

(pointers to scheduling queues) 、排程相關參數 (scheduling parameters) 。

6. 記憶體管理資訊 (memory management information)

包括如基底暫存器 (base register) 、限制暫存器 (limit register) 、分頁表 (page tables) 、分段表 (segment tables) 等的資訊。

7. 會計資訊 (accounting information)

會計資訊紀錄的資訊如下：

- (1) 行程用了多少CPU time。
- (2) 行程能使用CPU時間的最大值 (max time) 。
- (3) 行程的時間配額 (time quantum) 。
- (4) 目前作業系統中有多少行程數目 (job or process numbers) 等等。

8. I/O裝置狀態資訊 (I/O status information)

I/O裝置狀態資訊包括一些行程未完成的I/O request，例如在I/O queue中的等待編號，還有行程已開啓那些檔案的紀錄 (a list of open files) ，都是屬於I/O裝置狀態資訊的一部分。

執行緒 (Thread)

執行緒有時稱為輕量級行程 (lightweight process) ，它也是CPU使用時的一個基本單元 (就像行程一樣) ，並且是由程式計數器、一組暫存器以及一個堆疊空間所組成。一個行程裏面可以有一群執行緒在上面執行。這些執行緒共用程式碼 (code) 、記憶體位址空間 (address space) 和作業系統資源，且有各自獨立的暫存器 (registers) 與堆疊 (stacks) ，執行緒最大的優點為在作內容切換 (context switch) 時比行程更有效率。

多執行緒 (Multiple-threaded) 96 雄大資管

多個執行緒之間共用記憶體和作業系統的資源，但仍具獨立的排程器及CPU的執行時間，例如各執行緒具獨立的program

5. 反應時間 (response time)

定義：某個行程開始執行後到第一個產生回應的時間（例如在螢幕上看到輸出的畫面結果）。

觀念：適用交談式系統 (interactive system)，反應時間越短，使用者心裡才會滿意。

競爭條件 (Race Condition) ⑤清大資應

當兩個行程有共用資料，在不同的處理順序下會有相異的結果，即存在有不確定性問題 (non-determinism)，此一問題有時稱競爭條件。要解決此一問題，可以採互斥 (mutual exclusion) 的方式，即規定同一時間不能有兩個以上的行程同時使用共同變數。

臨界區間 (Critical Section) ⑥北科大資工、⑤中原資管

當多個行程並行合作時，採用共享記憶體 (share memory) 的方式溝通，每個行程各自存在一段特殊的程式段稱臨界區間，用來對共用變數的存取，此時合作的行程需保證不能有2個或2個以上的行程同時進入臨界區間內執行 (互斥)。下面是一些臨界區間常見的用語：

1. 進入區間 (entry section)

要進入臨界區間之前的一段控制程式。

2. 離開區間 (exit section)

要離開臨界區間之前的一段控制程式。

3. 剩餘區間 (remainder section)

除了臨界區間本身及進入區間、離開區間以外的部分稱為剩餘區間。

表示如下：

```
do {  
    進入區間  
    臨界區間  
    離開區間  
    剩餘區間
```

```
}while (1);
```

臨界區間的必要條件 (Critical Section Necessary Condition)

⑤中原資管

如果要能保證臨界區間有效的話，即同一時間，保證不能有2個或2個以上的行程同時進入臨界區間內執行，則進入區間和離開區間須滿足下列三個必要條件：

1. 互斥 (mutual exclusion)

同時最多只能有一個行程進入臨界區間。

2. 能持續進行 (progress)

當無行程在臨界區間時，只有不在剩餘區間的行程能參與競爭，而且必須在有限時間內產生優勝者，不能有死結 (deadlock) 無法決定的情形。

3. 有限等待 (bounded waiting)

當行程已經在進入區間中等待時，其他行程先行進入臨界區間的次數須有一個上限，換句話說，不能有饑餓 (starvation)，無限期等待的情況發生。

號誌 (Semaphore) ⑥北科大資工、⑤雲科資管乙、⑤中原資管

號誌是一種好用的同步工具，它分為二元號誌 (binary semaphore) 及計數號誌 (counting semaphore) 二種。計數號誌是一個整數變數 n ，除了初值設定，它只能經由和兩個標準的不可分割 (atomic) 的運算 P (wait) 及 V (signal) 來存取，利用 V 來增加 n 值，P 來減少 n 值。二元號誌與計數號誌不同之處，就在於二元號誌的號誌值只有 0 跟 1 而已，而計數號誌的 n 可以自行設定一個整數值。計數號誌 (count semaphore) 可用佇列 (queue) 或是二元號誌 (binary semaphore) 來製作。計數號誌程式碼如下：

```
P (wait):
```

```
Wait (n){
```

```
    while (n <= 0); // 若 n <= 0，則繼續check,直到 n > 0
```

```
    n--;
```

system) 為基礎，其目的是使計算機的CPU能和其多個專用的I/O裝置所產生的行程工作重疊執行。其管理上述動作的執行程式稱spooler。例如印表機的printer spooler，可以讓網路上的所有電腦共用一台印表機。

緩衝技術 (Buffering) ⑤台大資管

緩衝技術是一種可以使CPU和一個I/O工作可以同時並行運作的技術，其目的在平衡高速與低速裝置之間的差異，減少CPU閒置的時間，達到充分利用資源的目的。

多元程式 (Multiprogramming) ⑤清大資應

所謂多元程式，就是讓多個行程同時在系統中執行。它的精神在於，當一個行程在等候I/O事件時，若仍然占據CPU，會浪費CPU的資源，所以要求它放棄CPU使用權，讓其他行程可以使用CPU。因此多元程式會減少CPU的閒置 (idle) 時間，進而增加CPU使用率 (CPU utilization)。

死結 (Deadlock) ⑥雄大資管、⑨中央資管

當我們說系統發生死結時，這表示在系統內至少有一行程在等待一尚未發生的事件 (event)，而無法繼續執行，而能夠產生此事件的行程也已經進入等待中，當發生此情形時，我們稱作死結發生。會發生死結的原因，主要就是因為某些特定資源有獨占性及唯一性，即資源本身是不可強奪 (non-preemptive) 的。系統發生死結是很嚴重的事，因為除了發生死結的行程必須重新執行外，最糟的情況是系統必須重新啟動 (restart)，此外並無解決方法。本章將探討死結發生的必要條件，死結如何預防等問題。

死結發生的必要條件 (Deadlock Necessary Condition) ⑨中央資管

死結發生的四個必要條件如下：

1. 互斥 (mutual exclusion) 條件：

有些資源本質上是不能和別的行程共同使用的，這時候就會

分散式系統的優點在於能提供資源分享，計算加速，提升可靠度和跟遠端通訊的能力。

群集系統 (Clustered System) ㉖ 雄大資管

群集系統的定義為集合許多CPU以完成計算工作。與平行系統不同之處在於，它們是由兩個或更多個別系統集結而組成，彼此經由網路 (LAN) 緊密連結而成，且共同享有相同的儲存裝置 (storage device)，其目的在於提供高可用性 (high availability) 與高可靠度 (high reliability)。由以上的定義看來，群集系統也是一種分散式系統。一般來說，群集系統是分散式系統的子集，所以群集系統也是鬆散偶合系統。

容錯系統 (Fault Tolerant)

電腦系統因為採數位化的資訊處理，理論上不會發生人為的錯誤，例如人類很容易一不小心就把0寫成1，1寫成0。也就是說，電腦只要是第一次系統設計正確之後，就應該不會再發生任何錯誤。但由於真實世界中，舉凡電壓不穩定、電路設計的缺陷或是積體電路的高溫效應及人為的惡意破壞，都有可能導致一部電腦系統所負責的資料及計算能力出了問題，嚴重的情況是服務系統頓時停擺，損失可見一般。

所以現在越來越多的電腦系統採容錯系統的設計，也就是同樣的工作由兩套電腦系統來執行，當其中一套發生問題時，便立即由另外一套系統來取代，如此一來，便可允許電腦系統的錯誤，這種系統稱為容錯系統。

批次系統 (Batch System)

批次系統會先將所有要處理的工作蒐集在一起，執行的時候一個一個地依序執行，這種作業適合用於週期性且大量資料的處理。

多重開機 (Multi-boot) ㉗ 中央資管

在一部電腦上，安裝兩種以上的作業系統就是多重開機。我們可以使用電腦上的任一種作業系統，但不是同時使用。每個作

不是實體。

物件導向就是把現實世界中的物體或現象，以更接近自然的形式在電腦世界中反映出來。每一個物件都有屬於自己的屬性和方法，屬性是物件特有的性質，物件與物件之間是藉由方法來傳遞訊息，每一個物件藉由事件作出適當的反應為其處理程序。簡言之，物件導向設計就是以物件、屬性、事件、方法構成解決問題的語言。

封裝 (Encapsulation)

⑤雲科資工、⑤長庚資管、④中央資管、①四等特考

封裝是物件導向三大特性之一，封裝的動作是從要處理的資料裏面，選出抽象程度較高的成爲一個類別，而此類別中又可以分爲要處理的資料及處理資料的方法。爲了達到資訊隱藏的目的，應該將所有類別內的資料存取等級設成私有 (private)，外界只能透過此類別提供的函式來存取內部資料。

繼承性 (Inheritance)

⑤雲科資工、⑤台大資管、

⑤長庚資管、④中央資管、③台北教大資科所、①四等特考

繼承是物件導向三大特性之一，一個類別擁有你需要的部分功能，你可以採用繼承的方式，減少撰寫相同的程式碼，專注在相異的程式碼撰寫上。但是當原先的類別改變時，繼承此類別的子類別也會受影響。

多型 (Polymorphism) ⑥北科大資工、⑤長庚資管、⑤雲科資工、

④中央資管、③台北教大資科所、①四等特考

多型也是物件導向三大特性之一，當程式在執行期才真正決議 (resolve) 變數的型別，這稱爲後期繫結。多型就是應用這種技術，意含處理未知的資料型態，但是其前提是必須源至於相同的基底類別。

事件驅動 (Event Driven)

有了物件導向的概念之後，進一步要了解帶動程式運作之事件驅動觀念就容易的多。傳統的程式設計，寫程式者是主導整個