



檔案系統

大綱

- 檔案系統的架構與功能
- 檔案與目錄
- 使用者與檔案間的關係
- 檔案系統與裝置組織的方法
- 基本的檔案系統
- 檔案系統的保護機制
- 本章重點回顧

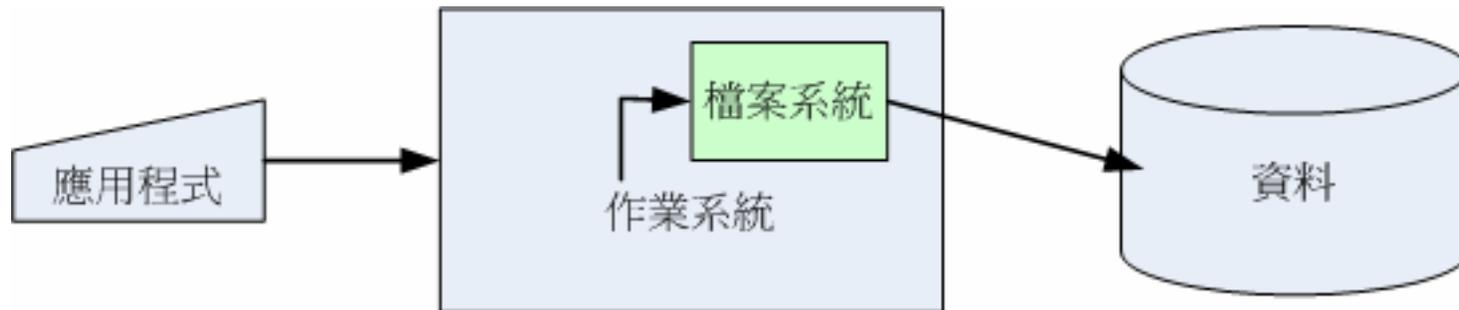
前言

■ 檔案系統的功能：

- 負責管理作業系統中的資料與存放這些資料
- 有效率的將這些資源進行管理並取得使用
- 舉例來說：
 - DVD出租店的DVD影帶管理方式
 - 分門別類或者是利用索引字查詢
 - 這種管理的方是就是檔案系統需要的！

檔案系統的架構

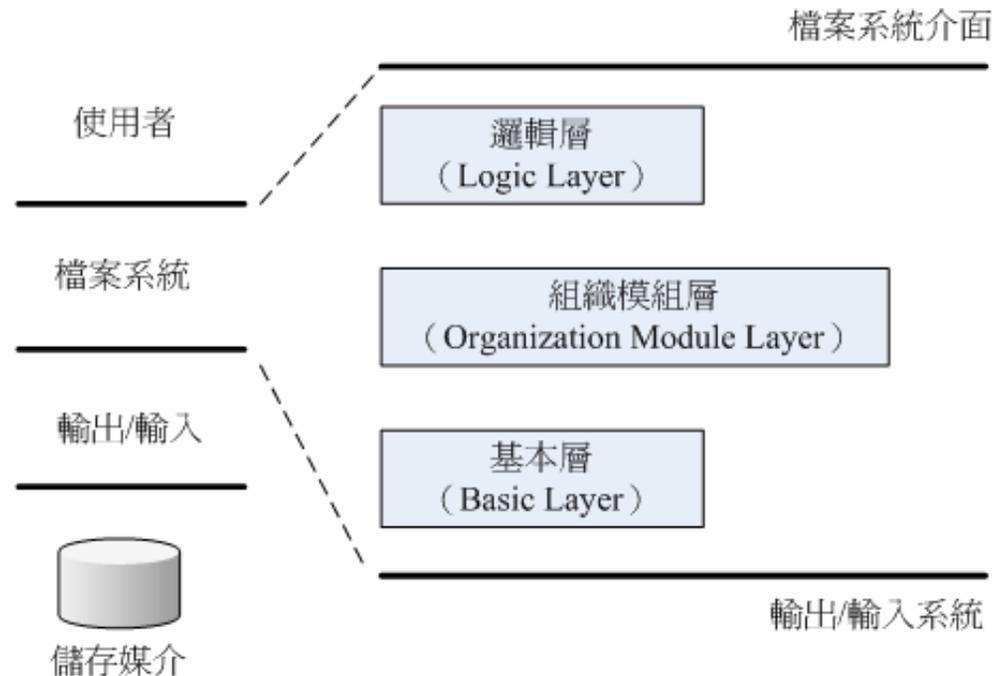
- 使用者透過應用程式向作業系統提出檔案需求
 - 使用者不需要知道檔案系統為何，
 - 因為是作業系統管理的功能
- 作業系統將工作提交檔案系統
- 檔案系統會到儲存媒介上面進行檔案的存取



檔案系統的架構與功能

■ 檔案系統階層關係

- 依據主要功能分成不同的部分
- 採用階層式的架構
- 每個特定層級只能夠被授權去呼叫相同或更低層級的服務



檔案系統階層關係

■ 檔案系統的三個主要階層關係

□ 檔案系統介面，亦稱為邏輯層 (Logic layer)

- 利用特定的檔案資料結構或目錄結構來記錄邏輯檔案
- 檔案可用兩種格式名稱來辨識『符號名稱』『開啟檔案的識別碼』

□ 組織模組層 (Organization Module layer)

- 使用符號檔案名稱來取得處理檔案所需的描述性資料
- 例如將檔案的邏輯位置轉到儲存媒體的實體位置
- 管理檔案的儲存空間

□ 基本層 (Basic layer)

- 處理與儲存裝置有關的事宜，如開啟、關閉某個裝置的位置
- 使用者不需要知道檔案放在那個媒體上！

檔案與目錄

■ 檔案系統的問題：

- 你如何取用儲存媒體中的資料？
- 你需不需要知道儲存媒體使用的裝置是啥？

■ 解決方案：

- 利用目錄樹：目錄與檔案來讀取資料
- 利用檔案系統：作業系統的子系統之一來控管所有的儲存媒體裝置，所以不需要知道媒體，只需要知道目錄/檔名即可！

檔案與目錄

■ 檔案/目錄所記錄的資訊

□ 檔案：

- 記錄實際的檔案資料，包括原始碼、binary program等等

□ 目錄：

- 本身也是個檔案
- 目錄內容記錄的其實是『檔名』，亦即目錄記錄其他檔案的資訊而已。

目錄的架構—無階層式

- 檔案清單中，每個檔案都是位於相同的地位與等級
 - 因為沒有分門別類，在執行方面搜尋不利

檔案一

檔案二

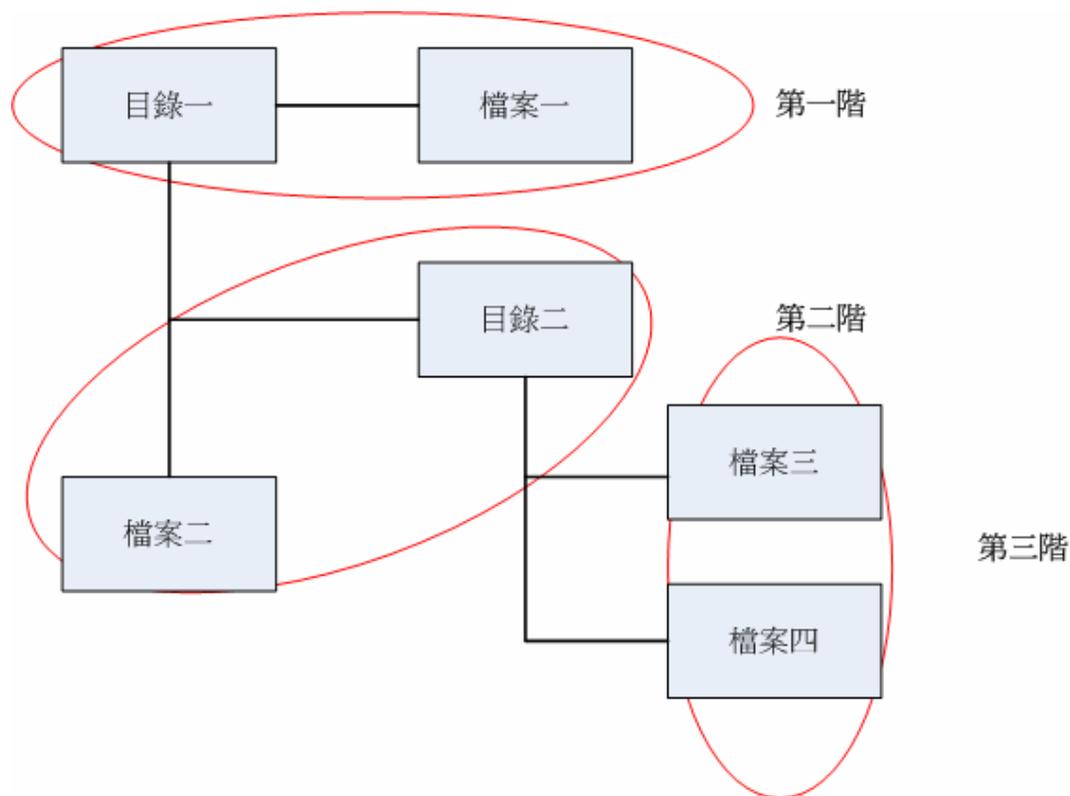
檔案三

檔案四

目錄的架構—階層式

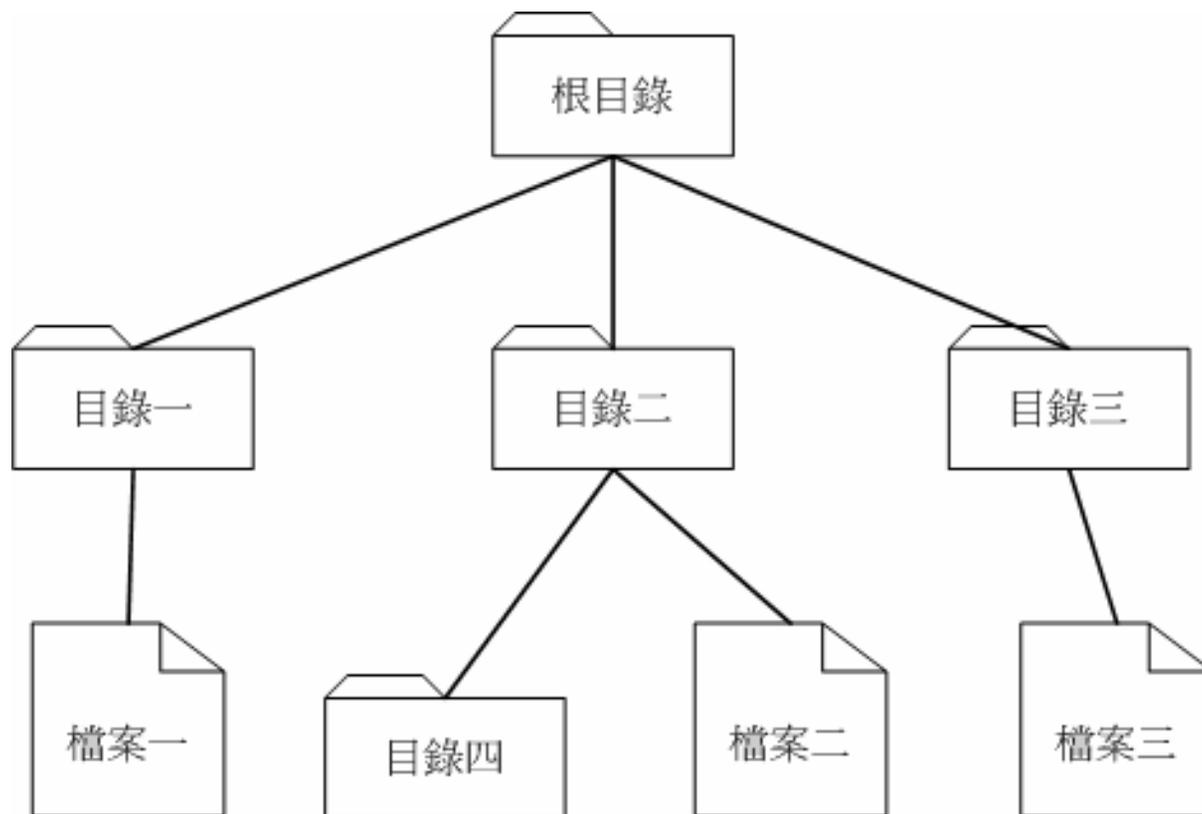
■ 階層式架構的目錄格式

- 每個檔案都會指向比自己更低階的目錄
- 依次分析各目錄
- Linux最上層目錄
 - 根目錄 (/)
- 優點是：
 - 方便搜尋
 - 方便維護



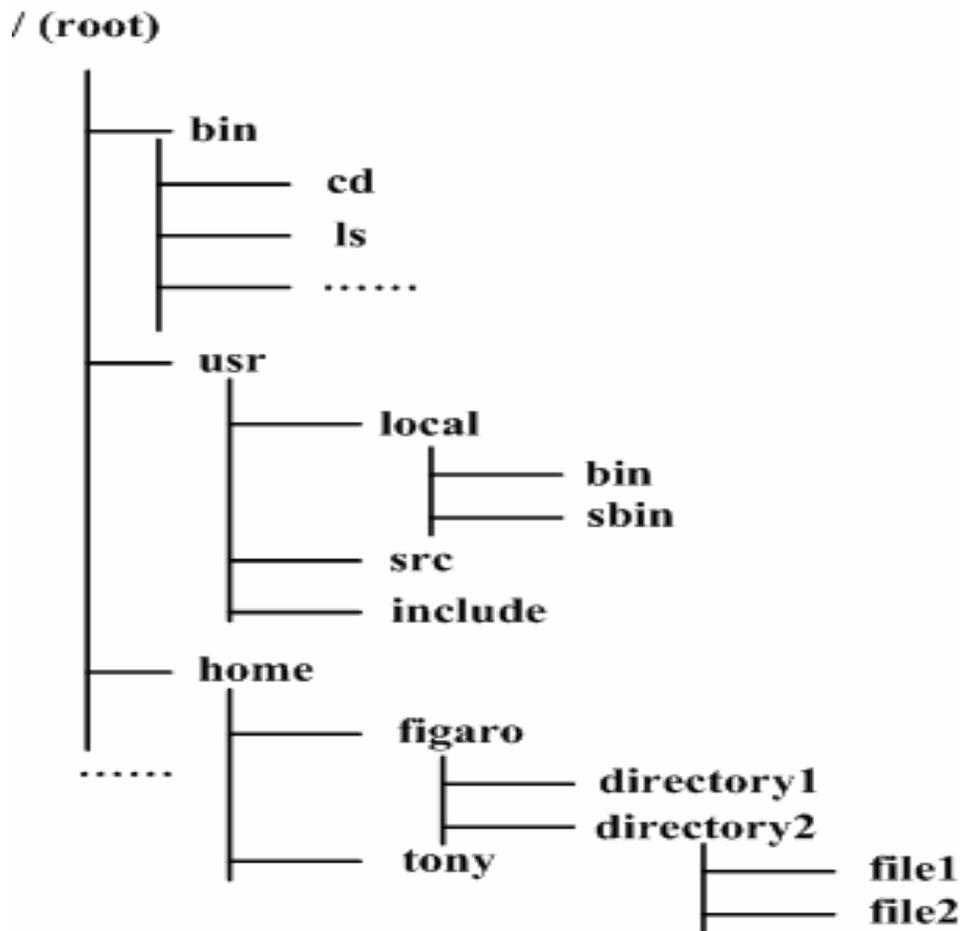
Linux樹狀架構目錄

- 根目錄、父目錄、葉、子目錄示意：



Linux樹狀目錄示意

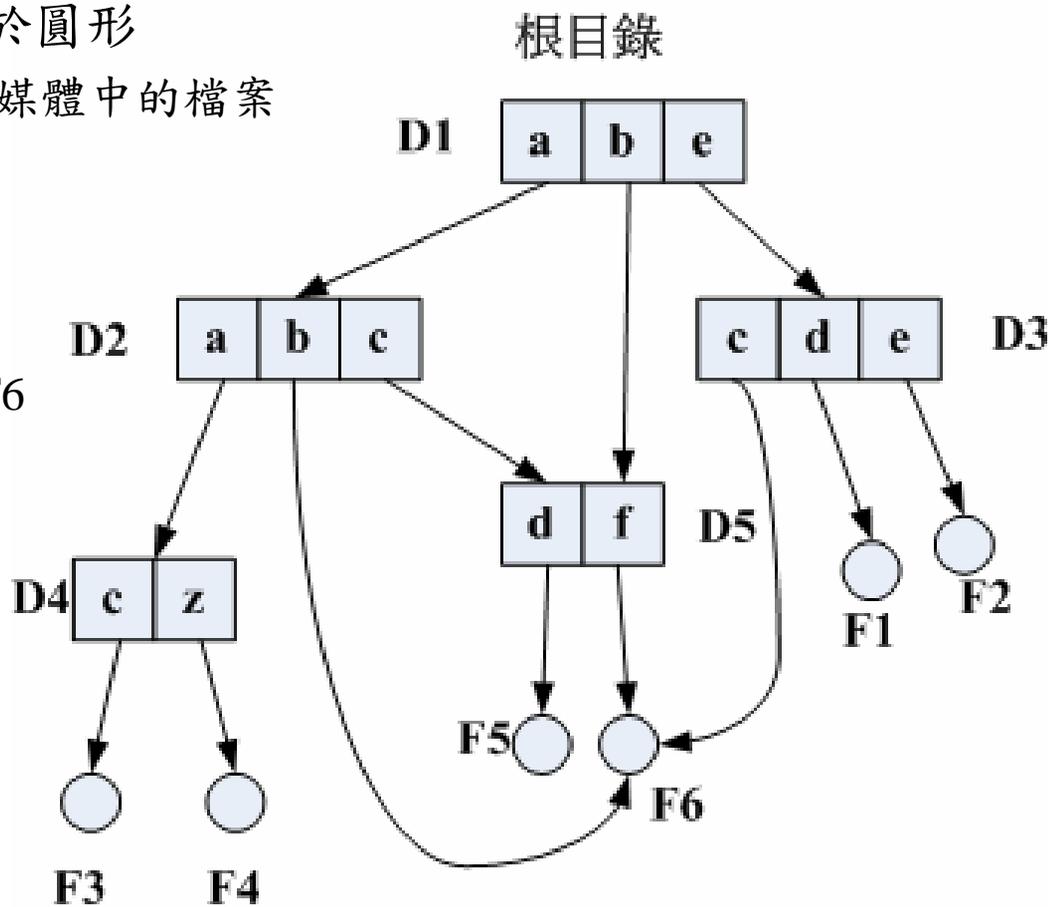
- 最頂層為 /
 - 遵循 FHS 規範
 - /
 - /var
 - /usr



檔案與上層目錄(父目錄)關係

■ 有向的無循環圖形 (Directed Acyclic Graph, DAG)

- 各檔案的實際資料放置於圓形
 - F1, F2...都是放於儲存媒體中的檔案
- D1, D2...都是目錄
- a, b, c.. 都是檔名而已
- D2, D3, D5 的 b,c, f
 - 其實都是同一個檔案 F6
- 若要刪除 F6
 - b, c, f 都需刪除!
- 可能造成無窮迴圈!



檔案與上層目錄(父目錄)關係

■ 符號連結法

□ 實際上 b, c 為實際的檔案存在

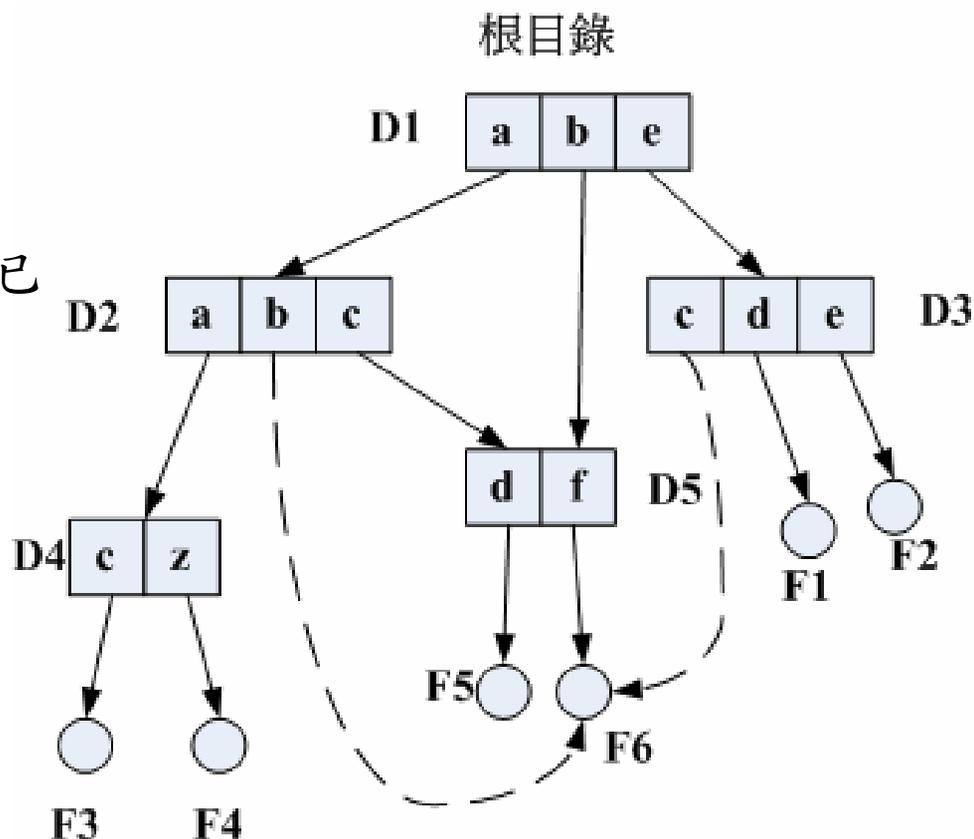
□ 只是該檔案『指向』

■ D5→f→F6

□ 事實上 b, c, f 為獨立的檔案

□ F6僅有 f 這個檔名的對應而已

□ b, c 稱為『連結檔』



Linux連結的實作

■ 連結的指令：ln

□ 實體連結 (Hard link)：

■ ln 來源 目標

■ 限制：不能跨檔案系統與不能連結目錄

■ 就是 DAG 類型

□ 符號連結 (Symbolic link)：

■ ln -s 來源 目標

Linux連結的實作—重要練習

- 複製 /etc/hosts 到你的家目錄
- 製作實體連結成為 hosts.hard
- 製作符號連結成為 hosts.sym
- 查閱每個檔案的大小與內容
- 刪除 hosts
- 是否能夠開啟 hosts.hard 與 hosts.sym ? why ?

使用者與檔案間的關係

- 檔案的分類依據：
 - 名稱、類別、邏輯組織以及屬性 (Attributes)

- 不同檔案系統對於檔名的限制：
 - MS-DOS：只允許八個字元且不可有特殊字元
 - 早期的Unix：只允許 14 個字元
 - Windows：最大可達 255 字元且可使用特殊字元
 - 現代 Linux：可達255字元以上，且大小寫不同

使用者與檔案間的關係

■ 副檔名的功能：

- 用來辨識該檔案的類別
- 僅是用來『識別』而已
 - Windows→執行檔：.exe, .com, .bat...
- Linux可使用 file 這個指令來識別
 - file /etc/hosts

常見的副檔名

副檔名	功能描述
.doc、.ppt、.xls、.vsd	Microsoft Office常用副檔名，分別為Word文件、PowerPoint簡報檔、Excel活頁簿、Visio繪圖檔。
.sh、.csh	Linux作業系統下，Shell Script檔案型態。
.zip、.rar、.tar、.gz、.bz2	常用壓縮檔案型態。
.avi、.mpeg、.wmv	常見影音檔案型態。
.bmp、.jpg、.gif	常見圖片檔案型態。
.c、.cpp、.asm、.fort	常見原始程式碼檔案型態。

檔案的資訊存放處

■ 檔案的主要資訊：

- 檔名：放置於目錄中
- 內容：放置於邏輯記錄區，或記錄區塊(block)
 - 固定大小的邏輯記錄(常用)
 - 可變動大小的邏輯記錄
- 重要屬性、參數、權限資訊：inode區塊中
 - 如所有權、檔案大小等

固定長度邏輯紀錄



可變動長度邏輯紀錄



檔案的重要資訊

- 檔案屬性的重要資訊主要有：
 - 檔案所有權(ownership)
 - 檔案大小
 - 檔案的使用：
 - 建立時間、存取時間與修改時間
 - 保護：
 - 存取的對象有誰，如r,w,x權限
 - 位置：
 - 實際資料存放區域的記載

Linux檔案資訊的觀察

■ 透過 ls ， 或者是 ll 指令



■ 練習：

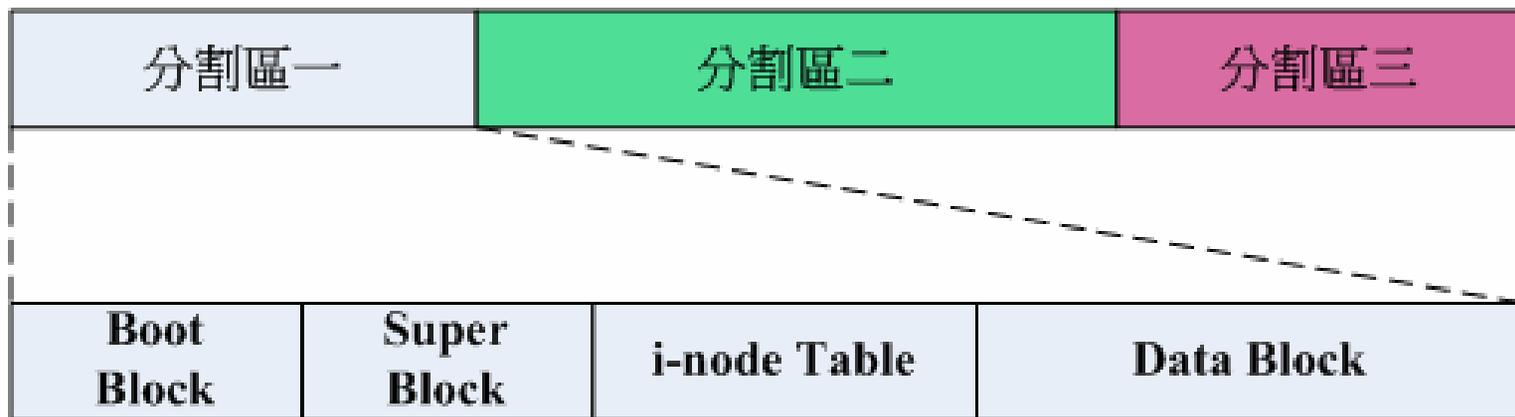
- 請察看 `/etc/hosts` 的相關資訊，並嘗試說明

檔案系統與裝置組織的方法

- 電腦系統中儲存媒體裝置的最小資料單位：
 - 實體區塊（Physical Block，或稱之為實體紀錄）
 - 硬碟可透過分割(Partition)做成獨立的分割槽
 - 分割槽就是檔案系統建立的單位了！
 - 建立檔案系統：就是格式化！格式化的結果會產生：
 - Block：就是放置實際資料內容的區塊，亦即邏輯區塊
 - inode：放置檔案屬性的位置

檔案系統與裝置組織的方法

- 分割的結果且格式化後的檔案系統配置示意圖
 - super block：記錄整個檔案系統的重要資訊歸納
 - inode：記錄檔案屬性、權限的地方
 - Data block：記錄實際檔案內容



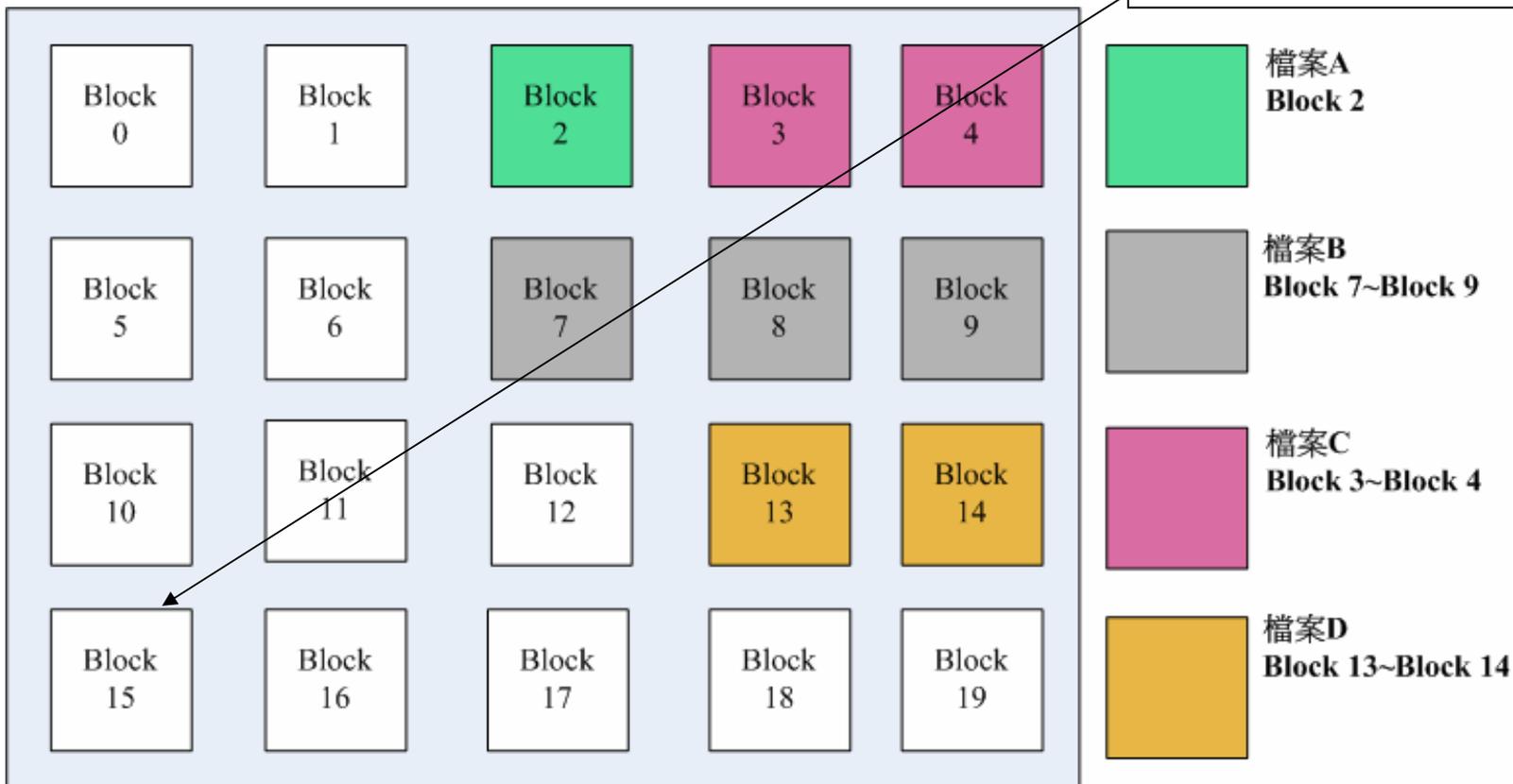
磁碟區塊的配置問題

- 一個檔案的內容如何寫入磁碟？且檔案系統如何知道該資料放置於何處？需要有系統的配置Data block內的區域
 - 連續組織配置
 - 鏈結組織配置
 - 索引組織配置

連續組織配置

- 將檔案直接配置到連續的區塊內
 - 減少存取的時間與次數
 - 但刪除/新增時沒彈性
 - 容易發生斷裂問題

若有需要5個block的檔案，則該檔案只能夠使用block 15~19了。

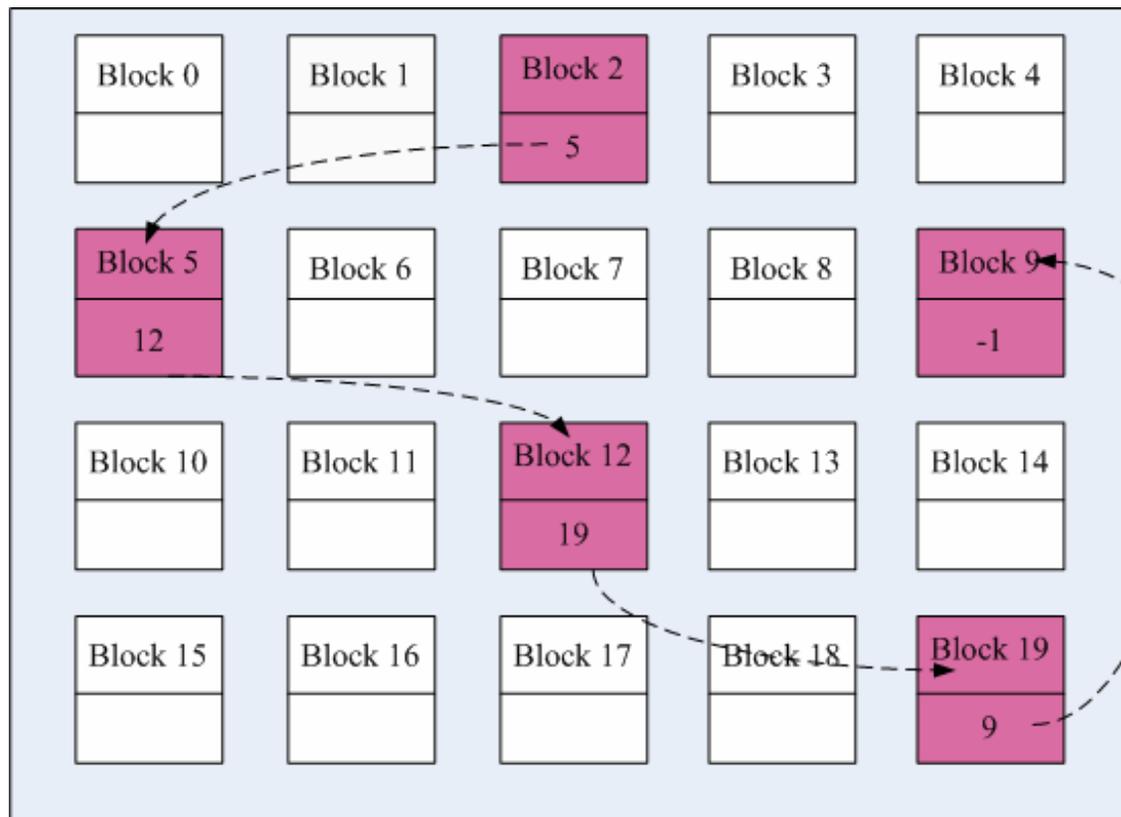


鏈結組織配置

- 資料可散佈於任何邏輯區塊，但有一個指標指向下一個邏輯區塊的位置上
 - 容易進行插入/刪除等
 - 效能很差，如果區塊在不同的磁軌上，更糟糕

Windows的FAT檔案系統就是使用這種方式來進行存取

問：為何需要磁碟重組的動作？

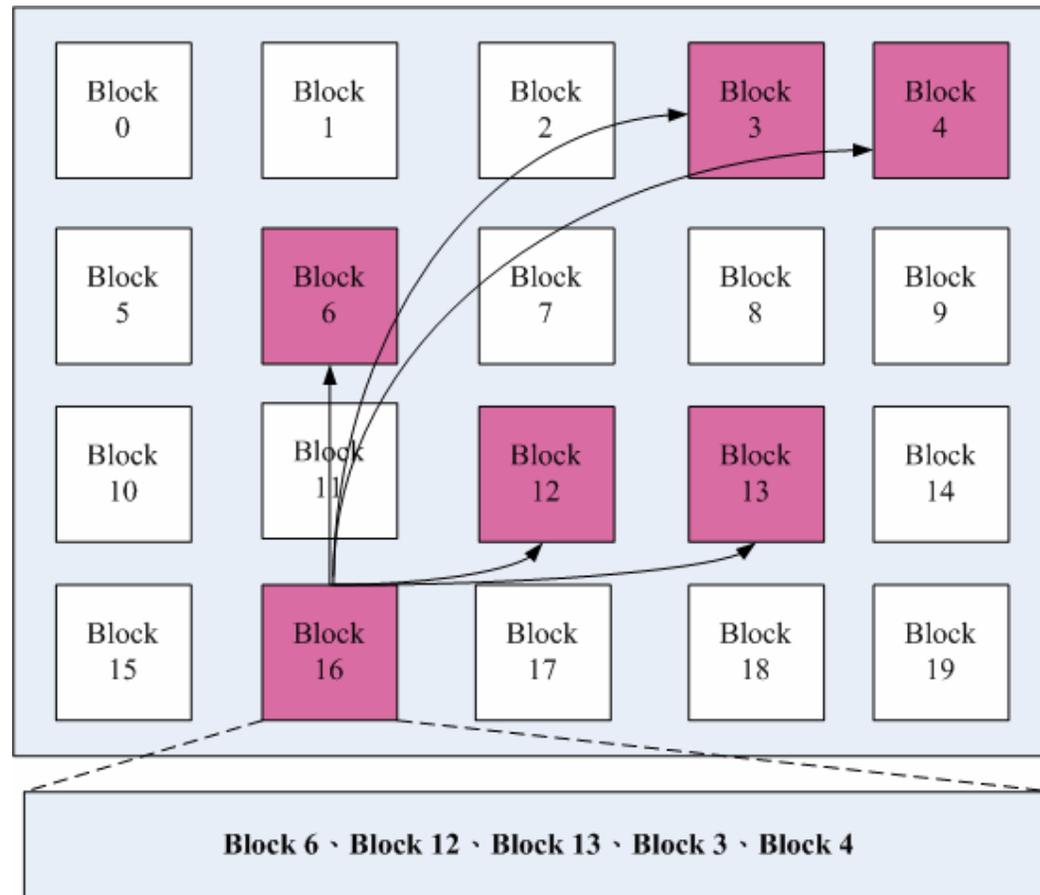


索引組織配置

- 透過一個索引表，將資料所在的所有區塊直接讀出
 - 可直接存取檔案，且不會有所謂的斷裂情況
 - 如果檔案太大，索引表太大，可用多個區塊來索引

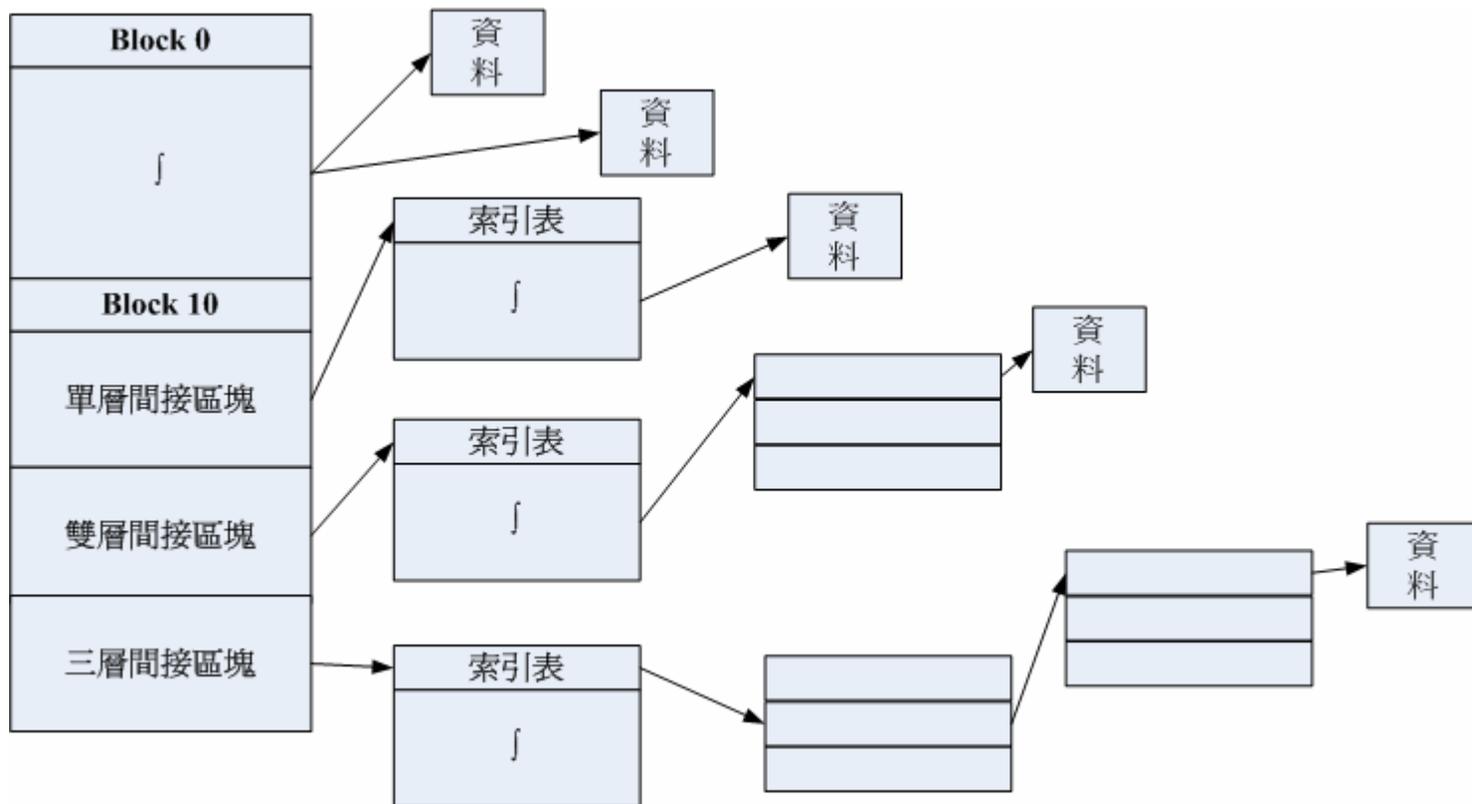
由 block 16 這個索引表得知資料放置於block 3, 4, 6, 12, 13，所以作業系統可以一口氣將這些block 全部直接讀出。

且所有的資料可不必連續放置，因此不會有斷裂問題



索引組織配置—inode配置

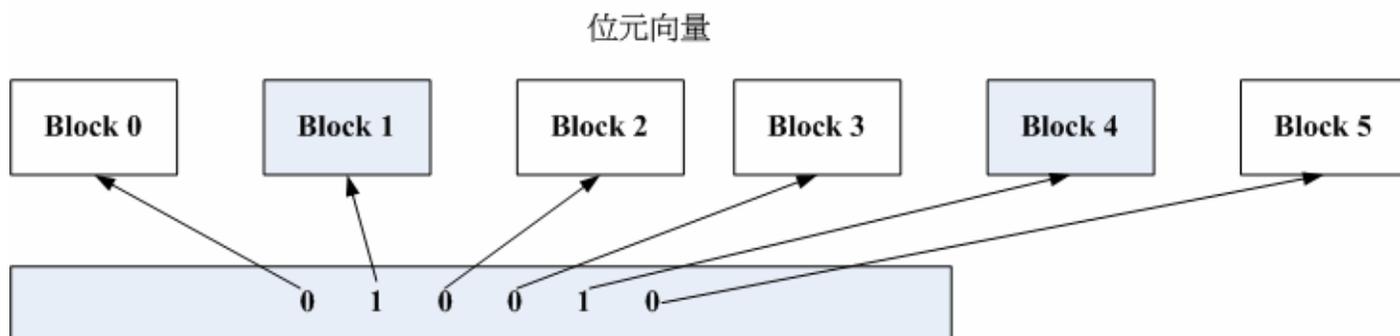
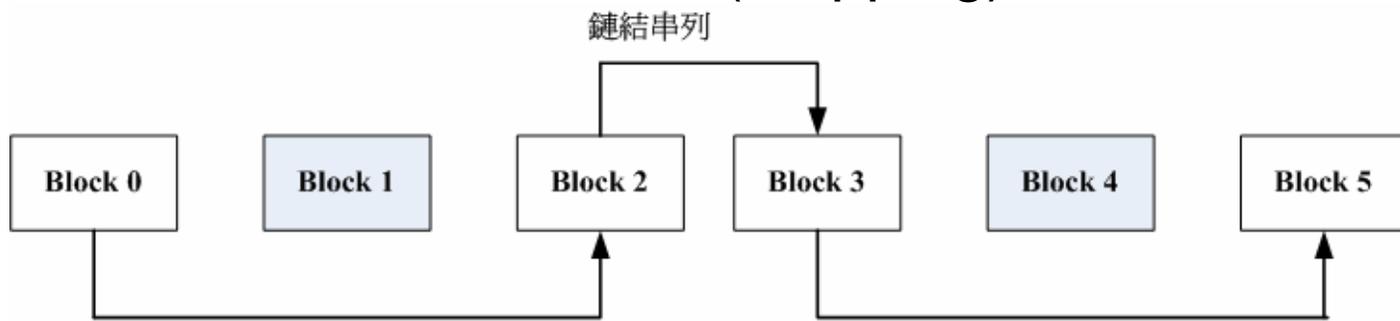
- 記錄block區塊的功能，有直接、間接、雙間接與三間接，據以記錄block的號碼。



可用磁碟空間的記錄方式

■ 問題：

- 新增檔案時，系統怎麼知道那個block是空的，可以記錄資訊？
- 透過一個位元方式來記憶 (mapping)



基本的檔案系統

- 檔案系統主要功能：
 - 開啟、關閉檔案
- 檔案使用的情況的紀錄：
 - 開啟檔案表（Open File Table）
 - 開啟檔案描述（Open File Description）
- 開啟與關閉檔案的函數運作項目

檔案系統的保護機制

- 檔案的保護規範具備的特性：
 - 它是一個抽象的概念。
 - 每個行程都必須運作在某一個保護規範內。
 - 每個保護規範中會定義行程所可以存取的系統資源為何。
- 存取權限（Access Right）
 - 這個就是重點啦！

檔案的擁有者與群組

■ 使用者識別碼 (UID)

- root : 0

- 其他使用者 : 非為 0

- 可登入者 : 大於 500

■ 群組識別碼 (GID)

檔案存取權限設定

■ 分三種身份，每種身份有三種權限

- r 可讀
- w 可寫入
- x 可執行



權限的設定方式

- 三種身份,每種身份有一個分數,分數代號:
 - r:4, w:2, x:1
 - 範例:
 - -rwxr-xr-- : -[rwx][r-x][r--] : 754
 - 練習:
 - 請問 /etc/passwd, 及 /etc/shadow 各是幾分?

本章重點回顧

- 了解檔案系統的架構與功能。
- 了解檔案與目錄間的差異。
- 了解使用者與檔案間的關係與使用者存取檔案的方式。
- 了解檔案系統與裝置配置間的關係。
- 了解檔案系統的保護機制，並透過Linux作業系統所提供的MACL實作。

