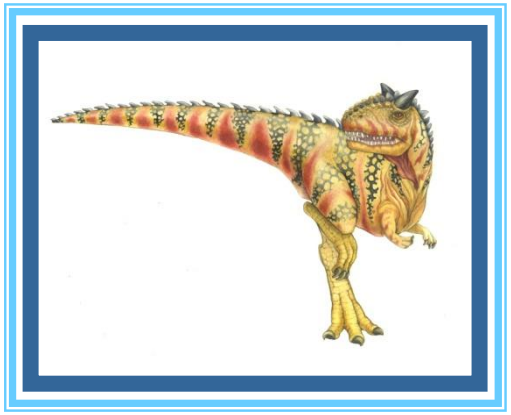
大量儲存系統



大量儲存系統

- 大量儲存體結構的概觀
- 磁碟結構
- 磁碟連結
- 磁碟排班
- 磁碟管理
- 置換空間管理
- RAID結構
- 穩定儲存體的製作

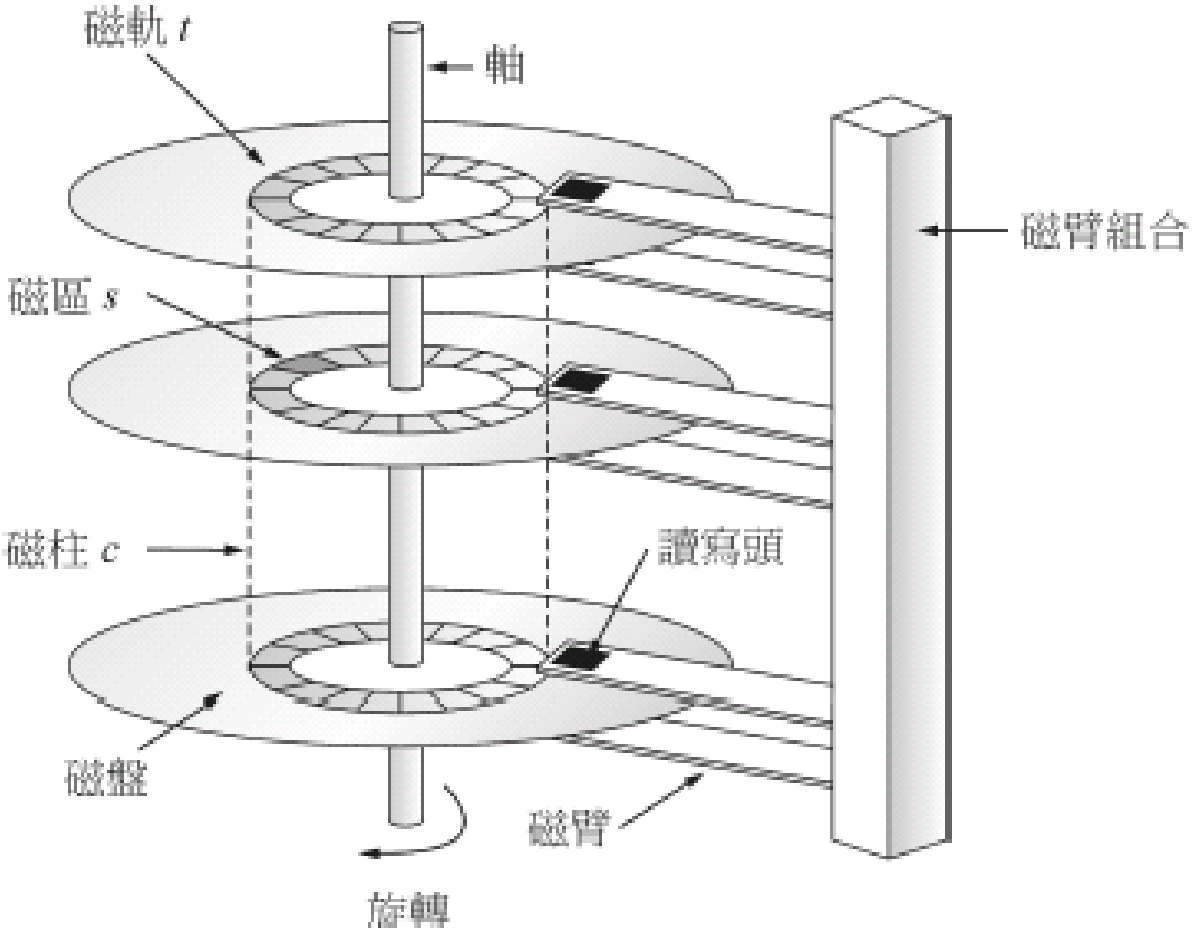
章節目標

- 描述輔助記憶體的實體架構和使用這些裝置的效果
- 解釋大容量儲存裝置的性能特徵
- 評估磁碟排班演算法
- 討論作業系統提供給大容量儲存裝置的服務，包含RAID

大量儲存體結構的概觀

- **磁碟**提供近代電腦大量的輔助儲存
  - 磁碟每秒旋轉60到250圈
  - **傳輸速率**是磁碟機和電腦間資料流動的速率
  - **定位時間（隨機存取時間）**包含移動磁臂到所在磁柱所需的時間稱為**搜尋時間**(seek time)，以及磁區轉到磁頭下所需的時間稱為**旋轉潛伏期**(rotational latency)
  - **磁頭匯損**是因為磁頭和磁碟表面接觸所造成
- 磁碟機是可以移式的
- 磁碟機經由**I/O匯流排**連接到電腦
  - 匯流排包含EIDE, ATA, SATA, USB, Fibre Channel, SCSI, SAS, Firewire
  - 電腦的**主機控制器**使用匯流排和內建於磁碟機或儲存陣列的磁碟機控制器交談

移動磁頭的磁碟機制



磁 碟

- 磁盤的大小從0.85英吋到14英吋
  - 通常是3.5英吋、2.5英吋和1.8英吋
- 磁碟機的容量從30GB到3TB
- 性能
  - 傳輸速率－理論值 6 Gb/sec
  - 有效傳輸速率－實際執是 1Gb/sec
  - 搜尋時間從3ms到12ms－桌上磁碟機一般是9ms
  - 平均搜尋時間的測量或計算是根據磁碟的 1/3
  - 潛伏期是根據旋轉速度
    - 60/RPM
  - 平均潛伏期= ½潛伏期

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

（從維基百科擷取）

磁碟機的性能

- 存取潛伏期=平均存取時間=平均搜尋時間+平均潛伏期
  - 對於最快的磁碟機　3ms + 2ms = 5ms
  - 對於慢速的磁碟機　9ms + 5.56ms = 14.56ms
- 平均I/O時間=平均存取時間+(傳輸量 / 傳輸率) + 控制器的額外負擔
- 例如，在一台7200 RPM磁碟機傳輸4KB 區段 ，平均搜尋時間是5ms, 傳輸率是1Gb/sec，控制器的額外負擔是0.1ms時，平均I/O時間=
  - 5ms + 4.17ms + 4KB / 1Gb/sec + 0.1ms =
  - 9.27ms + 4 / 131072 sec =
  - 9.27ms + .12ms = 9.39ms

第一台商用磁碟機

1956
IBM RAMDAC 電腦包含了 IBM Model 350磁碟儲存系統

5M (7 bit) 字元
50 x 24” 磁盤
存取時間 = < 1 second

固態硬碟

- 使用起來像硬碟的非揮發性記憶體
  - 許多技術變化
- 可能比硬碟更可靠
- 每MB更昂貴
- 生命期比較短
- 容量較小
- 但速度更快
- 匯流排可能太慢-> 例如直接連接到 PCI匯流排
- 沒有移動的部分，所以沒有搜尋時間或旋轉潛伏期

磁　帶

- 早期使用的輔助記憶體
- 相對地長久而且能儲存大量的資料
- 存取時間慢
- 隨機存取速度比磁碟機慢上千倍
- 主要使用在備份，儲存不常使用的訊息以及系統間傳遞資料的媒體
- 放在一個捲軸上並且通過讀寫頭向前轉動或反轉
- 一旦就定位，磁帶機傳輸速率與磁碟機差不多
  - 140MB/sec 或更快
- 容量從200GB 到 1.5TB
- 一般的技術是LTO-{3,4,5} 和T10000

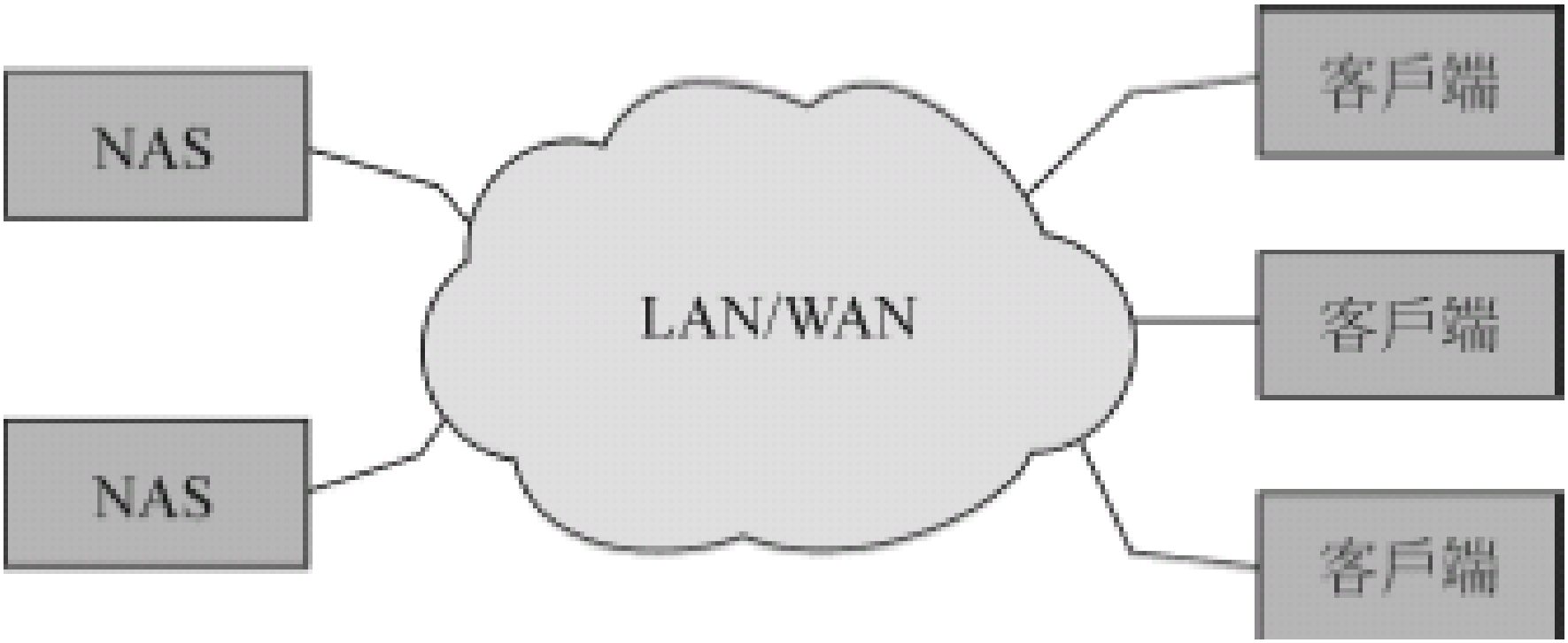
磁碟結構

- 磁碟機是以邏輯區段的一維陣列來定址，其中邏輯區段是傳送的最小單元
- 低階格式化可以在磁碟上產生邏輯區段
- 邏輯區段的一維陣列依序地映對到磁碟的磁區
  - 磁區0是在最外側磁柱第一個磁軌的第一個磁區
  - 接著依序對映到這個磁軌的全部，然後再對映這個磁柱剩餘的全部磁軌，接著從最外層到最內層對映剩餘的全部磁柱
  - 轉換邏輯區段號碼成為磁碟磁區
    - 損壞的磁區的處理
    - 固定角速度時每個磁軌的磁區數目不一樣

磁碟的連接

- 主機連結的儲存裝置是經由I/O埠和I/O匯流排交談
- 典型的桌上型PC使用稱為IDE或ATA的I/O匯流排架構
  - 這種架構每個I/O匯流排最多支援兩台磁碟機
  - SATA是一種較新類似的協定以簡化的電路相接
- 高階工作站和伺服器通常使用更複雜的I/O架構，例如SCSI或光纖通道(fiber channel, FC)
- FC是一種高速串列架構，它有兩種變形。
  - 一種是有24位元位址空間的大型交換式結構是儲存型區域網路(storage-area network, SANs)的基礎
  - 另一種架構是可以有126台裝置(驅動器和控制器)的仲裁迴路
- 資料傳遞的I/O命令是讀取和寫入邏輯資料區段到確認的指定儲存單元

網路連結儲存裝置

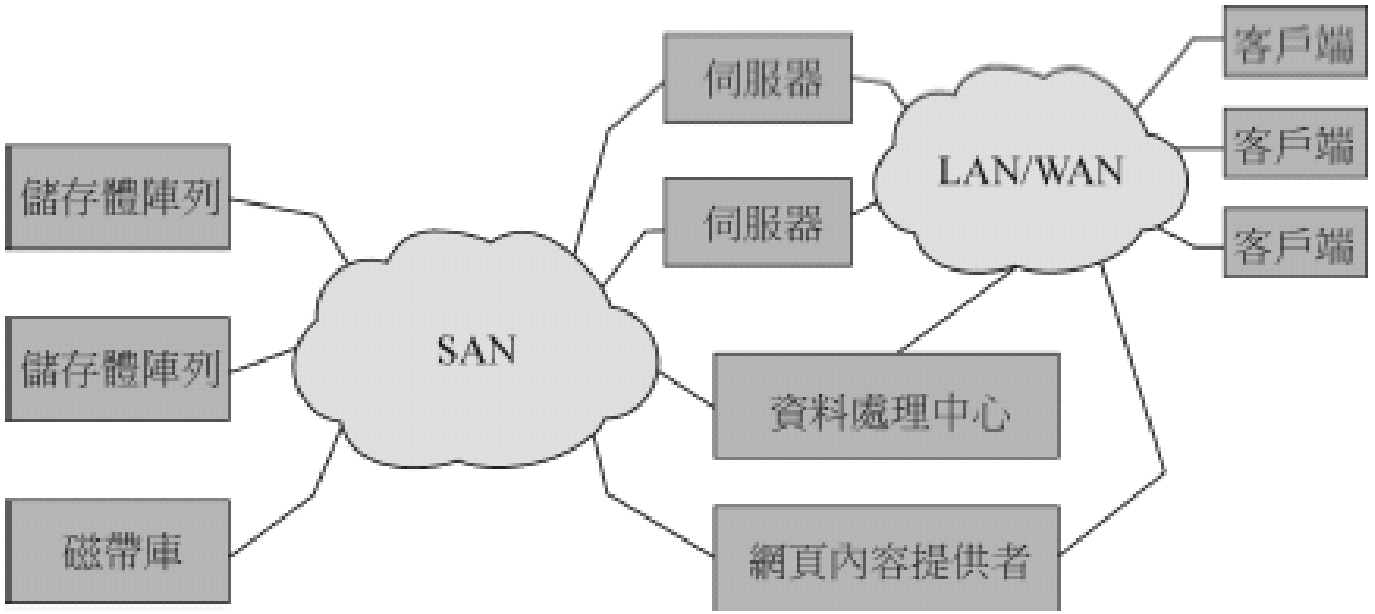


網路連結儲存裝置

- 網路連結儲存裝置(**NAS**) 是對於所有在LAN上的電腦提供一種方法共用一群儲存裝置，並且享受和區域主機連結儲存裝置相同的命名和存取方便
- NFS 和 CIFS是常見的協定
- 經由遠端程序呼叫(RPC)的介面存取網路連結儲存裝置。遠端程序呼叫在IP網路上經由TCP或UDP上執行
- **iSCSI**使用網路IP協定來實現SCSI協定
  - 主機和它們的儲存體之間互相連絡的是網路而非SCSI電線

儲存體區域網路

- **儲存體區域網路(storage-area network, SAN)**是伺服器和儲存體單元間的私人網路
  - 使用儲存體協定而非網路協定)
  - 常見於大型儲存體環境



儲存體區域網路(繼續)

- 許多主機和許多儲存體陣列可以連結到相同的SAN，而儲存體可以動態地分配給主機
- 一個SAN開關允許或禁止主機與儲存體之間的存取
- 如果主機的磁碟空間降低時，SAN可以分配更多儲存體給該主機
- FC是最普通SAN連接
- iSCSI的簡單讓iSCSI的使用增加
- 另一種SNA的連接是InfiniBand
  - 提供硬體和軟體支援的伺服器和儲存單元間高速連結的網路。

磁碟排班

- 作業系統負責有效地使用硬體 — 就磁碟機而言，這意味著快速的存取時間與大磁碟頻寬
- 降低搜尋時間
- 搜尋時間≈搜尋距離
- **磁碟頻寬(bandwidth)**的定義為傳送的位元組總數除以從第一個服務要求到最後傳送完成之間所需的總時間

磁碟排班(繼續)

- 有許多磁碟I/O 要求的來源
  - OS
  - 系統行程
  - 使用者行程
- I/O 要求包含了輸入輸出模式、磁碟地址、記憶體地址、傳輸的磁區數目
- OS 維護要求的佇列，每個磁碟機或裝置
- 閒置的磁碟機可以馬上工作，忙錄的磁碟機必須排隊服務
  - 當佇列存在時，最佳化演算法才有意義
- 磁碟機控制器有小的緩衝區，可以管理I/O 要求的佇列

FCFS排班演算法

- 先來先做(first-come, first-served, FCFS)是最簡單的一種排班法則
  - 公平的
  - 不能提供最快的服務
  - 考慮一個對於在磁柱上的區段有許多I/O要求，如以下的排列在磁碟佇列之中 ：

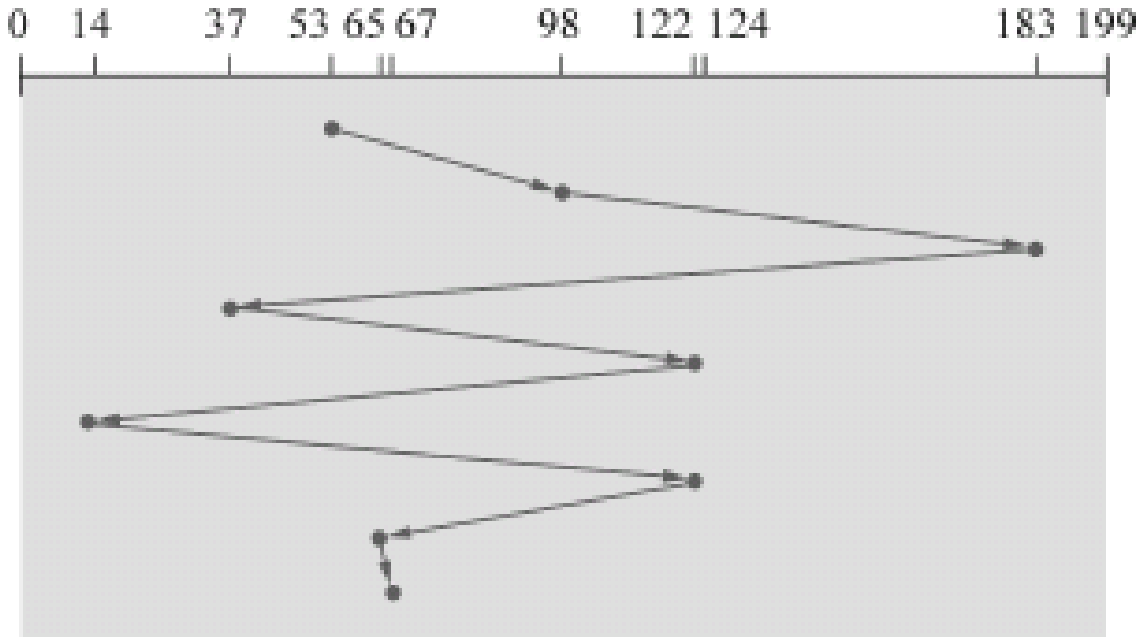
    98, 183, 37, 122, 14, 124, 65, 67
- 如果磁碟讀寫頭的起始位址為磁軌53，則它必先從53移動至98，然後再移動至183, 37, 122, 14, 124, 65，最後移動至67，故磁頭共需移動640個磁柱的距離

FCFS排班演算法

磁頭總共移動 640個 cylinders

佇列=98,183,37,122,14,124,65,67

讀寫頭自 53 啟始



SSTF排班演算法

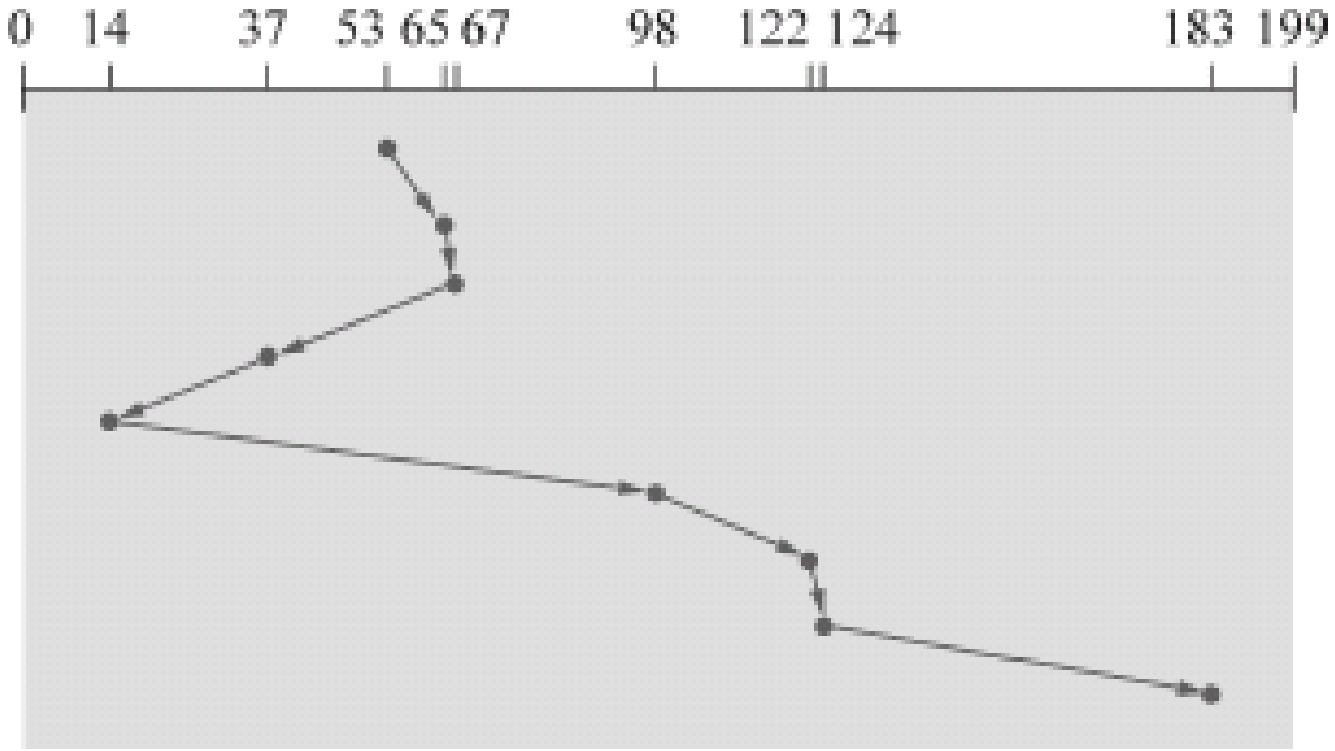
- 最短尋找時間先做(shortest-seek-time-first)SSTF
  - 根據現在的讀寫頭位置去選取出最小的搜尋時間
- SSTF基本上是SJF(shortest-job-first)排班法的型態，如同SJF排班法，可能會使一些要求產生飢餓的現象
- 考慮一個對於在磁柱上的區段有許多I/O要求，如以下的排列在磁碟佇列之中 ：

  98, 183, 37, 122, 14, 124, 65, 67
  - 要求佇列中與起始磁頭位址(53)最接近的要求是磁軌65。只要磁頭移到磁軌65，則下一個最接近的要求是磁軌67。在此，由於在37磁柱的要求比在98磁柱的要求接近，所以37是下一個服務。因此我們可以接著執行磁軌14的要求，然後是98, 122, 124，最後才至183
  - 用這排班方法，則磁頭移動的總距離僅為236磁柱

SSTF排班演算法(繼續)

佇列=98,183,37,122,14,124,65,67

讀寫頭自 53 啓始



SCAN演算法

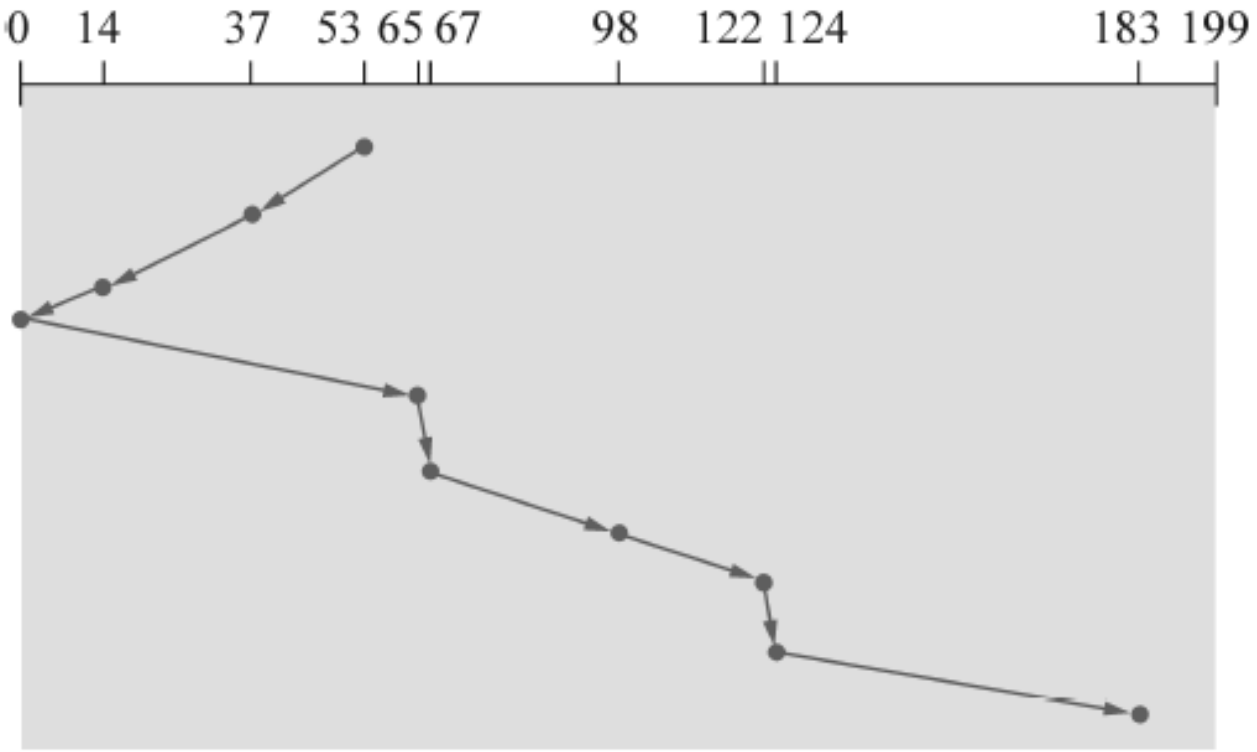
- 磁碟臂自磁碟的一端啟始，並向另一端移動，然後對其所到達之每一要求服務的磁柱服務，直到磁碟的另一端為止。而後再由另一端根據相反的順序移動磁頭，並繼續服務各要求，使得讀寫頭分別向前和向後跨越磁碟片地連續掃描
- SCAN演算法又稱為**升降梯演算法(elevator algorithm)**
- 假設對磁柱的要求是常態分佈時，最大的要求密度是在這磁碟的另一端，且具有最長的等候時間
- 考慮一個對於在磁柱上的區段有許多I/O要求，如以下的排列在磁碟佇列之中 ：

  98, 183, 37, 122, 14, 124, 65, 67
- 用這排班方法，則磁頭移動的總距離為208磁柱

SCAN演算法(繼續)

佇列=98,183,37,122,14,124,65,67

讀寫頭自 53 啓始



C-SCAN演算法

- 提供更均勻的等候時間
- 將讀寫頭自磁碟的一端移到另一端，並執行所經過的各個要求。但當它到另一端的時候，就立即返回磁碟的起始點，而不在回程服務任何的要求
- C-SCAN排班演算法基本上就是把最後一磁柱摺疊到第一個磁柱的圓形串列
- 考慮一個對於在磁柱上的區段有許多I/O要求，如以下的排列在磁碟佇列之中：

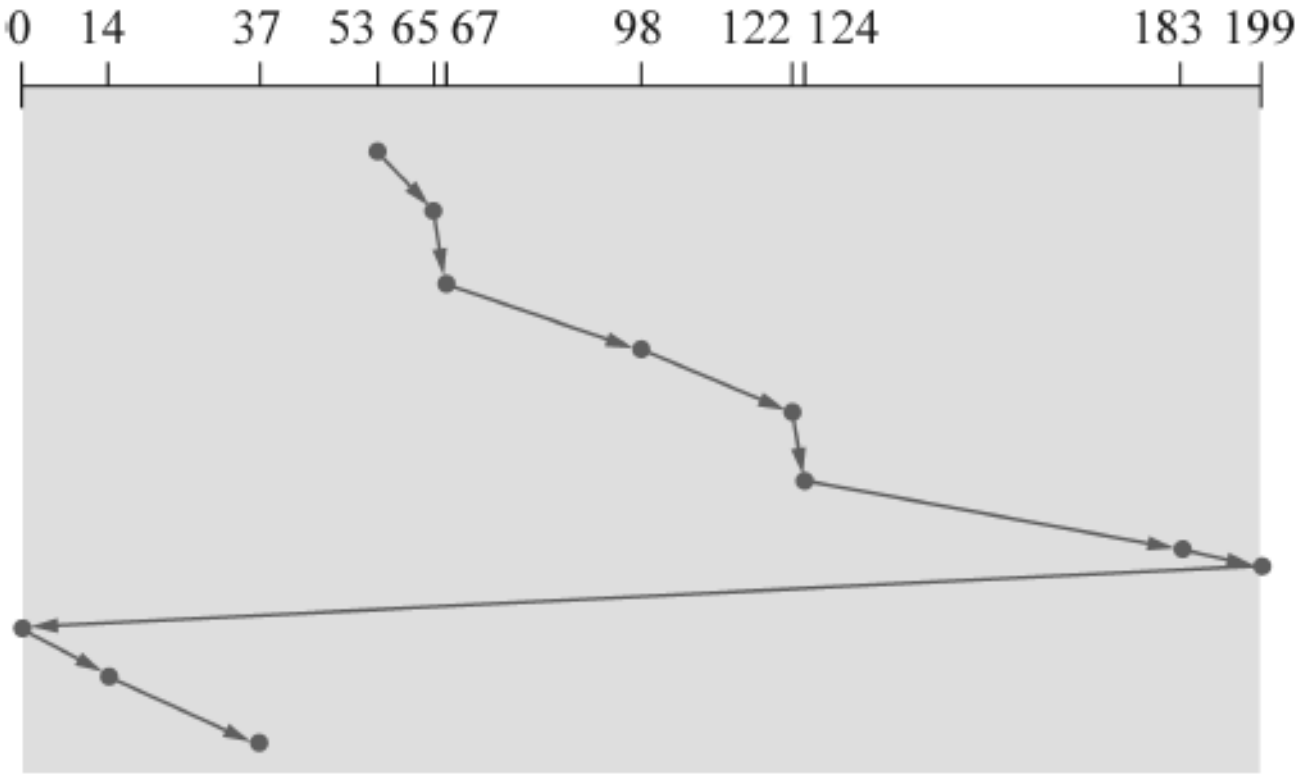
  98, 183, 37, 122, 14, 124, 65, 67

  - 用這排班方法，則磁頭移動的總距離為?

C-SCAN演算法(繼續)

佇列=98,183,37,122,14,124,65,67

讀寫頭自 53 啓始



LOOK 與 C-LOOK演算法

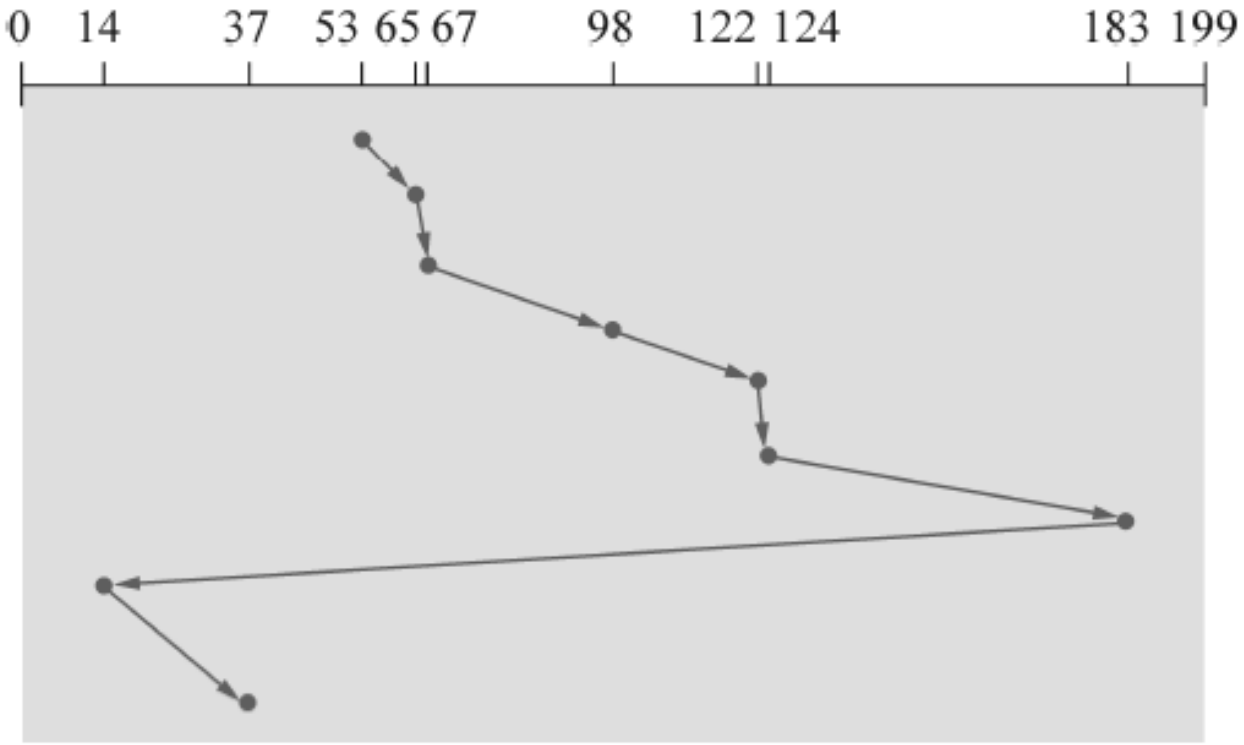
- 在每一個方向只將磁碟臂移到最後一個要求那麼遠。然後它不用先走到磁碟的所有路徑之末端，就立刻反轉方向。SCAN和C-SCAN的這些版本稱為LOOK和C-LOOK排班法則，因為它們在往已知的方向繼續移動之前，會先尋找下一個要求
- SCAN版本的稱為LOOK，C-SCAN版本的稱為C-LOOK
- 考慮一個對於在磁柱上的區段有許多I/O要求，如以下的排列在磁碟佇列之中 ：

  98, 183, 37, 122, 14, 124, 65, 67
  - 用C-LOOK排班方法，則磁頭移動的總距離為?

C-LOOK演算法(繼續)

佇列=98,183,37,122,14,124,65,67

讀寫頭自 53 啓始



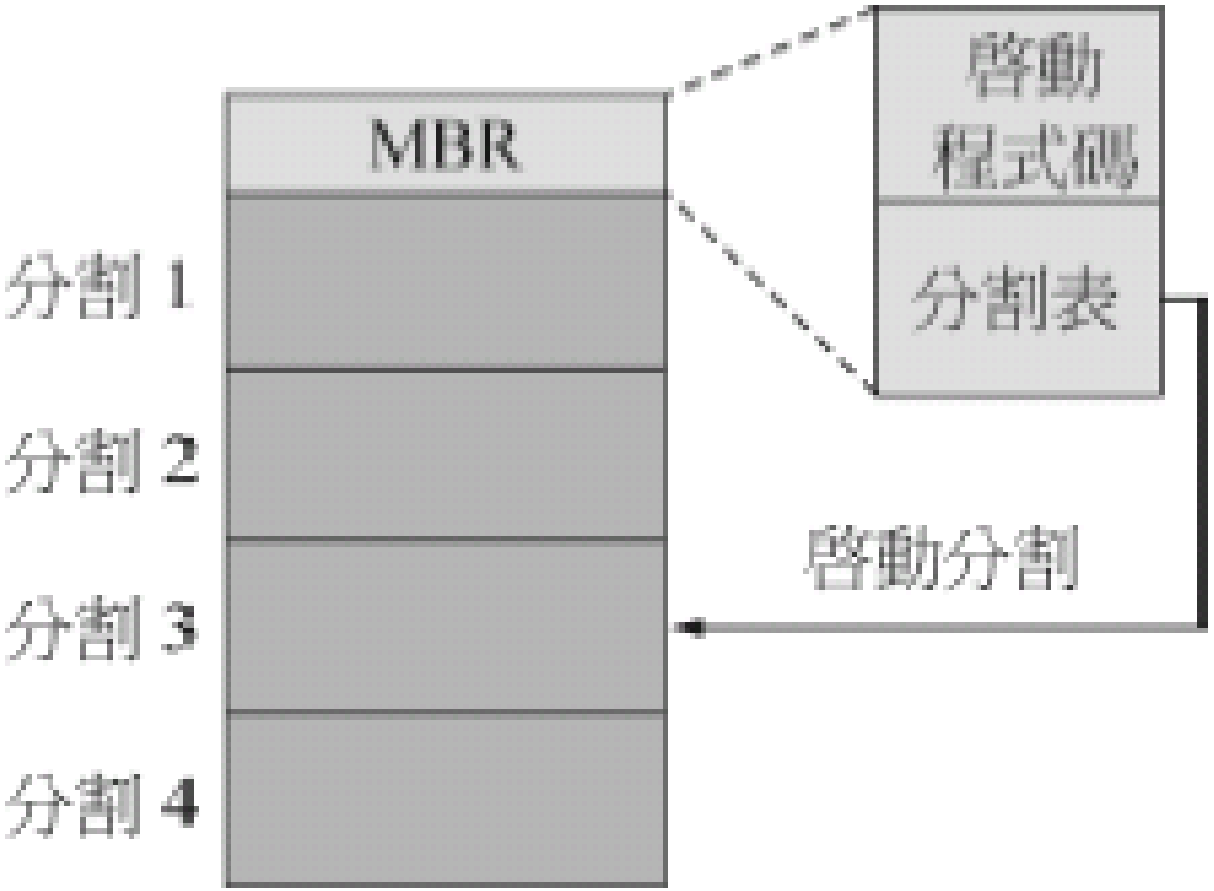
磁碟排班演算法的選擇

- SSTF是一個最通用且最為普遍的方法
- SCAN與C-SCAN法適用於大量使用磁碟的系統
  - 不會有飢餓問題Less starvation
- 性能的優劣取決於要求的多寡與型態
- 磁碟服務的要求也受檔案配置方式的影響
  - 和中介資料的排放
- 磁碟排班演算法應該寫成一個作業系統的分離模組。因此，在必要時，它可以用不同的演算法替代
- SSTF 或 LOOK都可以作為預設的演算法
- 旋轉潛伏期幾乎和平均搜尋時間一樣
  - 對於作業系統為了改善旋轉潛伏期所作的排班是困難的
- 實際的作業系統對於要求的服務次序可能有其它的限制
  - 分頁需求可能用優先順序取代應用I/O

磁碟管理

- 低階格式化或稱為實體格式化 — 將磁碟分割成磁碟控器可以讀寫的磁區
  - 每一個磁區由標頭(header)、資料區(data area，通常是以512個位元組為單位)以及錯誤更正碼(error-correcting code, ECC)所組成
  - 資料區通常是512個位元組，彈客以選擇
- 在磁碟保存資料前，作業系統需要記錄它自己的資料結構
  - 將磁碟**分割(partition)**成一個或多個磁柱群，作業系統可以視每一個分割為一個邏輯磁碟機
  - **邏輯格式化(logical formatting)**或稱為製作一個檔案系統
  - 為了增加效率，大部份檔案系統將區段聚集成**叢聚(cluster)**
    - 磁碟I/O經由區段完成
    - 檔案系統I/O經由叢聚完成
- **啟動區段**啟動作業系統
  - 靴帶式(bootstrap)程式儲存在ROM上
  - 靴帶式載入程式儲存在啟動分割區的啟動區段

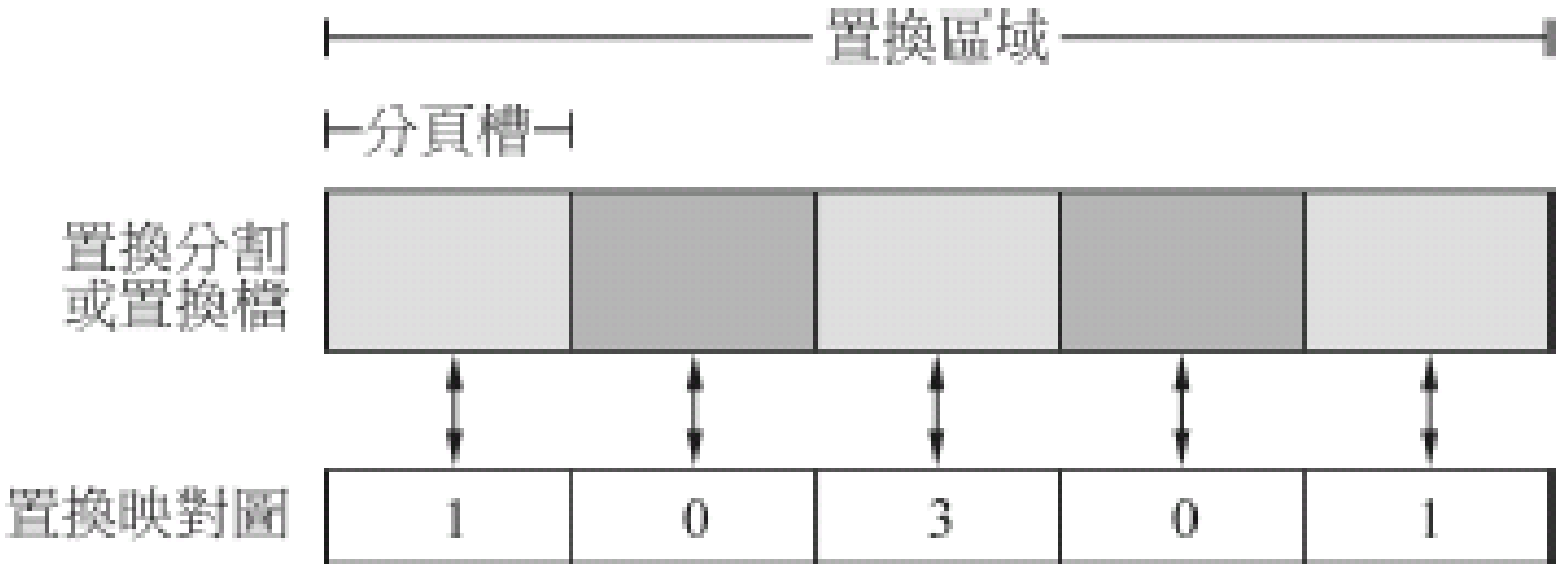
Windows中由磁碟啟動



置換空間管理

- 置換空間—虛擬記憶體使用磁碟做為記憶體的延伸
  - 因為記憶體容量增加，現在比較不普遍
- 置換空間可以產生在一個獨立的原始(raw)磁碟分割。沒有任何檔案系統或目錄結構放在此空間(raw)
- 置換空間管理
  - 有些作業系統比較有彈性，在原始的置換分割區和在檔案系統空間中都可以置換
  - Solaris 2 只有在分頁被強迫移出實體記憶體時才會配置置換空間，而不是在虛擬記憶體分頁第一次產生時配置
- 如果系統的置換空間不夠時該怎麼辦?
- 有些系統允許多個置換空間

Linux系統中置換的資料結構



RAID結構

- RAID – 不昂貴磁碟的重複陣列(redundant array of inexpensive disk)，簡稱為RAID
  - 多台磁碟機藉由重複(redundancy)提供可靠度
- 增加**平均失效時間**
- **平均修復時間** – 另一台失效可能造成資料遺失的詩間
- **平均資料遺失時間**是根據以上兩項因素
- 如果鏡像磁碟機的失效彼此獨立，考慮磁碟機的平均失效時間是1300,000小時，平均修復時間是10小時
  - 平均資料遺失時間是$100,\ 000^2\ /\ (2\ *\ 10)\ =\ 500\ *\ 10^6$ hours, or 57,000 years!
- 通常和NVRAM 結合以增進效率

RAID層次

- RAID層次0：RAID層次0是指在區段層次交插的磁碟機陣列，但是沒有任何重複的資料(例如鏡像或同位位元)
- RAID層次1：RAID層次1是指磁碟的鏡像
- RAID層次2：RAID層次2也稱為記憶體型式的錯誤更正碼組織
- RAID層次3：RAID層次3(亦即位元交錯同位位元組織，bit-interleaved parity organization)由RAID層次2精進而來，藉由考慮磁碟控制器可以偵測到一個磁區是否被正確地讀出，所以單一個同位位元可以被用來做錯誤更正和錯誤偵測
- RAID層次4：RAID層次4(亦即區段交錯同位位元組織，block-interlea ved parity organization)使用區段層次的交插(如同RAID 0一樣)，除此之外在另一台磁碟機存放其它N台磁碟機相關區段的同位位元區段
- RAID層次5(區段交插分散式同位位元)藉由把資料和同位位元分散到N+1台磁碟機，這和層次4的儲存資料在N台磁碟機，而同位位元在某一台磁碟機不相同

RAID層次

(a) RAID 0：非重複性交插

(b) RAID 1：鏡像磁碟

(c) RAID 2：記憶體型態的錯誤更正碼

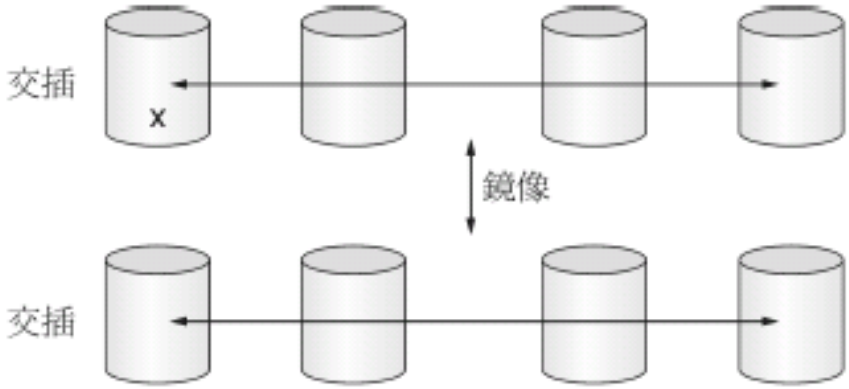
(d) RAID 3：位元交錯的同位位元

(e) RAID 4：區段交錯的同位位元

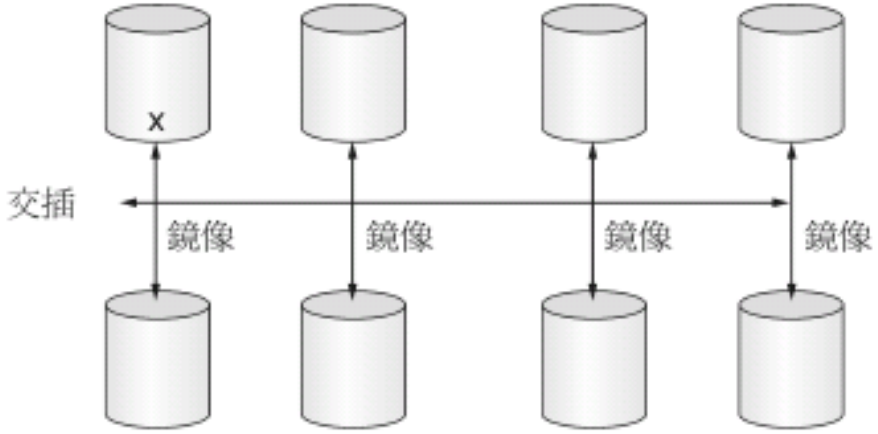
(f) RAID 5：區段交錯的分散式同位位元

(g) RAID 6：P+Q 重複

RAID (0 + 1) 和 (1 + 0)



(a) 單一磁碟機失效的 RAID 0+1

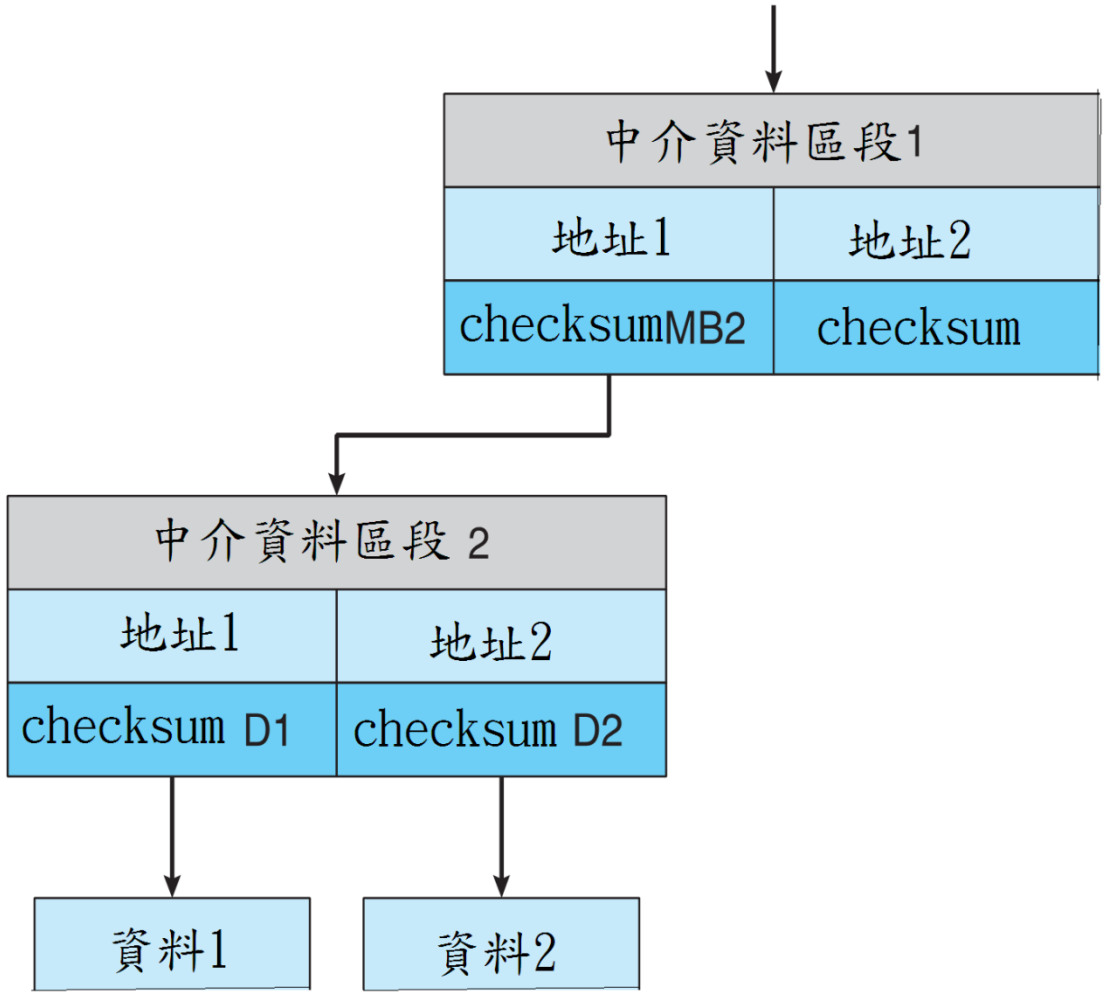


(b) 單一磁碟機失效的 RAID 1+0

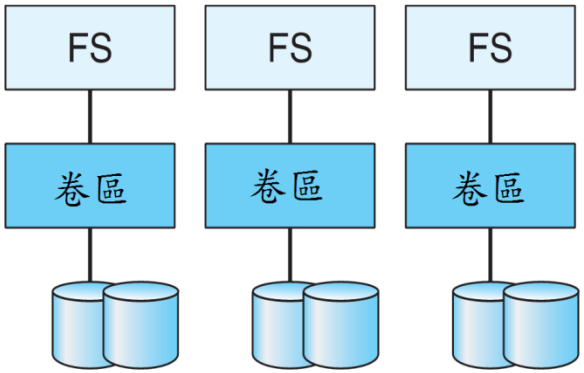
延　伸

- RAID只是防止磁碟的錯誤，但無法防止其它硬體和軟體的錯誤
- Solaris ZFS經由檢查碼(checksum)和中間資料來解決這些問題
- 檢查碼是與指向物件的指標儲存在一起，以檢查物件是否確或是否有經過修改
- 可以檢查或更正資料與中間資料的毀損
- ZFS 也移除了卷(volume)與分割區
  - 磁碟以池(pools)的方式配置
  - 檔案系統使用類似“malloc”和“free”的記憶體配置與釋放呼叫

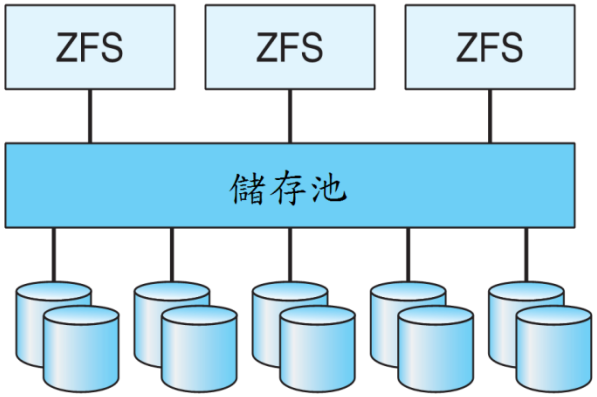
ZFS檢查所有中介資料和資料的檢查碼



傳統ZFS池儲存



(a) 傳統的卷區和檔案系統



(b) ZFS 和儲存池

穩定儲存體的製作

- 先寫入日誌的方法需要有穩定的儲存體
- 定儲存體的資料絕不會遺失
- 製作穩定儲存體：
  - 將所需要的資料複製到多種擁有獨立失效模式的儲存裝置
  - 協調更新寫入以確保更新失效時不致於留下所有複製都在毀損狀態
- 磁碟寫入造成三種結果:**成功完成**、**部份失敗**、**全部失敗**
- 在區段寫入期間發生錯誤時，我們要求系統能檢測出來並且啟動恢復程序以便復原這個區段到完整的狀態
  - 為了達到此目的，系統必須為每一邏輯區段保有兩段實體區段。輸出操作的執行步驟如下：

1. 把資料寫到第一段實體區段
2. 當第一段寫入成功之後，把相同的資料寫入第二段實體區段
3. 只有在第二次寫入成功地完成，才能宣佈操作完成