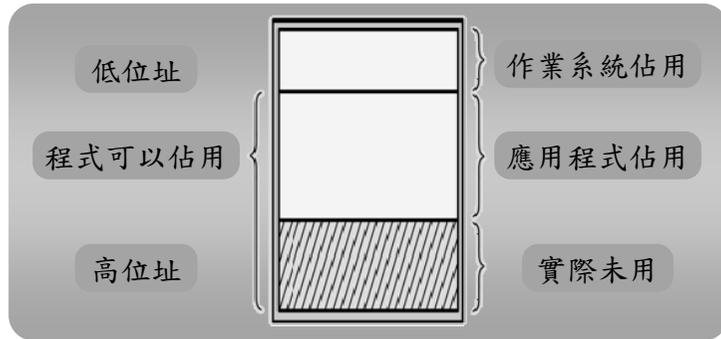


記憶體保護與共享

程序之間所使用的記憶體不能相互干擾，可是作業系統的部分要讓各個程序共享。

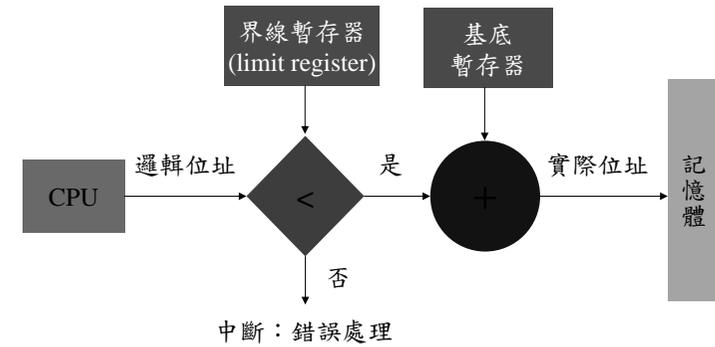
單一連續記憶體配置方式

- 記憶體被分成三個區塊：作業系統存放、應用程式佔用、未使用區塊



單一連續記憶體配置(cont)

- 利用界線暫存器和基底暫存器來提供記憶體保護

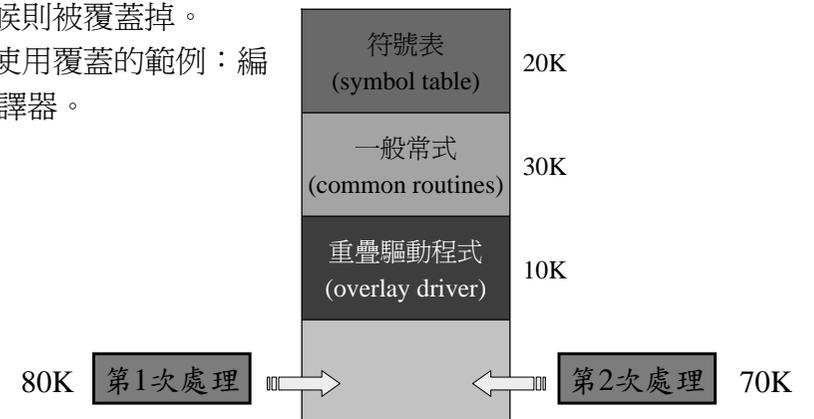


動態載入

- 動態載入是指常式(routine)只有在被呼叫的時候才會被載入，平常存放在磁碟空間內，執行時若發現所需的常式不在記憶體內則動態載入。
- 動態載入可以提供較大的彈性。

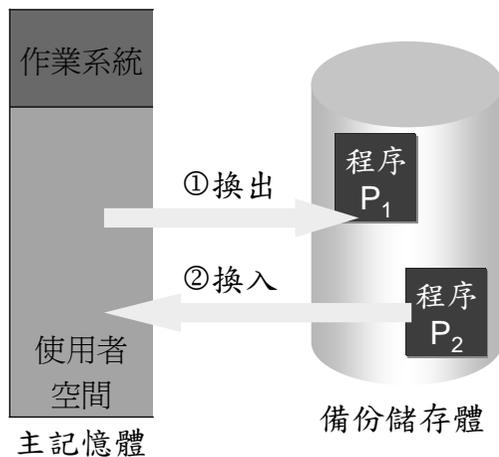
覆蓋

- 主要的部分會一直存放在記憶體中，在特定時候才需要用到的指令或資料，則只有用到時才放進記憶體中，其他時候則被覆蓋掉。
- 使用覆蓋的範例：編譯器。



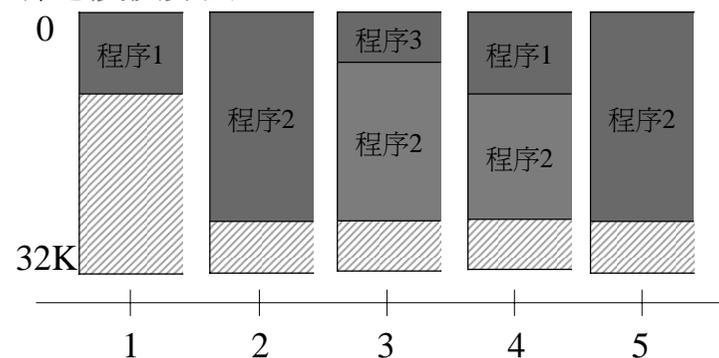
置換

- 利用磁碟當作備分的儲存體，用以置換兩個程序



置換 (cont)

- 洋蔥皮演算法



附註：Paging system是虛擬記憶體管理的重要機制，它將一個程序的記憶體需求分割成大小相同的pages，以便更有效管理。

4-4 檔案系統

- 檔案系統負責存取和管理檔案資料
- 檔案系統的重要屬性：

名稱 讓使用者辨別不同的檔案。

識別符號 獨一無二的標籤，讓作業系統辨別檔案。

型態 顯示檔案的類型。

位置 標示出檔案所在的磁碟及目錄位置。

大小 顯示檔案目前的大小。

時間日期 顯示檔案建立日期、修改日期、最後開啟日期等。

檔案的基本操作

- 檔案的基本操作

建立檔案

讀取檔案

寫入檔案

刪除檔案

- 目錄結構必須支援的功能

搜尋

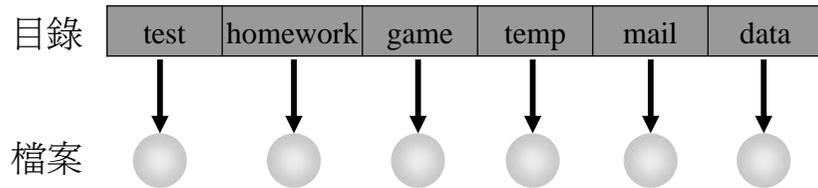
刪除檔名

建檔

更改檔名

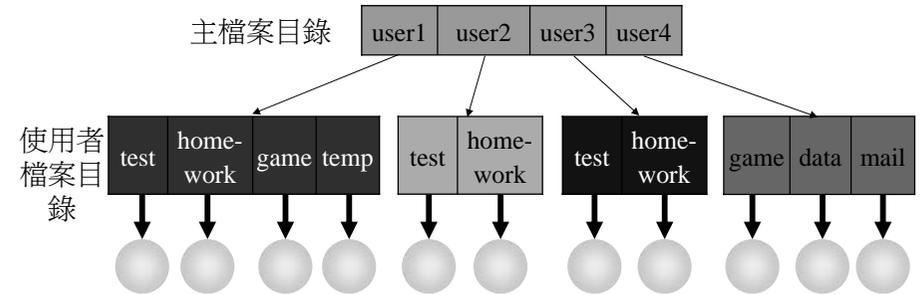
目錄結構：單層目錄

- 檔案皆在同一層目錄
- 同一個目錄下不能有兩個同樣檔名的檔案



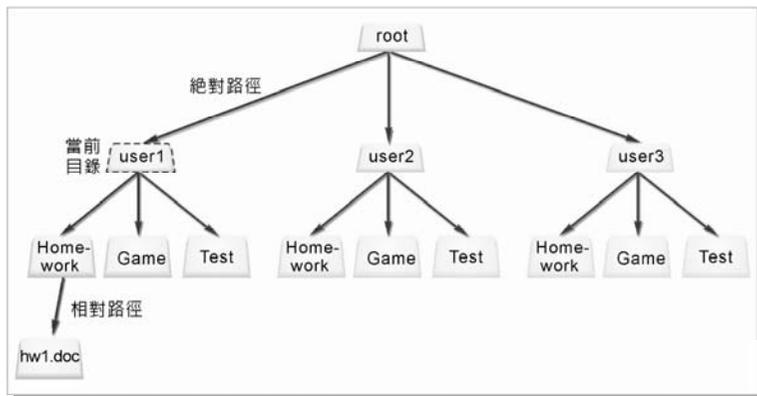
目錄結構：雙層目錄

- 每個使用者的目錄結構相似
- 開啟檔案時，只會搜尋使用者自己的目錄



路徑

- 絕對路徑：root/user1/homework/hw1.doc
- 相對路徑：/homework/hw1.doc



4-5 熱門作業系統介紹

- Linux
 - 創始人托瓦茲認為軟體免費的比較好
 - 開放原始碼，透過大家發現漏洞，一起改善系統
- Linux的特色：
 - 多工、多用戶作業系統
 - 支援數十種檔案系統格式
 - 提供了先進的網路支援
 - 採用先進的記憶體管理機制，更加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



熱門作業系統介紹 - Mac

第一個圖形化的作業系統

優異的使用者介面

漂亮的外型設計

針對影像圖片等工具有較好的支援程式



熱門作業系統介紹 – Windows

最多人使用的作業系統

1990年推出Windows 3.0

分成DOS支脈及Windows NT

Windows NT代表New Technology

Windows XP代表eXPerience

2007年推出Vista

2009年推出Windows 7

