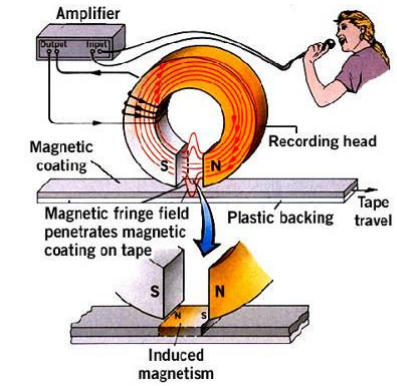
大量儲存結構

資科系
林偉川

磁帶(magnetic tape)

- 早期使用的輔助記憶體，雖然磁帶相對地一成不變而且能儲存大量的資料，但它的存取時間比主記憶體和磁碟來的慢。
- 磁帶隨機存取較磁碟隨機存取慢上千倍，所以磁帶用來當輔助儲存體不是很好用。磁帶主要使用在備份(儲存不常使用的訊息)以及系統間傳遞資料的媒體。



磁碟結構

- 磁碟機以一組稱為I/O匯流排連接到電腦上。常用幾種匯流排包含:
  - 加強整合型電子裝置(EIDE)
  - 進階技術連接(ATA)，串列進階技術連接 (SATA)
  - 萬用串列匯流排 (USB)
  - 光纖通道(fiber channel, FC)
  - 小型電腦系統介面 (SCSI)

3

磁碟(magnetic disk)

- 每個磁盤像CD，是一個平坦圓形。一般磁盤直徑由3 1/2到5 1/4吋。磁盤兩面表面上覆蓋磁性物質在磁盤上以磁性錄製方式來儲存訊息。
- 傳遞速率為資料在磁碟機與電腦間流動速率。
- 定位時間或稱為隨機存取時間
- 包含移動磁臂到所在磁柱所需的時間稱為搜尋時間
- 磁頭轉到所在磁區所需的時間稱為旋轉潛伏期 (rotational latency)。

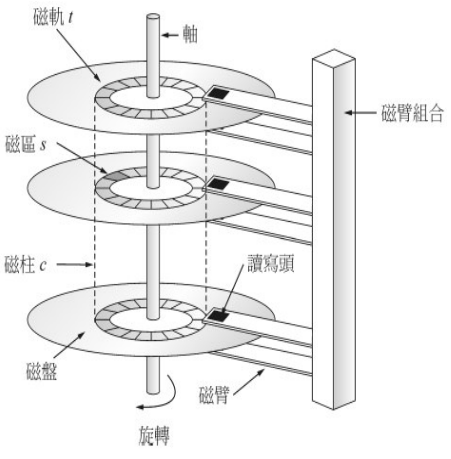


圖 12.1 移動磁頭的磁碟機制

4

磁碟結構

- 磁碟驅動程式將磁碟視為一個邏輯上的一維區塊陣列，邏輯區塊是傳輸最小的單位為512位元組，有些硬碟則可透過低階格式化來提供不同大小的邏輯區塊。
- 邏輯區塊陣列會循序地對應至磁碟的磁區，經此方法可將邏輯上的區塊編號對應至以磁柱號碼、磁軌號碼與磁區號碼定址的傳統磁碟定址方法

5

磁碟結構

- 現代的磁碟驅動器，定址成像是一個大型邏輯區段 的一維陣列，其中的邏輯區段是傳送器的最小單元。雖然有些磁碟可以用低階格式化的方式來選擇不同的邏輯區段大小(例如1024位元組)，一般的邏輯區段大小為 512位元組。(93nttu5,7, 95nttu2, 96nttu3)

6

磁碟結構

- 固定線性速度(linear velocity, CLV)，每個磁軌的位元密度是相同。從磁碟的中心來看，較遠的磁軌有較大的長度，可以擁有較多的磁區。在最外側區域的磁軌要比在最內側區域磁軌擁有多出40%的磁區數。當磁頭從外側區域移到內側區域時，磁碟會增加其旋轉速度以使得磁頭移過的資料速率保持固定。這種方法被用在CD-ROM和DVD-ROM磁碟機上。
- 固定角速度(constant angular velocity, CAV)磁碟機的旋轉速度可以保持固定，而從內側軌道到外側軌道時，位元的密度降低以保持資料速率為固定值 ，這種方法被用在硬碟上。

7

磁碟結構

- 由於磁碟技術的改進，使每顆磁碟的磁柱與磁軌數量逐漸增加，位於磁盤外側磁軌包含的磁區數量也比位於磁盤內側的磁軌多，所以每個磁軌所對應的磁區數量並不固定
- 光碟片儲存資料的密度是固定的，所以光碟內側可儲存的資料比外側少，但由於光碟機讀寫頭在讀取內圈資料時轉速較高，在讀取外圈資料時轉速較低，所以每秒讀寫頭所經過的面積會很接近，所以光碟機每秒讀取的資料也會固定，稱為常線速度

8

磁碟結構

- 常角速度
  - 如磁碟機轉動磁盤存取資料時，每秒鐘旋轉的角度是固定的 (硬碟機與軟碟機)

9

磁碟排程

- OS主要工作是有效地使用硬體，磁碟就要有較短的存取時間與較高的磁碟頻寬。磁碟的存取時間是磁碟的搜尋時間與旋轉延遲之和
  - 搜尋時間是指磁碟臂將磁頭移到目標資料磁區所在的磁柱上所花的時間
  - 旋轉延遲是指該資料磁區旋轉到磁頭所花費的時間
  - 磁碟頻寬是指從開始到完成所傳輸的位元組總量除以傳輸花費的總時間

10

磁碟排程

- 磁碟排程是降低磁碟平均搜尋時間最有效的方法，以降低存取時間，並提高頻寬來增進磁碟的執行效能(97nuut3, 92tpu1(e), 95tpu5, 96tpu8, 97tpu4)

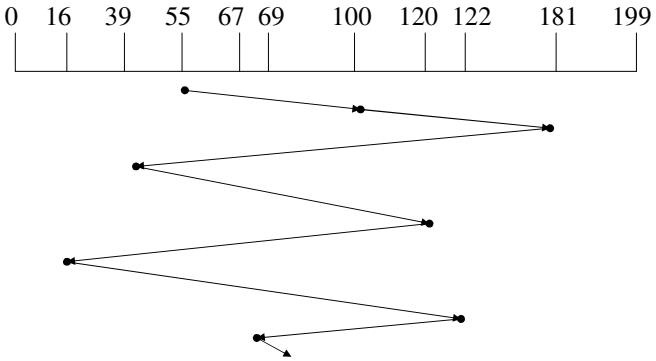
11

磁碟排程--FCFS 排程

- 驅動程式一次只接受一個要求，並依要求的次序服務，即先到先做的演算法
- FCFS 排程雖然簡單公平，卻不能提供最佳化的服務
- 假設磁碟佇列存放多個I/O要求，其磁柱編號依序為100、181、39、120、16、122、67、69，磁頭初始位置在磁柱55，則FCFS磁頭移動方式如下：

12

磁碟排程-- FCFS 磁碟排程



佇列 = 100→181→39→120→16→122→67→69
讀寫頭的起始位置55

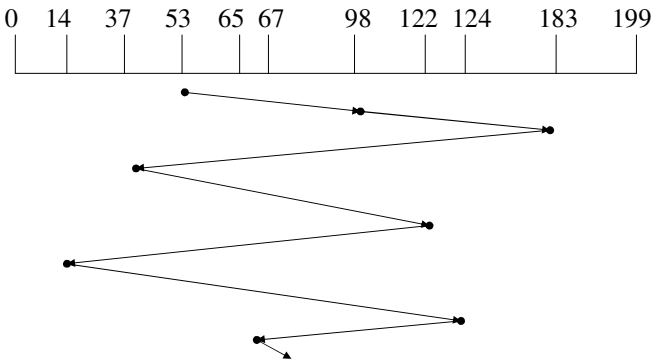
13

磁碟排程-- FCFS 排程

- 55→100→181→39→120→16→122→67→69 完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(100-55)+(181-100) +(181-39) +(120-39) +(120-16) +(122-16) +(122-67) +(69-67)=616
- FCFS 排程雖然簡單公平，卻不能提供最佳的效果
- 磁柱要求120與122及39、16，事實上可以分別循序執行，不需兩兩交錯執行，如此可以減少磁頭移動量，而且可以增加執行效能。

14

磁碟排程-- FCFS 磁碟排程



佇列 = 98→183→37→122→14→124→65→67
讀寫頭的起始位置53

15

磁碟排程-- FCFS 排程

- 53→98→183→37→122→14→124→65→67 完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(98-53)+(183-98) +(183-37) +(122-37) +(122-14) +(124-14) +(124-65) +(67-65)=640
- FCFS 排程雖然簡單公平，卻不能提供最佳的效果
- 磁柱要求124與122及37、14，事實上可以分別循序執行，不需兩兩交錯執行，如此可以減少磁頭移動量，而且可以增加執行效能。
- 94ncu、91nttu、97nttu、95tku、97tpu、95tpu、92tpu、96tpu

16

排程練習

- 假設磁碟佇列存放多個I/O要求，其磁軌編號為0-199 ，I/O要求之磁軌依序為95、189、34、120、10、126、62、68，磁頭初始位置在磁柱50，則FCFS、SSTF、SCAN、C-SCAN、Look、C-Look磁頭移動方式及讀寫頭總共移動的磁軌數？
- 假設移動讀寫頭式硬碟，其磁軌編號為0-199 ， I/O要求之磁軌次序91、183、37、122、14、124、67、65，磁頭目前位置在磁軌53，則FCFS、SSTF、SCAN、C-SCAN、Look、C-Look磁頭移動方式及讀寫頭總共移動的磁軌數？

17

排程練習

- Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 1143, and the previous request was at cylinder 3225. The queue of pending requests, in FIFO order, is 2186, 1740, 293,1774, 890, 3150, 1022, 2750, 4320, 753, 1235, 3491

  Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms? Draw a diagram to illustrate the search path. (20%)

FCFS、SSTF、SCAN、C-SCAN、Look、C-Look.

18

磁碟排程-- SSTF 排程

- 先讀取最接近目前磁頭位置的磁柱，以減少磁頭的搜尋時間
- SSTF 排程是一種最短工作優先的排程，先完成花費時間較少的工作，以降低平均等待時間，但可能導致飢餓現象發生
  55→100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭的起始位置55
- 以上述例子SSTF總位移量大約只要FCFS的1/3

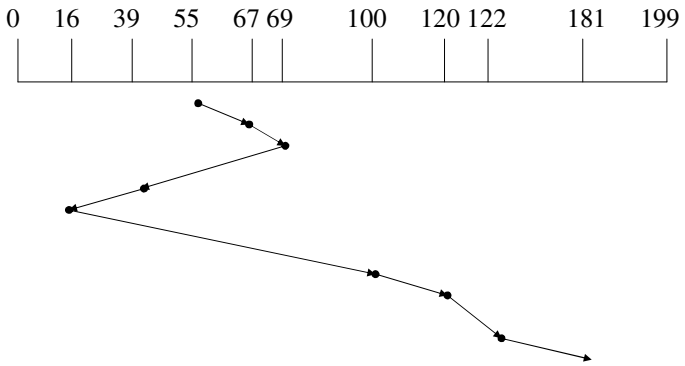
19

磁碟排程-- SSTF 排程

- 假設佇列內兩個I/O要求分別是磁柱16及184，當磁碟在服務磁柱16的要求，另一個較接近16的新要求到達，這個新要求將會優先被服務，而延遲磁柱184的服務，若服務此新要求時又有更新的要求到達，且相對於磁柱184更接近磁頭所在的位置，而又將延遲磁柱184的服務，這就是饑餓的現象

20

磁碟排程-- SSTF 磁碟排程



佇列 = 100→181→39→120→16→122→67→69
讀寫頭的起始位置 55

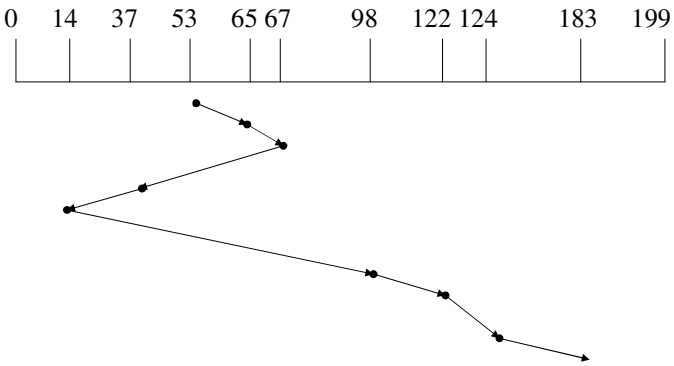
21

磁碟排程-- SSTF 磁碟排程

- 100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(67-55)+(69-67)+(69-39)+(39-16)+(100-16)+(120-100)+(122-120)+(181-122)=232
- 55→67→69→39→16→100→120→122 →181

22

磁碟排程-- SSTF 磁碟排程



佇列 = 98→183→37→122→14→124→65→67
讀寫頭的起始位置53

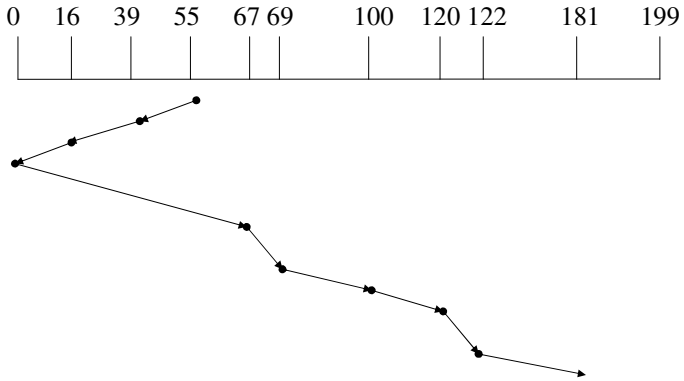
23

磁碟排程-- SCAN 排程

- SCAN 排程演算法是讓磁碟臂從磁碟的一端向另一端移動，在此移動的期間依序服務每個磁柱的 I/O 要求；當到達磁碟底端時再反轉向另一端前進，來回地存取磁碟➔電梯演算法，首先服務往上的乘客，待最上層的乘客抵達目的地後，在反向服務往下的乘客

24

磁碟排程-- SCAN 磁碟排程



佇列 = 100→181→39→120→16→122→67→69
讀寫頭的起始位置55

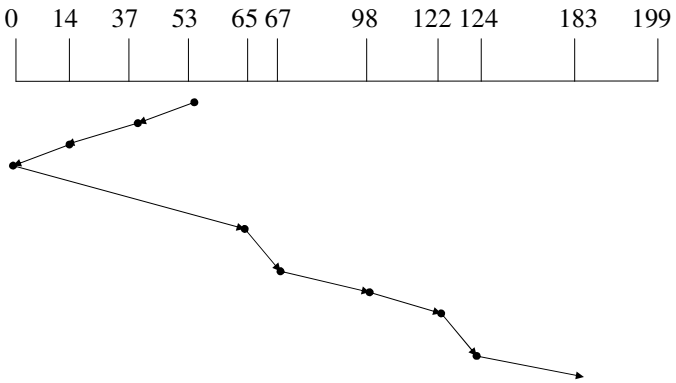
25

磁碟排程-- SCAN 磁碟排程

- 100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(55-39)+(39-16) +(16-0) +(67-0) +(69-67) +(100-69) +(120-100) +(122-120) +(181-122)=236
- 55→39→16→0 →67→69→100→120→122 →181
- 55→67→69→100→120→122 →181 →199→39→16

26

磁碟排程-- SCAN 磁碟排程(總共移動??)



佇列 = 98→183→37→122→14→124→65→67
讀寫頭的起始位置53

27

磁碟排程-- C-SCAN 排程

- 若系統對磁柱讀取的要求在磁碟上呈平均分布，當讀寫磁頭動到一端再反向時，由於先前靠近讀寫磁頭這端的要求才剛被服務過，所以要求服務的密度相較於另一端必定低很多，且另一端會因磁頭在這端連續來回服務，而造成很長的等待時間，所以不如先將讀寫頭移至另一端，由原先磁碟臂在磁碟來回服務的方式，改為由磁頭起始位置往磁碟尾端服務，到底之後再立刻拉回磁碟開頭，重新向尾端進行單向服務，此法為C-SCAN排程

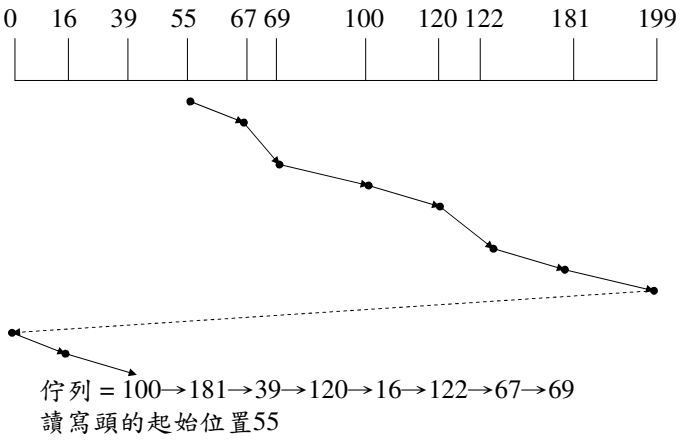
28

磁碟排程-- C-SCAN 排程

- C-SCAN 排程演算法是SCAN 演算法的變形，將磁柱視為一個循環串列，即會由最後一個磁柱繞回第一個磁柱，磁碟臂的移動如SCAN 一般，將磁頭由一端移動至另一端，沿著移動方向服務對該磁柱的存取要求，當磁碟臂到達底端時，立即移回至磁碟的開頭端，不再於回程時服務磁柱的要求，如此提供每個磁柱的要求都有一致的等待時間

29

磁碟排程-- C-SCAN 磁碟排程



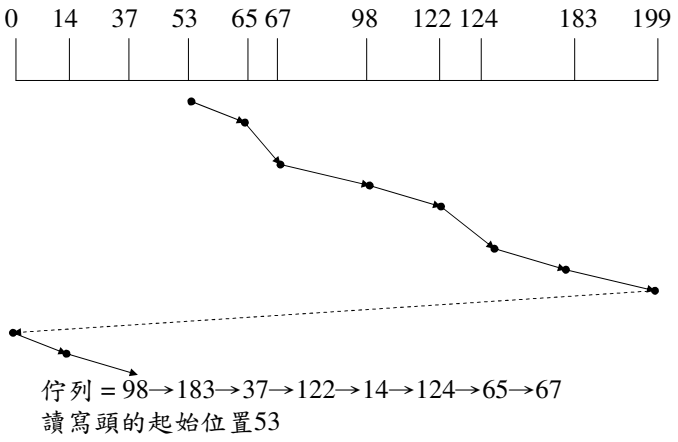
30

磁碟排程-- C-SCAN磁碟排程

- 100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(67-55)+(69-67) +(100-69) +(120-100) +(122-120) +(181-122) +(199-181) +(199-0)+(16-0)+(39-16)
- 55→67→69→100 →120→122→181→199 → 0 →16→39

31

磁碟排程-- C-SCAN 磁碟排程



32

磁碟排程-- LOOK 與 C-LOOK 排程

- SCAN與C-SCAN排程演算法，磁碟臂會移動至磁碟的兩端，所以最長移動長度就是磁碟寬度，大多數的OS並不用此二方法，而改用LOOK與C-LOOK
- LOOK 排程與 SCAN 排程相似，不同的地方在於 SCAN會讓磁碟臂來回於磁碟的兩端，而 LOOK 則是只移動到有讀寫要求的最外側及最內側磁柱

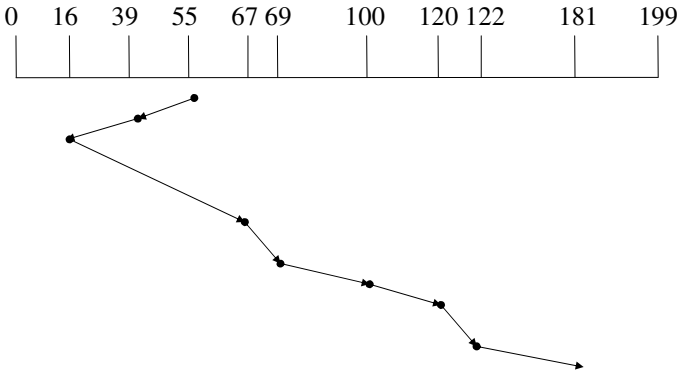
33

磁碟排程-- LOOK 與 C-LOOK 排程

- C-LOOK 及 LOOK 的差異與 C-SCAN 及 SCAN的差異均為是否將磁柱視為一個循環串列
- LOOK演算法會來回雙向地服務磁碟要求，而C-LOOK只會單向地服務後，立即再由磁碟開頭處開始服務

34

磁碟排程-- LOOK 磁碟排程



佇列 = 100→181→39→120→16→122→67→69
讀寫頭的起始位置55

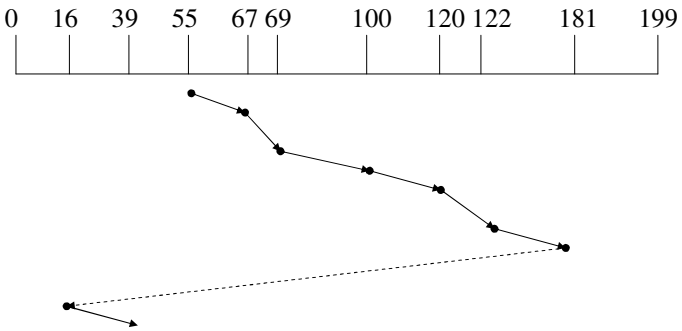
35

磁碟排程-- LOOK磁碟排程

- 100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(55-39)+(39-16)+(67-16) +(69-67) +(100-69) +(120-100) +(122-120) +(181-122)=204
- 55→39→16→67→69→100→120→122→181
- 佇列= 98→183→37→122→14→124→65→67，讀寫頭的起始位置53??

36

磁碟排程-- C-LOOK 磁碟排程



佇列 = 100→181→39→120→16→122→67→69
讀寫頭的起始位置55

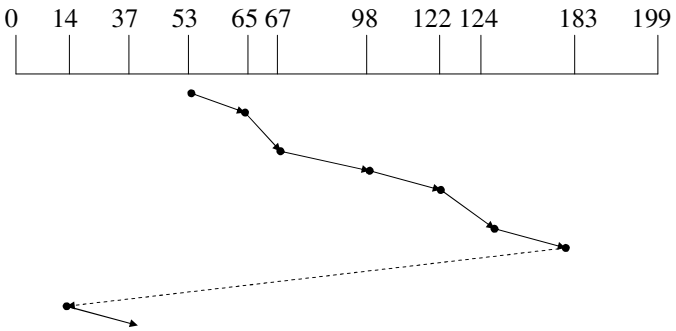
37

磁碟排程-- C-LOOK磁碟排程

- 100→181→39→120→16→122→67→69
  完成以上8個I/O要求，讀寫頭總共移動的磁軌數為(67-55)+(69-67)+(100-69) +(120-100) +(122-100) +(181-122)+(181-16)+(39-16)=334
- 55→67→69→100→120→122→181→16→39
- 佇列= 98→183→37→122→14→124→65→67，讀寫頭的起始位置53??

38

磁碟排程-- C-LOOK 磁碟排程



佇列 = 98→183→37→122→14→124→65→67
讀寫頭的起始位置53

39

磁碟排班演算法選擇

- 在許多磁碟排程演算法中，要選擇一個最適合系統的演算法相當困難，SSST已被許多OS所採用
- SCAN和C-SCAN適合負載較大的磁碟，因為負載大，所以浪費在磁頭移動的時間相對減少，較不會產生飢餓問題
- 對一連串磁碟I/O要求，可以先定義一個服務最佳順序，但是需要額外的計算時間與硬體設計，所以不見得是最佳的效率

40

磁碟排班演算法選擇

- 在這麼多的磁碟排班演算法中，如何挑選出最好的一個?
- SSTF是一個最通用且普遍的方法，因為SSTF較FCFS增加效能。
- SCAN與C-SCAN法適用於大量使用磁碟的系統，因為它們比較不會有飢餓問題
- 對於任何一個要求串列而言，可以定義出一個最佳化的修正順序，但是最佳化的排班需要許多的計算，無法認同SSTF或是SCAN法的節省性。

41

演算法選擇

- 排程演算法的執行效率會隨磁碟要求的先後順序和要求的數量而改變，所以很難評估磁碟排程演算法的優劣
- 磁碟服務也受檔案配置方式的影響，例如讀取鏈結或索引檔時可能需要讀取包含數個分散於磁碟各處的區塊，所以產生較多的磁頭移動，假設一筆紀錄是在第一個磁柱，檔案資料在最後一個磁柱，磁頭必須移動很長的距離才能完成這個檔案存取，但可藉快取機制解決

42

演算法選擇

- 作業系統大多將演算法寫成一個獨立的模組，以便對不同情況能夠使用不同的磁碟排程演算法
- 除了降低搜尋時間外，現今的旋轉延遲幾乎和平均搜尋時間一樣長，由於OS無法得知邏輯區塊的實際位址，所以很難在OS內改善旋轉所造成的延遲，有些磁碟製造商已經將磁碟演算法實作在磁碟控制器內，來解決此問題

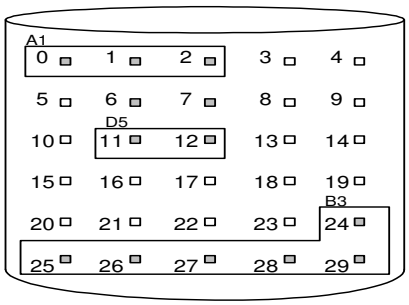
43

輔助儲存體之管理

- 檔案在磁碟上配置之方式一般可分成下述三種：
  - 連續式配置法 (Contiguos Allocation)
  - 鏈結式配置法 (Linked Allocation)
  - 索引式配置法 (Indexed Allocation)

44

- 連續式配置法是每個檔案佔用磁碟上的一組連續位址；即每個檔案配置一組連續的區塊以供該檔案使用。若沒有一塊足夠大的連續空間，則此檔案就無法建立。



磁 碟

目錄表

檔名	起始區塊	長度
A1	0	3
B3	24	6
D5	11	2

45

- 連續式配置法優點：
  - 邏輯記錄的次序與實際記錄的儲存次序相同，因此存取速度較快。
  - 檔案目錄表結構簡單，容易建立，每一個檔案只需要記錄起始位址和長度。

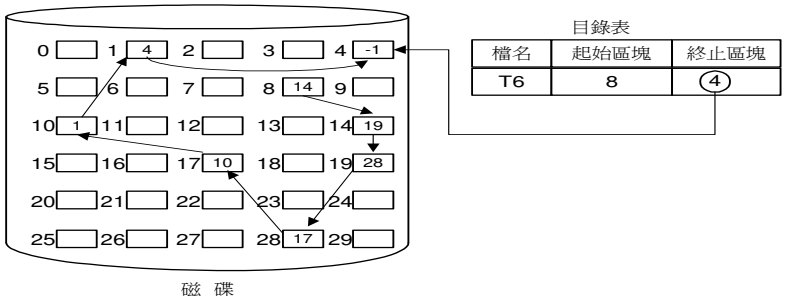
46

- 連續式配置法缺點：
  - 當檔案被刪除時，空間回收時會產生碎片，因為新的檔案大小不一定會恰好等於這空間。
  - 因要維持一個足夠大且連續的可用空間，則需要定期的執行壓擠工作。
  - 因為檔案在執行輸出入時，檔案會因此有擴張的現象，所以需要預留可能擴張的空間。若是空間預留太大，則會造成浪費；若不夠大時，則須重新把檔案移到較大的空間上。(defrag.exe??)

47

- 鏈結式配置法－

  每個檔案不需要佔用連續的區塊，其佔用的區塊可以散佈於磁碟上的任何位置，區塊與區塊之間利用指標將之相連。每個檔案由一個起始指標指向檔案的開端，檔案擴張時，則利用 GetNode 的方式來取得額外的區塊，以便儲存衍生出來的資料；縮小時，利用 Return的方式來歸還過多的區塊。



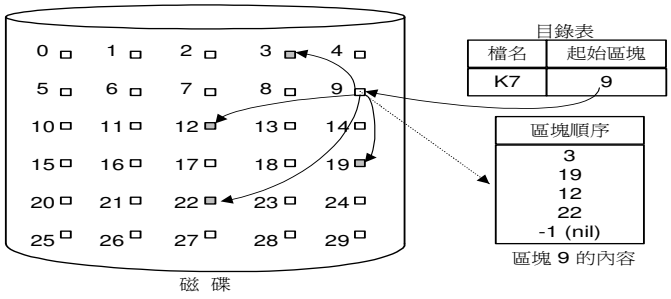
48

- 鏈結式配置法 優點：
  - 克服連續式配置法的缺點。
- 鏈結式配置法缺點：
  - 取出邏輯上連續的區塊需要花較長時間尋找。
  - 建立並維護一個鏈結串列結構，需要增加額外的時間。
  - 指標需佔用一些空間。

49

- 索引式配置法—

  每個檔案不需要佔用連續的區塊，其佔用的區塊可以散佈於磁碟上的任何位置。其方式是每個檔案擁有自己的索引區塊，該索引區塊內含有一些指標，藉以指向配置給該檔案的區塊。



50

- 索引式配置法優點：
  - 克服連續式配置法的缺點。
- 索引式配置法缺點：
  - 索引區塊需佔用額外的空間，且其大小與檔案的大小有關。

51

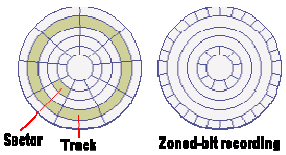
磁碟格式化

- 低階格式化
  - 磁碟在使用前必須先經過磁區的劃分，使得磁碟控制器能夠讀取與寫入這些磁區
  - 對每個磁區填入一特別的資料結構：標頭、資料區(512為元組)和結尾
  - 標頭和結尾包含磁碟控制器所需要的資訊(磁區號碼和ECC)，當磁碟控制器執行一般的I/O寫入時，會依照磁區內的資料重新計算並更新ECC，磁區於讀取時，ECC也會重新計算比較，檢查是否與原來儲存的值一致
  - ECC是磁碟控制器於讀取或寫入時自動處理的程序

52

磁碟格式化

- 在磁片可以儲存資料之前，它必須被區分為磁碟控制器可以讀寫的磁區，這個過程稱為低階格式化或稱為實體格式化，低階格式化對磁碟的每一個磁區填入一種特定的資料結構。
- 每一個磁區的資料結構是由標頭、資料區 (通常是以512個位元組為單位)以及結尾所組成。標頭與結尾包含為磁碟控制器所使用的資訊，
- 磁區號碼與錯誤更正碼(error-correcting code, ECC)。當控制器寫入一個磁區的資料到一般的I/O之中，ECC的值用所有在資料區之中的位元組之計算值更新。在讀取這個磁區時，重新計算ECC的值，並且與儲存的值比較。如果儲存的和計算的值不同，這個不吻合指出這個磁區的資料區已經變不可靠，這個磁區可能是壞的。



53

Error Correction Code (ECC)

- 錯誤改正可以分為兩大類：linear block code, cyclic code
- 線性區塊碼的一種特例為cyclic code
- cyclic code的 一種特例為BCH code
- BCH code的一種特例為RS code(Reed-Solomon code 通道編碼技術)
- Parity check、CRC and Hamming Code(漢明碼)

54

CRC

- CRC是Cyclical Redundancy Check的縮寫，這是一種用在資料傳輸之後驗證正確性的演算法
- CRC可以分為16位元的CRC-16與32位元的CRC-32兩種，前者使用16位元的數字運算來產生一個CRC碼，一般說來使用將所收到的訊息除以一個2進位數字，所得到的餘數(R(x))便是CRC碼，16位元的CRC適合資料量在4 KB以下的傳輸，它可以檢查出99.998%的錯誤

55

CRC

- 32位元的CRC則使用更長2進位數字作為除數，它適合用於64 KB以下的資料傳輸，可以檢查出99.999999977% 的錯誤，一般作為乙太網路及Token Ring網路的錯誤檢查

56

CRC

- Message set consists of the numbers 0 and 1 with arithmetic defined modulo 2. It means that you perform arithmetic as usual, but if you get something greater than 1 you keep only its remainder after division by 2. In particular, if you get 2, you keep 0. Here's the addition table:
  0 + 0 = 0，0 + 1 = 1 + 0 = 1
  1 + 1 = 0 (because 2 has remainder 0 after dividing by 2) ➔XOR運算

57

CRC

- In fact the subtraction table is identical to the addition table and so is division (except for division by zero). From the point of view of computer programming, is that both addition and subtraction modulo 2 are equivalent to bitwise *exclusive or* (XOR).

58

CRC

- A polynomial with coefficients modulo 2 can be represented as a series of bits. Conversely, any series of bits can be looked upon as a polynomial. Any binary message, which is nothing but a series of bits, is equivalent to a polynomial.
- First calculate the CRC for a message to which you have appended 32 zero bits. Suppose that the message had N bits, thus corresponding to degree N-1 polynomial. After appending 32 bits, it will correspond to a degree N + 31 polynomial.

59

CRC

- The top-level bit that was multiplying X^{N-1} will be now multiplying X^{N+31} and so on. In all, this operation is equivalent to multiplying the message polynomial by X^{32}.
- Take a binary message and convert it to a polynomial then divide it by another predefined polynomial called the *key*. The remainder from this division is the CRC.

60

CRC

- Now transmit both the message and the CRC. The recipient of the transmission does the same operation (divides the message by the same key) and compares his CRC with yours. If they differ, the message must have been mangled. If, on the other hand, they are equal, the message went through uncorrupted.
- If we denote the original message polynomial by M(x), the key polynomial by K(x) and the CRC by R(x) (remainder) we have:

61

CRC

- $M * X^{32} = G(x) * K(x) + R(x)$
- Now add the CRC to the augmented message and send it away. When the recipient calculates the CRC for this sum, and there was no transmission error, he will get zero. That's because: $G(x) = X^{32}$
- $M * X^{32} + R(x) = G(x) * K(x)$ (no remainder!)
- Maybe it should be -R(x) on the left. However, that in arithmetic modulo 2 addition and subtraction are the same (XOR運算)

62

練習

- 若原始欲傳送訊息為M(x)= 10101011，多項式$G(x)=X^3+1$，則傳送訊息T(x)為何？(A) 10101011001 (B) 10101011010 (C) 10101011100 (D) 10101011111
- 若原始欲傳送訊息為M(x)= 110011，多項式$G(x)=X^4+X^3+1$，則傳送訊息T(x)為何？(A) 110011 1001 (B) 110011 1010 (C) 110011 1100 (D) 110011 1111

63

練習

- 下列何者不符合偶同位？(A) 11111111 (B) 00000000 (C) 10101010 (D) 01001001
- 若採用偶同位方式來傳送7位元資料，下列各筆已接收的資料中，何者在傳送過程已發生錯誤？ (A) 00001111 (B) 00110001 (C) 11111111 (D) 10101010
- 若採用偶同位方式來傳送7位元資料，下列各筆已接收的資料中，何者在傳送過程已發生錯誤？ (A) 10110001 (B) 00000000 (C) 11111111 (D) 10101011
- 若採用奇同位方式來傳送7位元資料，下列各筆已接收的資料中，何者在傳送過程已發生錯誤？ (A) 01100001 (B) 11000011 (C) 01010100 (D) 11010101

64

Hamming Code

- Hamming Distance(漢明距離) H_d
- Error Detection(錯誤偵測能力) H_d-1
- 更正能力 $[(H_d-1)/2]$
- CRC只能偵測無法更正，漢明碼可以更正

65

Hamming Code

- 當要傳送n個位元的資料時，必須插入r個同位位元，且n和r要滿足 $2^r >= n+r+1$
- 同位位元的位置安插在資料的第1、2、4、8…位元處。
- $P_1P_2D_3P_4D_5D_6D7P_8D_9D_{10}D_{11}D_{12}$ →→→→→→ ……→→→→→→ $p_1p_2d_3p_4d_5d_6d_7p_8d_9d_{10}d_{11}d_{12}$
- $P_1 = D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$
- $P_2 = D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$
- $P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$
- $P_8 = D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$

66

練習

- 給兩個表示式010101和111110，則此二表示式的漢明距離為何？(A) 2 (B) 3 (C) 4 (D) 5
- 給兩個表示式011011和100011，則此二表示式的漢明距離為何？(A) 2 (B) 3 (C) 4 (D) 1
- 下列哪個碼與010011之漢明距離為3？ (A) 000001 (B) 011111 (C) 001101 (D) 001111
- 經傳輸後的漢明碼為0011000，其原始資料為？ (A) 0100 (B) 1010 (C) 1000 (D) 1001

67

練習

- 漢明距離為5的一組編號，至多能校正幾個位元的錯誤？ (A) 1 (B) 2 (C) 3 (D) 4
- 漢明距離為7的一組編號，至多能校正幾個位元的錯誤？ (A)2 (B) 3 (C) 4 (D)5
- 下列何者錯誤？(A) 資料為1011，則漢明碼為0110011 (B)漢明碼為0111110，則第6個位元是錯誤的 (C)資料為10110，以漢明碼表示時同位元需要4個 (D)漢明碼為011001000，正確資料為11100

68

練習

- 利用奇同位，將資料10110轉成漢明碼傳送，其值為何？ (A) 1011001100 (B) 1011011100 (C) 1100011100 (D) 1001100110
- 一組8位元的資料，經過漢明碼編碼後成為一組12位元的資料，假設該組資料經傳送後在接受端所收到的資料為000011101010，則正確的8位元資料為何？ (A) 11101010 (B) 01111010 (C) 00001100 (D) 01011010

69

練習

- 假設原始碼為6bit，利用漢明碼編碼後，最少需加入幾位元，才能達到更正一個位元的功能？ (A) 3 (B) 4 (C) 5 (D) 6
- 假設原始碼為4bit，利用漢明碼編碼後，最少需加入幾個檢查碼，才能達到更正一個位元的功能？ (A) 1 (B) 2 (C) 3 (D) 4
- 利用Huffman編法方法，找出AABBBBCCCDDDDD字串之最小編碼長度為多少位元？ (A) 28 (B) 32 (C) 26 (D) 30

70

磁碟格式化

- 低階格式化可以幫助製造商測試硬碟好壞，多數硬碟於硬碟控制器發出低階格式化命令後，會決定在每一個磁區的標頭和結尾之間可以使用的資料空間大小(256、512、1024位元組)，為使硬碟能使用這些空間來儲存檔案，OS要在硬碟上記錄自己的資料結構，需透過磁碟分割、和邏輯格式化的動作。
- 磁碟分割
  - 將磁碟分成一個或多個磁柱的群組
  - 作業系統會將這些磁碟分割視為一個個獨立的磁碟

71

磁碟格式化

- 邏輯格式化
  - 作業系統會儲存初始的檔案系統資料結構於磁碟之中；資料結構中包含未被配置和已配置的空間、及一個初始的空白目錄
- 某些OS允許將一磁碟分割當作一大的邏輯區塊陣列來存取，整個磁碟空間可稱為一個裸磁碟，裸磁碟只提供裸I/O的基本服務，使用裸I/O可以發展更有效率的應用程式，並在裸磁碟上實作特殊目的的儲存服務。

72

練習

- 假設有雙面磁片具有下列特質：
  - 每面具有35個磁軌
  - 每個磁軌具有10個磁區(X)
  - 每秒傳送速率為250000位元/秒➔每轉一次??位元
  - 每分鐘300轉➔每秒轉??次

  就上述資料計算該磁片之容量為多少個位元組？
  (250000/(300/60))*35*2=437500位元組

73

練習

- 假設有一磁片結構具有下列特質：
  - 有雙面，每面具有203個磁軌
  - 每個磁軌具有512個位元組

  (a)就上述資料計算該磁片之容量為多少個位元組？
  2*203*512=203KB

  (b)若OS以512個位元組當作一個區塊，而磁碟控制卡硬體所接受的磁碟位址格式為(bit 0-12)

磁軌(5-12)	面(4-4)	磁區(0-3)

  則寫出將區塊轉成磁碟位址X之公式為何？
  一區塊=一磁軌，基本單位與磁區無關但與磁軌有關，(0-3)皆為0，$X=X/2*2^5+(X \bmod 2)*2^4+0$

74

練習

- 假設有一磁片經格式化後有10個磁軌，每個磁軌具有個20磁區。磁片在磁碟機中可以採順時鐘或逆時鐘方向旋轉。已知：
  - 每轉一圈需200ms
  - 將磁頭由磁片中央移至磁片邊緣需20ms
  - 傳送一個區塊的資料需0.3ms

  若有三個檔案A、B與C，其儲存於磁片上之位置與檔案大小分別描述如下：

75

練習

檔案A位置：第6磁軌，第1磁區；大小：2個區塊

檔案B位置：第2磁軌，第5磁區；大小：5個區塊

檔案C位置：第5磁軌，第3磁區；大小：1個區塊

試問

(a)磁頭目前位置在第0磁軌，第0磁區，且欲讀取檔案A、B與C，則應以何種讀取順序，才能使讀取時間花費最少？

(b)磁頭目前位置在第0磁軌，第0磁區，試求讀取檔案C、B、A所需時間為何？

(c)試求平均潛伏時間(Latency Time)？

(d)試求平均搜尋時間(Seek Time)之最大與最小值？

76

練習

(a)轉一圈需200ms，每個磁軌具有個20磁區，每磁區需用10ms，從最內圈至最外圈需20ms，磁片共10個磁軌，每經一磁軌需用2ms。經過一磁區的時間是磁軌的五倍，應減少經過磁區，儘可能走磁軌並以磁區之順序依次讀取。故磁頭移動為磁區0→2→5→6，為B →C→ A

(b)磁頭移動順序若為C➔B➔A，磁軌移動順序0→5→2→6，磁區移動順序0→3→5→1，共移動(5+3+4)磁軌與(3+2+4)磁區，且共傳輸8個區塊故總共耗時(12*2+9*10+8*0.3) ms

(c)平均潛伏時間為旋轉時間，只旋轉至指定之磁區所經過之磁區最少為0個磁區，最多為20個磁區，平均會經過10個磁區，故為10*10ms

77

練習

(d)平均搜尋時間指將磁頭移至所指定的磁軌會耗用時間：

若磁頭在最內圈，最少移動磁軌為0個磁軌，最多移動磁軌為10個磁軌，平均為5個磁軌， 故為5*2ms

若磁頭在最內圈與最外圈的中間，最少移動磁軌為0個磁軌，最多移動磁軌為5個磁軌，平均為2.5個磁軌， 故為2.5*2ms

所以平均搜尋時間介於5至10ms

95tku、96tku、96nttu、93nttu

78

啟動區塊

- 對一台電腦開始運作而言(例如，當打開電源或重新啟動)，它需要有一個啟始靴帶式(boot strap)程式先執行。它對系統所有的事做初始化，從CPU的暫存器到裝置控制器以及主記憶體內容，並接著啟動作業系統。
- 靴帶式程式是儲存在唯讀記憶體中。這個安排是因為ROM不需要初始化，而且它是在打開電源或重置的時候，處理器可以開始執行的固定位置上。

圖 12.9 Windows 2000 中由磁碟啟動

79

啟動區塊

- 電腦的在打開或重置電源之後，需要一個初始化電腦的啟動載入程式(bootstrap program)，負責初始化 CPU 暫存器、裝置控制器、和主要記憶體內容，然後執行作業系統
- 啟動載入程式會先找出磁碟內OS存放位置，將OS核心載入到記憶體，最後再跳至OS開始執行的初始位置

80

啟動區塊

- 大部份的電腦啟動載入程式儲存在ROM上固定的位置，使CPU於啟動時可以直接從此位址開始執行，因為ROM不會受電腦病毒感染，但若有修改啟動載入程式碼的需要，則有更新ROM的問題，所以大部分的系統只有在ROM儲存一小段啟動載入程式，此小段程式主要作用在從磁碟上載入完整的啟動載入程式，這樣的作法使得完整的啟動載入程式能夠容易地被更新，而不需要更動ROM中的內容

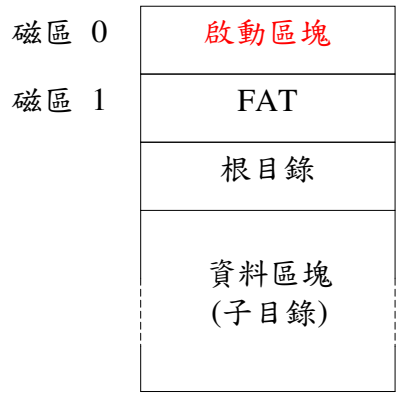
81

啟動區塊

- 完整的啟動載入程式是儲存在一個特別的磁碟分割中，稱為啟動區塊 (MBR)
- 含有啟動區塊的磁碟稱為啟動磁碟(分割)
- ROM中的程式，命令磁碟控制器讀取啟動區塊內的程式，等載入到記憶體中之後，開始執行啟動載入程式來載入並執行存放在磁碟不固定位置的OS，通常完整的啟動載入程式都很小，MS-DOS只使用512個位元組

82

MS-DOS 磁碟資料存放方式



83

壞損區塊

- 磁碟使用非常頻繁，所以很容易損毀，甚至有壞軌，一但有壞軌就需要更新新硬碟，並備份原硬碟的資料
- MS-DOS 的 format 指令可以在執行邏輯格式化時一併掃描磁碟，如果發現到一個壞損磁區，會在相對映的 FAT 紀錄中寫入一個特定值，以告知程式不要使用這個磁區

84

壞損區塊

- 若是在使用時所造成的損毀，則需要以人工方式執行一個特定程式，如 chkdsk，以搜尋磁碟內壞損的磁區、並將它們記錄下來禁止使用
- SCSI磁碟控制器於低階格式化時產生壞損磁區串列，並設置OS也無法查知的額外備援磁區，控制器會使用備援的磁區來替代壞損的磁區，這樣做法稱為磁區備援(sector sparing)或磁區轉交(sector forwarding)

85

壞損區塊

- SCSI磁區壞損處理如下：
  - OS嘗試讀取邏輯區塊56
  - 磁碟控制器先進行ECC檢查，發現該磁區壞損，並告訴OS
  - 等下一次系統重新啟動時，系統執行一個特殊命令告訴SCSI控制器以額外備援磁區取代該損毀磁區
  - 當在次要求讀取邏輯區塊56時，控制器會將此要求轉換成取代後的磁區位址

86

壞損區塊

- 處理方式：
  1. 鎖住
  2. 磁區備份(sector sparing)或磁區轉換
     - 作業系統要嚐試讀取邏輯區段N。
     - 控制器計算ECC值，並且發現此磁區是壞的。就將此發現報告作業系統。
     - 作業系統在下一次重新啟動時，就執行一特殊指令以告訴SCSI控制器，將壞的磁區用空白的磁區換掉。
     - 之後，系統要求邏輯區段N時，要求被控制器轉換成取代的磁區位址。

87

壞損區塊

- 為避免破壞OS排程演算法，每個磁柱中都提供一些備援磁區，甚至是備援磁柱，這樣在取代壞損磁區時，儘可能使用位於相同磁柱的備援磁區
- 另一方法為磁區順延，將壞損的磁區空下來，其他磁區延後一個位置(199→200 …)，但是壞損磁區並無法自動修復，也導致磁區中儲存資料的遺失，就算可修復壞損磁區的資料，也需要藉由人工處理

88

RAID 結構

- 磁碟裝置體積越來越小，也越來越便宜，在一電腦系統上加入多顆磁碟也越來越普遍，因此有一種新型的I/O裝置稱為磁碟陣列 (RAID Redundant Arrays of Independent Disks)
- 將數個磁碟串接在一起，並將平行處理的觀念應用於 I/O 之上，改善磁碟效能

89

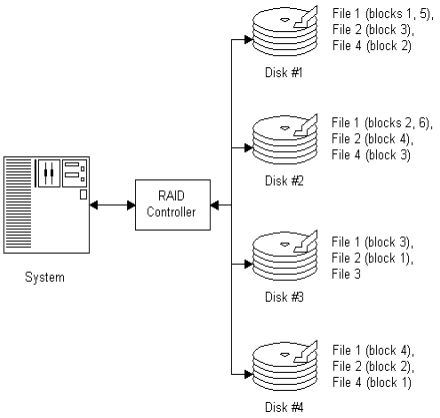
RAID 結構

- 經由重複改進可靠度
  - N台磁碟機的磁碟機組某一台磁碟機失效的機率比一台特定磁碟機失效的機率還高。假設單一台磁碟機的平均失效時間(mean time to failure)是100,000小時，則 100台磁碟機的磁碟陣列中某一台磁碟機的平均失效時間是 100,000/100=1,000小時 (或41.66天)!
  - 一台磁碟機失效不會連結到另一台磁碟機。那麼如果單一台磁碟機的平均失效時間是100,000小時，而且平均修復時間是 10小時，則一個鏡射系統的平均資料遺失時問 (mean time to data loss)是 $100{,}000^2/(2*10)=500*10^6$小時，亦即 57,000年!

90

RAID 結構

- 藉由並行改善性能
  - 考慮多磁碟機並行存取的優點。使用磁碟鏡射時，讀取要求能夠被處理的速率可以加倍，因為讀取要求可以被送往任何一台磁碟機 。每一次讀取的傳輸速率和單一磁碟系統相同，但是單位時間的讀取數目加倍。
  - 有了多台磁碟機時，也可以藉由在多台磁碟間交錯儲存資料來改進傳輸速率。資料交插(data striping)最簡單格式是將每一位元組分成位元分散在許多台磁碟機上:這種交插稱為位元層次交插
1. 經由負載平衡增加多次小規模存取(亦即頁存取)的生產量。
2. 降低大量資料存取的反應時間。



91

可靠度和效能

- 為解決可靠度的問題，可以在磁碟中多儲存一些額外的資訊，甚至備份整個磁碟的資料，在磁碟發生故障時，就可以利用這些額外的資訊來重建遺失的資料，即使磁碟發生故障，資料也不會遺失
- 基於提高可靠度(MTBF、MTTF)最簡單的方法就是對每顆磁碟都再附加一顆額外的磁碟，此技術稱為鏡像(Mirroring or Shadowing)

92

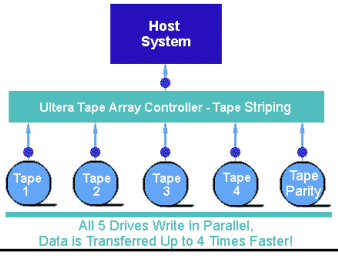
可靠度和效能

- 鏡像是指一個邏輯磁碟是由一對磁碟所組成，每一個寫入的動作都會將資料同時寫入這兩顆磁碟之中，若有一顆發生故障，資料可由另一顆鏡像的磁碟取出，所以只有同時兩顆發生故障，資料才會遺失，因此鏡像磁碟系統比單顆磁碟系統提供了更佳的可靠度➔RAID level 1
- 鏡像系統也可平行讀取多顆磁碟，因為鏡像是由一對磁碟組成，因此可將讀取要求送至任何一顆，通常鏡像磁碟系統可將磁碟讀取速度提高兩倍以上，即每單位時間可同時讀取兩倍資料

93

RAID 結構

- RAID層次
  - 鏡像(mirroring)提供高的可靠度，但是卻很昂貴。交錯提供高的資料傳輸率，但是卻不能增進可靠度。數種藉由使用磁碟機交錯結合"同位"位元的低價格重複性技巧被提出。這些技巧有不同的價格-性能調整，而且被歸納成為RAID層次的數種階層。



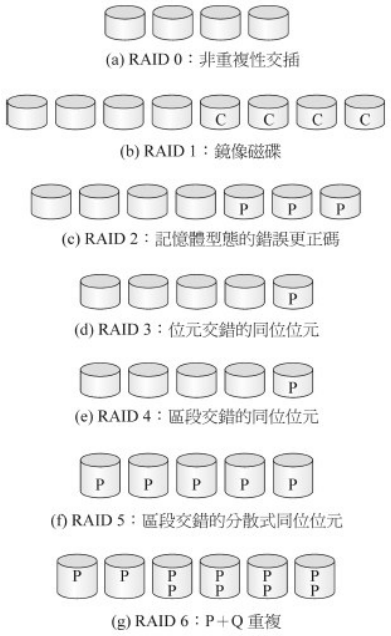


圖 12.11 RAID 層次

可靠度和效能

- 將資料的每個位元組的各個位元分散配置於不同的磁碟上，如此在讀取資料時可以同時跨過多個磁碟，分別讀取某個位元組的不同位元而提高效率，稱為資料跨分(data striping)，跨分單位可以是位元階層跨分，其他如位元組、檔案區塊、與磁碟上的磁區也可使用跨分方法
- 資料跨分主要有 2 個目的
  - 藉由同時存取多顆磁碟中的跨分資料，以增加產量
  - 降低存取大量資料的反應時間

95

RAID 階層 (94tku)

- RAID就像一個大磁碟，將資料分散存在多顆磁碟上，使系統能平行存取各個不同磁碟上的資料
- RAID 0：資料以區塊為單位跨分於多顆磁碟
- RAID 1：複製所有的磁碟，即磁碟鏡像
- RAID 2：以位元為跨分單位，加入多個漢明碼的檢查位元
- RAID 3：對區塊只產生單一個同位檢查位元，並將檢查的結果寫入專門儲存同位檢查位元的磁碟中

96

- RAID 0:在區段層次交錯的磁碟機陣列，但是沒有任何重複的資料(例如鏡像或同位位元)。
- RAID l: 指磁碟的鏡像(mirroring)。
- RAID 2: 也稱為記憶體型式的錯誤更正碼組織(memory style error-correction-code, ECC, organization)。記憶體系統早已經使用同位位元製作錯誤偵錯。記憶體系統中的每一個位元組都有一個和它相關的同位位元，其中記錄著此位元組中位元被設定為1的個數是偶同位(Parity=0)或奇同位(Parity=1)。

97

RAID 階層

- RAID 4：與RAID 3幾乎相同，不過是以區塊為跨分的單位
- RAID 5：不同於RAID 4的是RAID 5將資料和同位檢查位元資料散佈於所有的磁碟上
- RAID 6：與RAID 5類似，RAID 6使用Reed-Solomon編碼方式
- RAID 0+1 / 1+0：RAID 0和RAID 1的組合，既可以得到RAID 0的效率，也可獲得 RAID 1的可靠度

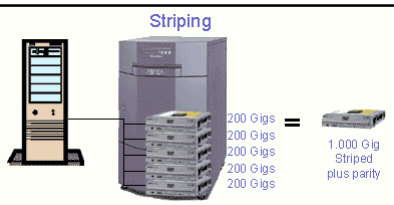
98

- RAID 3: 位元交錯同位位元組織(bit-interleaved parity organization)由RAID層次2精進而來(不像記憶體系統)，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，這和記憶體系統不同。它的觀念如下。如果磁區當中有一個損壞，則可以正確地知道是那一個磁區，而對於磁區中的每一位 元，可以藉由其它磁碟機之磁區中的相關位元所計算出的同位位元以猜出 它是1或0。如果剩餘位元的同位位元和儲存的同位位元相等，則遺失的位元是0:否則就是1。
- RAID 4: 區塊交錯同位位元組織(block-

99

- RAID 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。對於某一區段而言，其中一台儲存同位位元，而其餘的儲存資料。
- RAID 6(也稱為P+Q重複技巧)和RAID層次5極相似，但是儲存額外重複性資料以保護多台磁碟機失效。RAID6並沒有使用同位位元，而是使用類似於Reed-Solomon碼的錯誤更正碼。每4個位元的資料就有2位元的額外資料被儲存 (這和層次 5中的 1個同位位元不相同)，而此系統可以忍受兩台磁碟機壞掉。

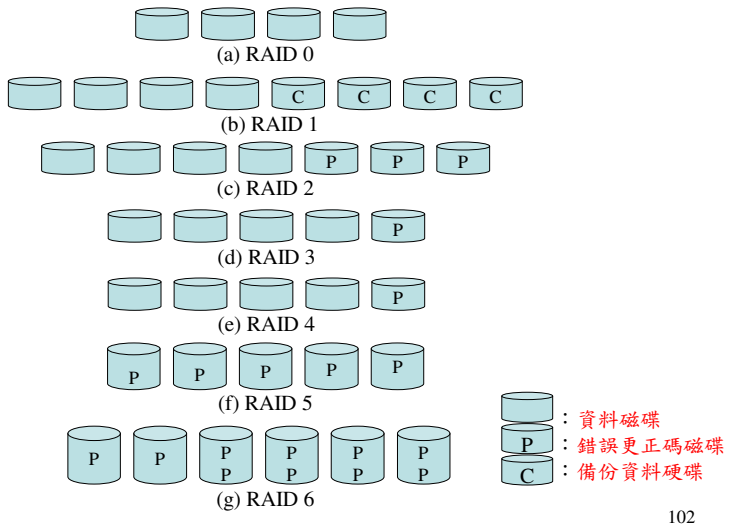
100



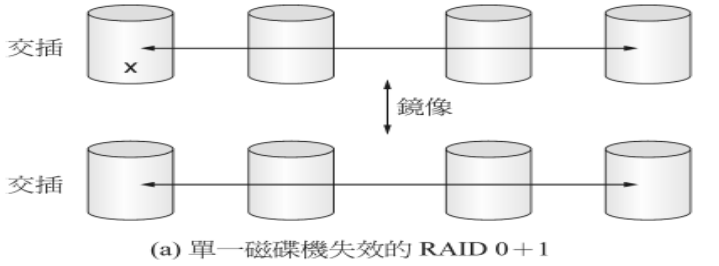
- RAID 0+l: RAID層次0和1的結合。RAID 0提供了性能，而RAID1提供可靠度。它提供了比RAID 5更好的性能。性能和可靠度都很重要的環境非常普遍。這種情況會使需要儲存的磁碟機數目加倍，所以也很昂貴。在 RAID 0+1中，一組磁碟機被交錯分散開，然後交錯的磁碟機再被鏡射到另一組 (對應的交錯)。另一種RAID的選項在商業上已經逐漸可以取得的是RAID 1+0，在這種組態中磁碟機是成組的鏡像，而產生的鏡像組再被交錯儲存。這種 RAID有理論優於RAID 0+l 。

101

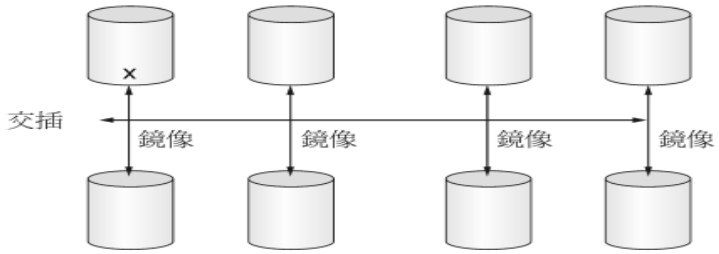
RAID 階層



102



(a) 單一磁碟機失效的 RAID 0 + 1

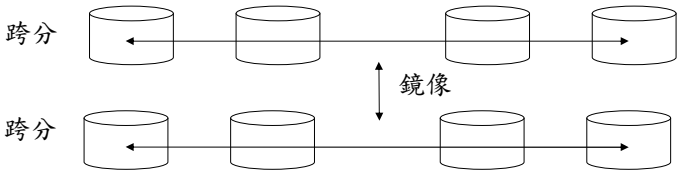


(b) 單一磁碟機失效的 RAID 1 + 0

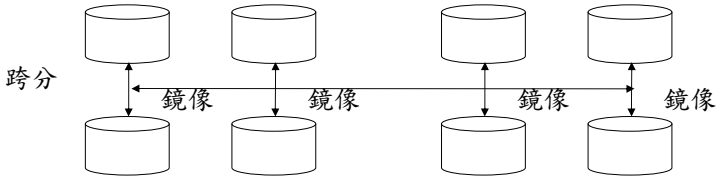
圖 12.12 RAID 0 + 1 和 1 + 0

103

RAID 0+1 和 1+0



(a) RAID 0 + 1



(b) RAID 1 + 0

104

選定 RAID 階層

- RAID 0 要求在高效率不在乎資料遺失
- RAID 1 需要高可靠度與快速復原
- RAID 0+1 和 RAID 1+0 可靠度和效率皆重要的環境
- RAID 5 適合用於儲存大量資料的環境下
- RAID 6 提供比 RAID 5 更佳的可靠度

105

選定 RAID 階層

- 選擇一個 RAID 層次
  - 重建對於RAID 1最簡單，因為資料可以從另一台磁碟機拷貝過來;對於其它層次RAID而言，需要存取陣列中所有其它磁碟機以重建毀損的磁碟機資料。如果需要連續提供資料時，RAID系統的重建性能就是一項重要的因素，這在高性能或是交談式的資料庫系統正是如此。另外，重建性能還會影響到平均失效時間。對大型磁碟機組的RAID 5重建時間可能要幾個小時。

106

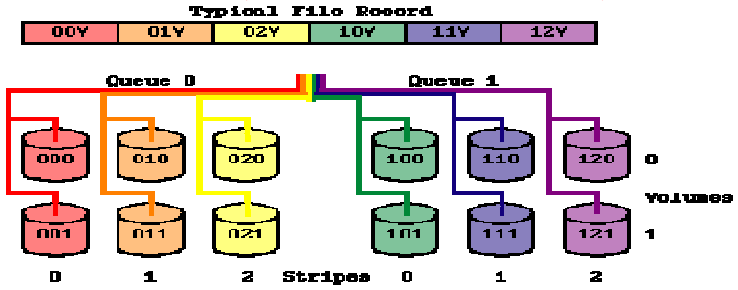
選定 RAID 階層

- RAID 0被用在高性能應用上，但是資料損失不是很重要時。RAID層次1對於需要有快速復建的高可靠度應用上很普遍。RAID 0+1和1+0被用在性能和可靠度都很重要的場合，例如小型的資料庫。
- 因為RAID 1在空間上的額外負擔，RAID 5通常最適於儲存大量的資料。層次6目前沒有被許多RAID的製作所支援，但是它應該提供比層次5更好的可靠度。

107

RAID延伸

- RAID觀念可以被推廣到其它的儲存裝置，包括陣列式磁帶，甚至到無線系統的資料廣播。當RAID結構被應用到陣列式磁帶時，陣列式磁帶中的一個磁帶毀損，依然可以用來恢復資料。當RAID應用到廣播式的資料，一個區段的資料被分成短的單元，並且伴隨著同位單元被廣播出去;如果其中一個單元因為某些原因沒有被收到時，它可以從其它單元重建。磁帶機的機器手包含許多台磁帶機，並且將資料散佈到所有磁帶交錯儲存以增加產量和降低備份時間。



108

RAID的問題

- 對作業系統和它的使用者RAID不常保證資料是可以使用。檔案的一個指標可能是錯誤的或檔案結構的指標可能是錯誤的。不完全的寫入如果不適當地恢復，會造成錯誤百出的資料。
- 一些其它的行程也可以偶然在檔案系統的結構上寫入。RAID保護資料的錯誤，但不是其它硬體和軟體的錯誤。軟體和硬體的錯誤就多少有潛在危險在系統的資料上。

109

儲存設備附加方式

- 電腦上附加的儲存設備，其存取資料的方式主要有 3 種：
  - 從本地主機上經由 I/O 埠存取，這種設備稱為主機附加儲存設備(Host-Attached Storage)
  - 經由遠端主機的分散式檔案系統來存取，這種設備稱為網路附加儲存設備(Network-Attached Storage)
  - 儲存區域網路，是綜合以上兩種方式，加上獨立的儲存設備協定來建構有效率的大型網路儲存系統(Storage Area Network)。

110

主機附加儲存設備(HAS)

- 一般電腦所用的儲存設備都是經由 I/O 埠作存取的主機附加儲存設備，如 IDE(ATA、ATA66、ATA100)，最多提供每個I/O匯流排2個裝置
- 高階工作站和伺服器一般則使用更複雜的架構，如 SCSI 或光纖通道

111

主機附加儲存設備(HAS)

- SCSI實際傳輸媒介由50-68條導線的電纜，最多可連接16個裝置，包括一個控制卡與15個裝置，SCSI協定能對每個裝置設備定址，可設定1-15個邏輯單位，如直接控制8個硬碟的RAID
- 光纖通道是一個利用光纖或同軸電纜來運作的高速串列架構，其有兩種形式，一種是大型可交換的光纖架構，有24位元的位址空間，因大量定址和可交換的通訊特性，使得各種主機與儲存裝置都可以加入光纖架構，在I/O通訊上具有很大的彈性，此為SAN的基礎。另一種為仲裁迴路的形式，只能定只到126個裝置

112

主機連結儲存裝置(host-attached storage)

- 主機連結的儲存裝置是經由區域性I/O埠所存取的儲存裝置。典型的桌上型PC使用稱為IDE或ATA的I/O匯流排架構。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)。
  - SCSI是一種匯流排架構。它在硬體上的媒體通常是一條有許多接線(一般是50或68)的帶狀電線。SCSI協定最多支援匯流排上有16個裝置。通常這是由主標上的一片控制卡(SCSI initiator)，和多達15個儲存裝置(SCSI target)所組成。SCSI硬碟是一個典型的SCSI目標(SCSI target)，但是協定在每一個SCSI目標上提供了位址到8個邏輯單元(logic unit)的能力。

113

主機連結儲存裝置(host-attached storage)

- FC是一種高速的串列架構。這個架構可以在光纖上操作，或是在4-導體的銅纜。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。這種方法被視為未來的主導者，而且是儲存型區域網路(storage-area network, SANs)的基礎。

114

網路附加儲存設備(NAS)

- NAS 是特殊目的的網路儲存系統，藉由資料網路作遠端的資料存取
- NAS 在可將資料集中管理，讓使用者透過區域網路進行儲存空間的共享，這種直接連接在區域網路上高效能的儲存設備，可以支援多種檔案協定，讓客戶端在不同的平台環境上共享檔案(ftp://nas.im.takming.edu.tw、ftp://asp.im.takming.edu.tw)

115

網路連結儲存裝置

- 網路的連結儲存裝置，是經由資料網路從遠端存取的特殊目的儲存系統。客戶經遠端程序呼叫的介面(例如UNIX系統的NFS，或Windows 的CIFS)存取網路連結儲存裝置 (network-attached storage, NAS)；遠端程序呼叫(RPC)在IP網路上經由TCP/UDP執行。
- 網路連結的儲存單元通常是用遠端程序呼叫介面的軟體製作成RAID陣列。把NAS視為只是另一種儲存體存取協定。

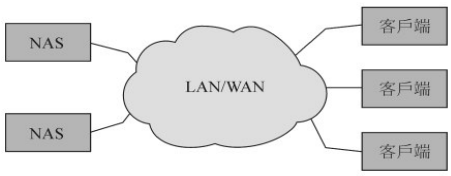


圖 12.2 網路連結儲存體

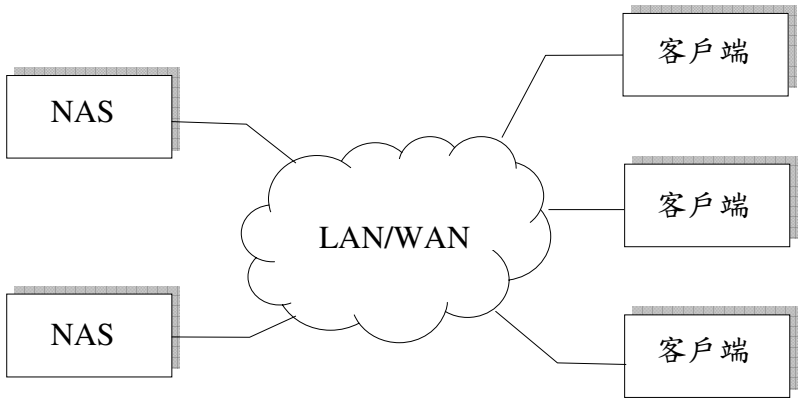
116

網路附加儲存設備(NAS)

- NAS在 IP 網路協定上是使用遠端程式呼叫，而不是以使用裝置驅動程式的方式來存取儲存設備
- NAS 系統的缺點是，I/O 設備的操作會消耗掉資料網路的頻寬，而造成網路通訊的延遲，這問題在大量客戶端與伺服器的環境下會相當的嚴重，因伺服器和客戶端的通訊，會與伺服器和設備間通訊互相競爭網路頻寬

117

網路附加儲存設備(NAS)



118

儲存區域網路(SAN)

- SAN 是介於伺服器與設備間的私有網路
- 使用設備協定而不是網路傳輸協定，與連接伺服器及客戶端的區域網路或廣域網路分離
- 在同一個SAN上可以附加多種主機和多樣化的的設備陣列，儲存設備還可以動態地配置給主機，如果主機的磁碟空間逐漸減少，SAN能夠設定給予這個主機更多的磁碟裝置，以獲得更多的磁碟空間

119

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 。SAN的功能在於它的彈性。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。

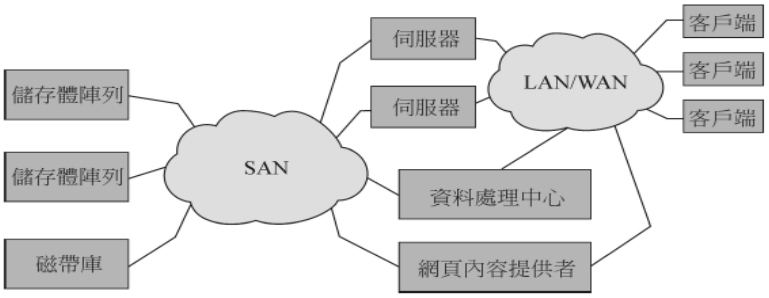


圖 12.3 儲存體區域網路

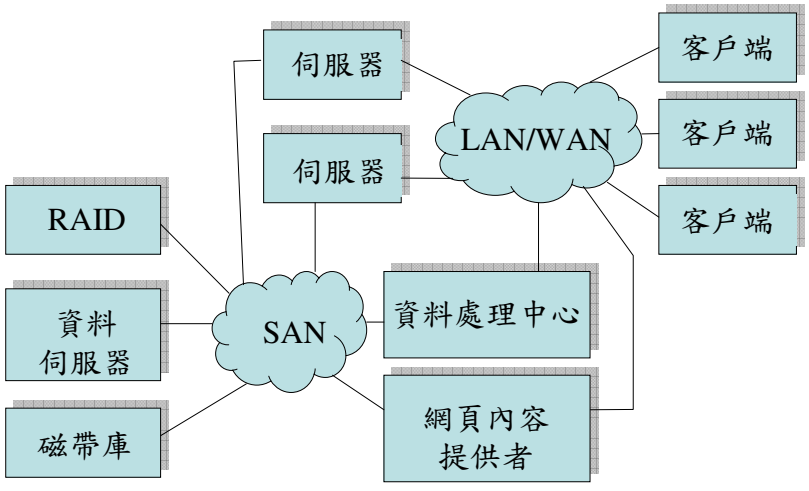
120

儲存區域網路(SAN)

- SAN元件仍未達到標準化與跨平台使用的情形為其缺點
- 許多 SAN 系統是架構在光纖通道迴路或是光纖通道交換式網路上，必須將以光纖通道互相連結的SAN系統架構在 IP 網路基礎的設施上(Gigabit Eithernet) ，或是另一種以一個特殊目的的SAN架構(Infiniband)對於伺服器和設備單元提供支援硬體及軟體的高速連接網路

121

儲存區域網路



122

穩定儲存體

- 資料存於穩定的儲存體不會有任何損毀的情形，所以在實作穩定儲存設備時必須複製一些必要的資訊到其他儲存裝置中，且必須保證在更新期間，不論發生任何錯誤，都不會影響其他複製的資料，在回復損壞的檔案時，即使有其他錯誤的發生，也能保證所有複製資料都仍保有正確的內容

123

穩定儲存體

- 一般磁碟寫入會有可能結果：
  - 完全成功
  - 部分失敗：資料傳輸發生錯誤，只有部分磁區有寫入新值，發生失敗時所寫入的磁區資料可能已經損毀(rollback)
  - 完全失敗：磁碟寫入前已經發生錯誤，所以之前在磁碟內的資料，仍完整無缺

124

穩定儲存體

- 穩定儲存設備要求寫入一個區塊時，若有錯誤發生，系統必須能夠偵測並執行一個復原的程序，將這個區塊復原到之前的狀態
- 為達此目的，必須對每個邏輯區塊維護2個實體區塊，其操作如下：
  - 寫入資訊到第1個實體區塊
  - 當第1個實體區塊已經完全成功寫入後，再將相同的資訊寫入第2個實體區塊
  - 只有當兩個實體區塊完全成功寫入後，才宣佈操作完成

125

第三儲存裝置

- 第三儲存裝置最大的特性是價格低
- 第三儲存裝置大多是用可抽換的媒介構成的，如：
  - 可抽換的磁碟片
  - 磁光碟片(MO)
  - 光碟片：對雷射光反映出明暗變化的特殊材質
  - 相位改變光碟：凍成結晶或是非結晶狀態的材料
  - 唯讀光碟
  - 磁帶(DAT)

126

作業系統議題

- 作業系統的兩項主要工作
  - 管理實際裝置
  - 提供虛擬機器的抽象層
- 應用程式介面
  - 作業系統對於可抽換磁碟的處理，幾乎都是將它看待為固定式的磁碟來處理
  - 作業系統通常以裸 I/O 的方式來使用磁帶，磁帶的應用程式不是開啟磁帶中的一個檔案，而是將磁帶當成一個裸磁碟一樣的裸裝置來開啟整個磁帶

127

作業系統議題

- 檔案命名
  - 在個人電腦上檔案名稱是跟隨著路徑名稱和裝置字元所組成
  - 在 UNIX 系統上的檔案名稱是以掛載表讓作業系統發現檔案是安置在那一個裝置中
- 階層式的儲存管理
  - 主記憶體、輔助記憶體與第三儲存裝置形成了階層式的儲存系統

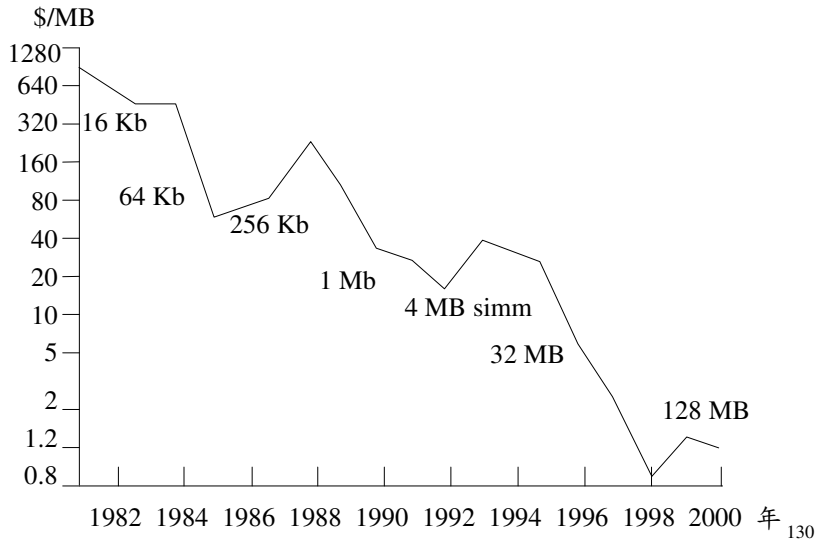
128

效能議題

- 速度
  - 承受頻寬是平均資料傳輸率，也就是傳輸位元組的個數除以傳輸時間
  - 有效頻寬是指裝置所提供的資料傳輸速率，速率的計算是整個 I/O 時間的平均值
- 可靠度
  - 固定式磁碟比可抽換式磁碟與磁帶可靠
  - 光學式的儲存裝置比磁碟與磁帶來的可靠
- 成本
  - 價位是決定第三儲存裝置效能的重要因素

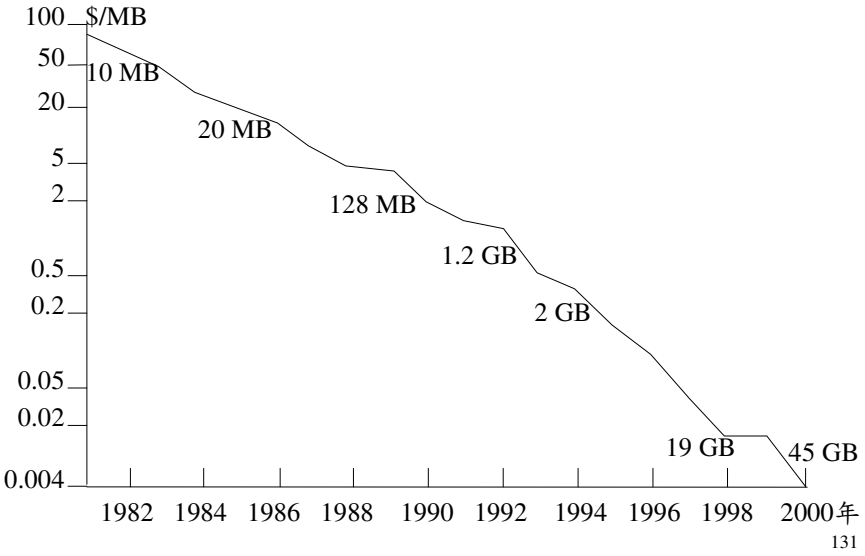
129

DRAM 價格



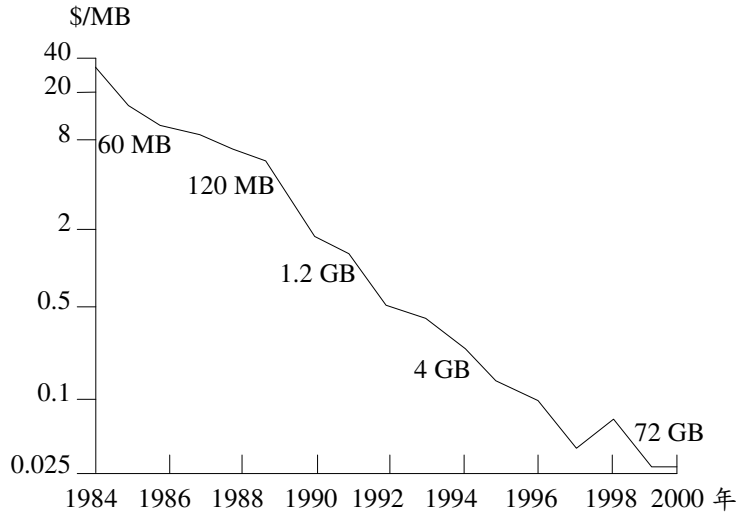
130

硬碟價格



131

磁帶驅動裝置價格



132

摘要

- 大部分電腦的主要輔助儲存裝置是磁碟
- 磁碟排程是降低磁碟的平均搜尋時間的方法
  - SSTF
  - SCAN 與 C-SCAN
  - LOOK 與 C-LOOK
- 低階格式化對每個磁區填入特別的資料結構
  - 標頭
  - 資料區(通常是 512 個位元組)
  - 結尾
- RAID 提供良好的效率和可靠度

133

摘要

- 穩定儲存體保證資料不會有任何的損毀
- 第三儲存體是由可抽換的媒介所構成的
  - 軟碟
  - 光碟
  - 磁帶
- 作業系統議題
- 效能議題

134