

ASIA EDITION

# 作業系統

趙涵捷 審閱

吳庭育 駱詩軒 譯

Operating System  
Concepts TENTH EDITION

ABRAHAM SILBERSCHATZ

PETER BAER GALVIN

GREG GAGNE

東華書局 WILEY



## Chapter 13

# 檔案系統介面





# 章節目標

- 解釋檔案系統的功能
- 描述檔案系統的介面
- 討論檔案系統設計的權衡，包含存取方法、檔案共享、檔案上鎖和目錄結構
- 探討檔案系統的保護



# 13.1 檔案的觀念

- 原始或可執行程式、數字或文字資料、圖像、音樂或影片
- 文字檔案 (text file) 是由一串字元組成一行行文字 (或頁)
- 原始檔案 (source file) 則是由一堆函數組成
  - 每個函數則由一些宣告和一些緊跟著的可執行指令組成
- 可執行檔案 (executable file) 是一系列可讓載入器載入記憶體執行的程式碼區段



## 13.1.1 檔案特性

- 名稱：符號式檔名是唯一用人看得懂的格式儲存
- 識別符號：獨一無二的標籤，通常是一個數字，用來辨識檔案系統內的檔案；它不是人類看得懂的檔名
- 型態：這項資訊對於支援不同檔案型態的系統有需要
- 位置：這項資訊是一個指標指向該檔案所在裝置的位置
- 大小：該檔案目前容量的大小 (以位元組、字元組或區塊為單位)，以及允許以後擴增的最大範圍
- 保護：存取控制資訊控制誰能讀、寫、執行等資料
- 時間、日期和使用者辨識：這項資訊可以保存建立、上次修改和上次使用資料，可以作為保護、安全，以及使用監督



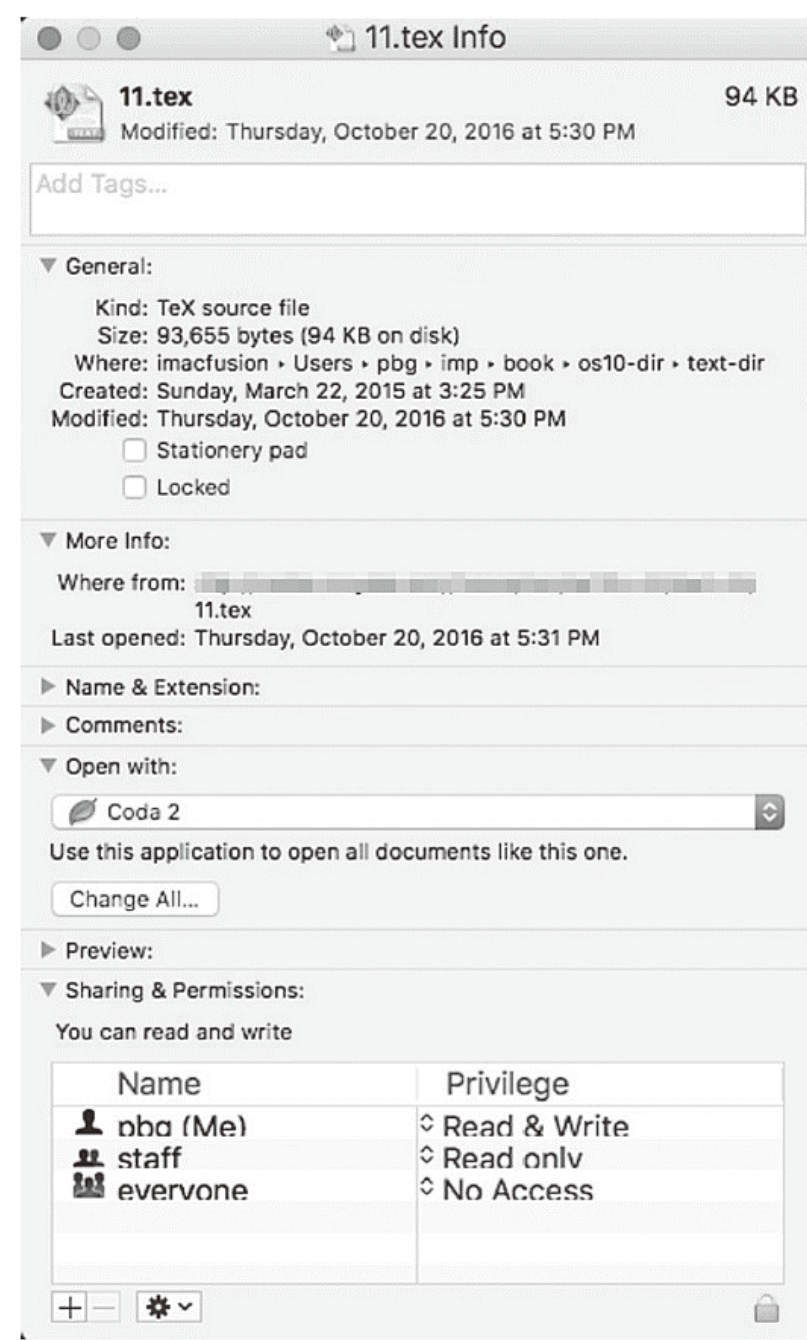
## 13.1.1 檔案特性

- 有些新的檔案系統也支援擴展文件屬性 (extended file attribute)，包含檔案的字元編碼和安全特性
  - 例如檔案核對和
- 圖 13.1 描述 macOS X 上的檔案資訊視窗 (file info window)，它可以顯示一個檔案的屬性





# 圖 13.1 macOS 上的 檔案資訊視窗





## 13.1.2 檔案運作

- 檔案是一個抽象的資料形式
  - 建立檔案：建立檔案需要兩個步驟
    - ◆ 首先，為了這個檔案，其空間必須在檔案系統中被找到
    - ◆ 其次，必須在目錄中為新檔案做一個目錄
  - 開啟檔案：並非所有檔案執行都需要指定檔案名稱
    - ◆ 這個工作是交給作業系統判斷檔案名稱、檢查其存取權限等
  - 寫入檔案：為了寫入一檔案，要做一次系統呼叫，指定開啟檔案名稱和要寫入檔案的資訊
    - ◆ 系統必須保持一個寫入指標 (write pointer) 到檔案的位置，下一個寫入在這個位置





## 13.1.2 檔案運作

- 讀取檔案
- 重置檔案
  - ◆ 這個檔案操作也稱為檔案搜尋 (seek)
- 刪除檔案
- 縮減檔案
- 開啟檔案表 (open-file table) 的表格，以包容所有開啟檔案的資訊





## 13.1.2 檔案運作

- 檔案指標：對於 `read()` 和 `write()` 系統呼叫沒有包含檔案位移的系統而言，它們必須追蹤上一次讀一寫的位置，以作為目前檔案位置的指標。這個指標對於每一個操作此檔案的行程都有一個
- 檔案開啟計數：當檔案關閉時，作業系統必須重新使用該檔所占用的開啟檔案表位置，否則此表的空間可能不足夠
- 檔案的磁碟位置
- 存取權限



## 圖 13.2 Java 中的檔案鎖定範例

```
import java.io.*;
import java.nio.channels.*;

public class LockingExample {
    public static final boolean EXCLUSIVE = false;
    public static final boolean SHARED = true;

    public static void main(String args[]) throws IOException {
        FileLock sharedLock = null;
        FileLock exclusiveLock = null;

        try {
            RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");

            // get the channel for the file
            FileChannel ch = raf.getChannel();

            // this locks the first half of the file - exclusive
            exclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE);

            /** Now modify the data . . . */

            // release the lock
            exclusiveLock.release();
        }
    }
}
```





## 圖 13.2 Java 中的檔案鎖定範例

```
// this locks the second half of the file - shared
sharedLock = ch.lock(raf.length()/2+1,raf.length(),SHARED);

/** Now read the data . . . */

// release the lock
sharedLock.release();
} catch (java.io.IOException ioe) {
    System.err.println(ioe);
}
finally {
    if (exclusiveLock != null)
        exclusiveLock.release();
    if (sharedLock != null)
        sharedLock.release();
}
}
```





# 鎖

- **共用鎖** (shared lock) 類似於讀取者鎖，幾個行程可以同時獲得鎖
- **互斥鎖** (exclusive lock) 表現像寫入者鎖
- 作業系統可能提供**強制** (mandatory) 或**建議** (advisory) 檔案上鎖機制
  - 如果一個鎖是強制的，則因為行程獲得一個互斥鎖，作業系統將避免任何其它行程存取上鎖的檔案





# 圖 13.3 常用的檔案型態

檔案型態	常用延伸部份	功能
可執行檔	exe、com、bin 或 none	準備執行的機器語言程式
目的檔	obj、o	編譯成機器語言，但未鏈結的檔案
原始程式檔	c、cc、java, pas、asm、a	不同語言的原始程式檔
批次檔	bat、sh	命令解譯器的命令
標示檔	xml、html、tex	文書資料、文件
文書處理器檔	xml、rft、docs	不同文書處理器的格式
程式庫	lib、a so、dll	程式人員用的常式庫
列印或觀看	gif、pdf、jpg	ASCII 或二位元可列印或觀看檔案
備份檔	rar、zip、tar	相關檔案組成一個檔案，有時候經過壓縮以做備份儲存
多媒體	mpeg、mov、 mp3、mp4、avi	包含 audio 或 A/V 資訊的二位元檔





## 13.2 存取方法

- 循序存取
  - 最簡單的存取方法
  - 最通用的檔案存取模式

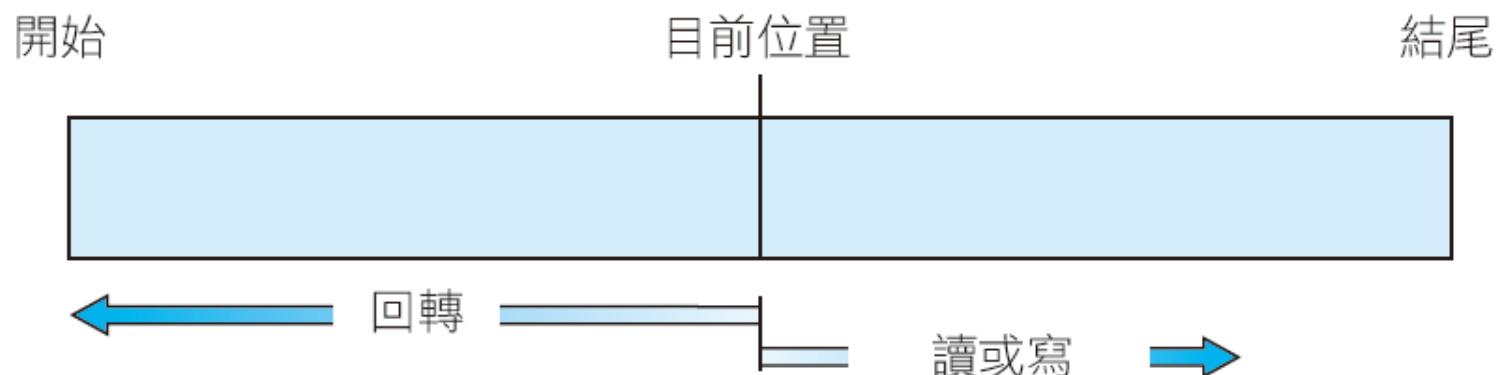


圖 13.4 循序存取的檔案







## 13.2 存取方法

- 直接存取
  - 以檔案存放在磁碟上的模式為基礎，因為磁碟允許隨機存取任何檔案區塊
  - 檔案是由固定長度的**邏輯記錄** (logical record) 所組成，程式不必以一定的順序，快速地讀寫記錄

循序存取	實作直接存取
reset	cp = 0;
read_next	read cp; cp = cp + 1;
write_next	write cp; cp = cp + 1;

圖 13.5 在直接存取檔案上模擬循序存取



## 13.2 存取方法

- 其他的存取方法
  - IBM 的 ISAM (indexed sequential-access method) 使用一個小的主索引指向副索引在磁碟中的區塊

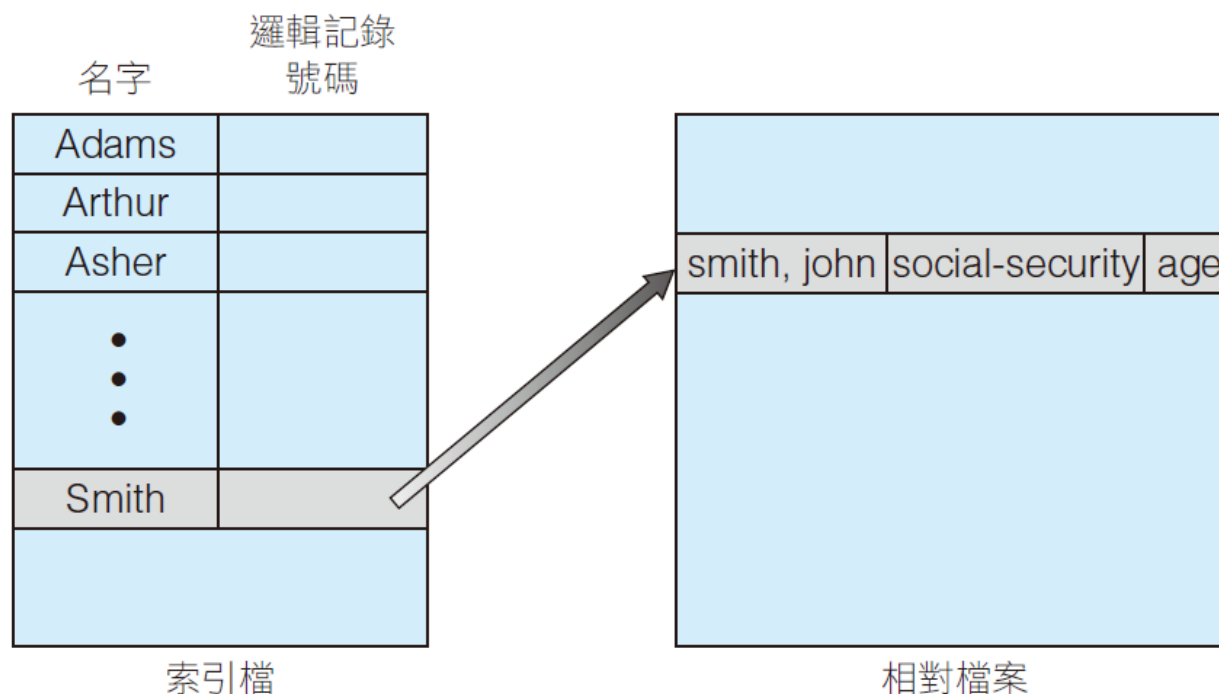
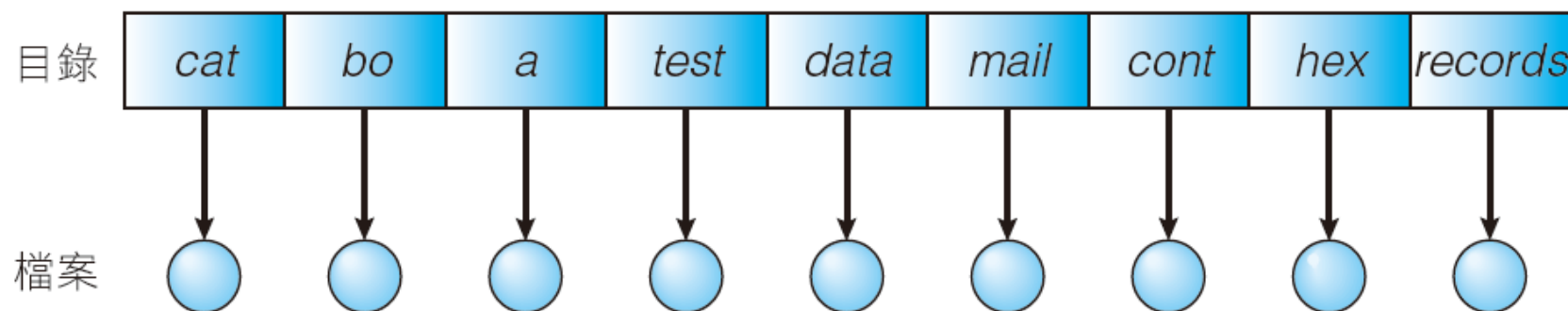


圖 13.6 索引和相對檔案的例子



## 13.3 目錄結構

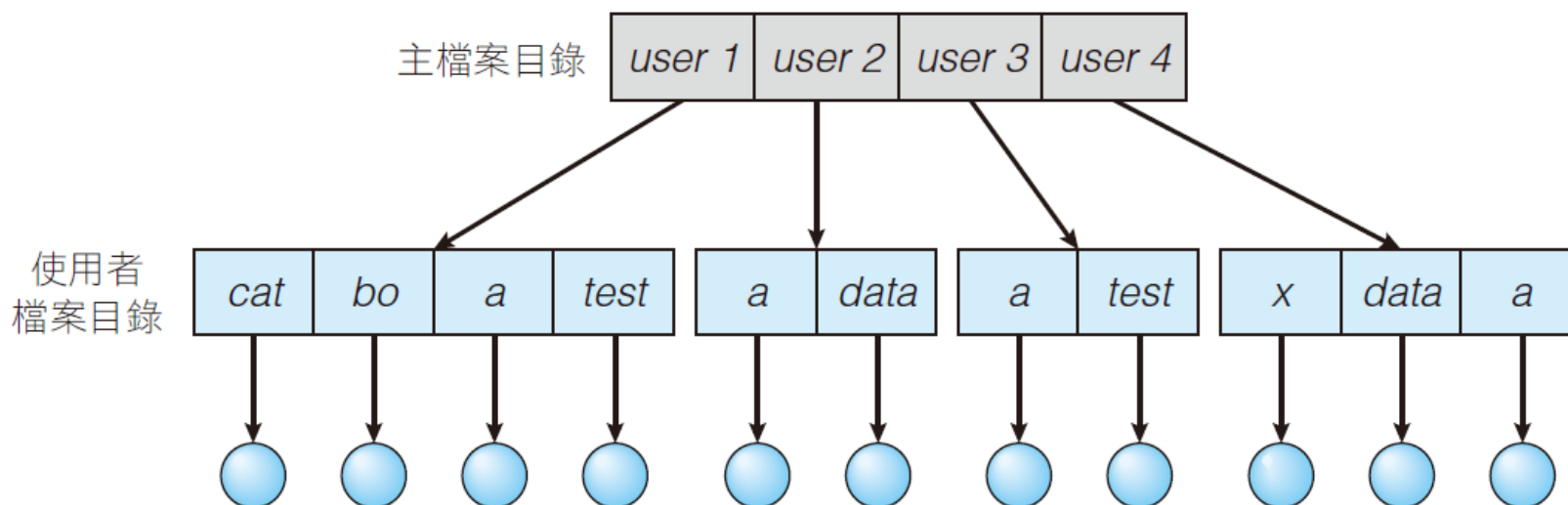
- 單層目錄





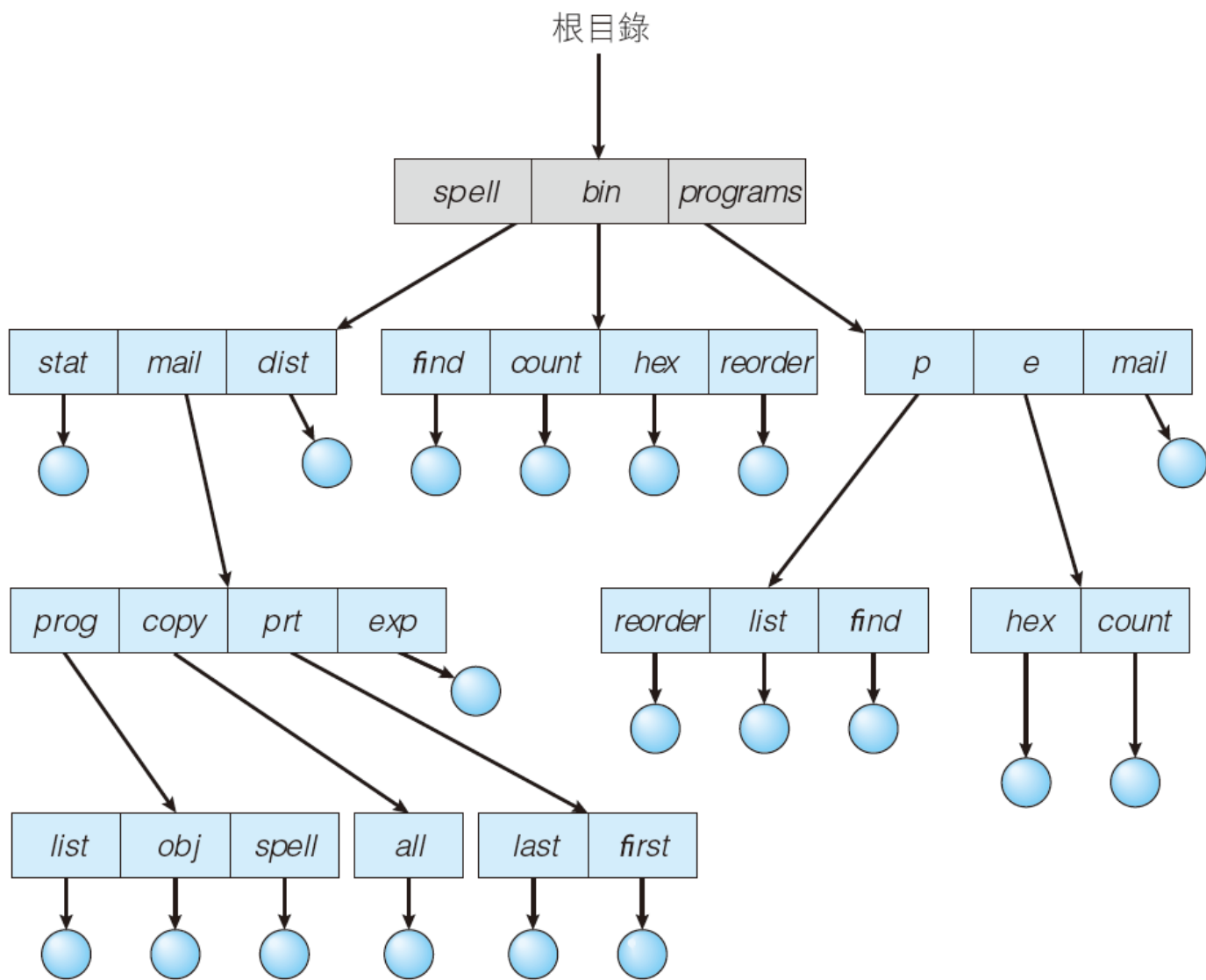
# 13.3 目錄結構

- 雙層目錄



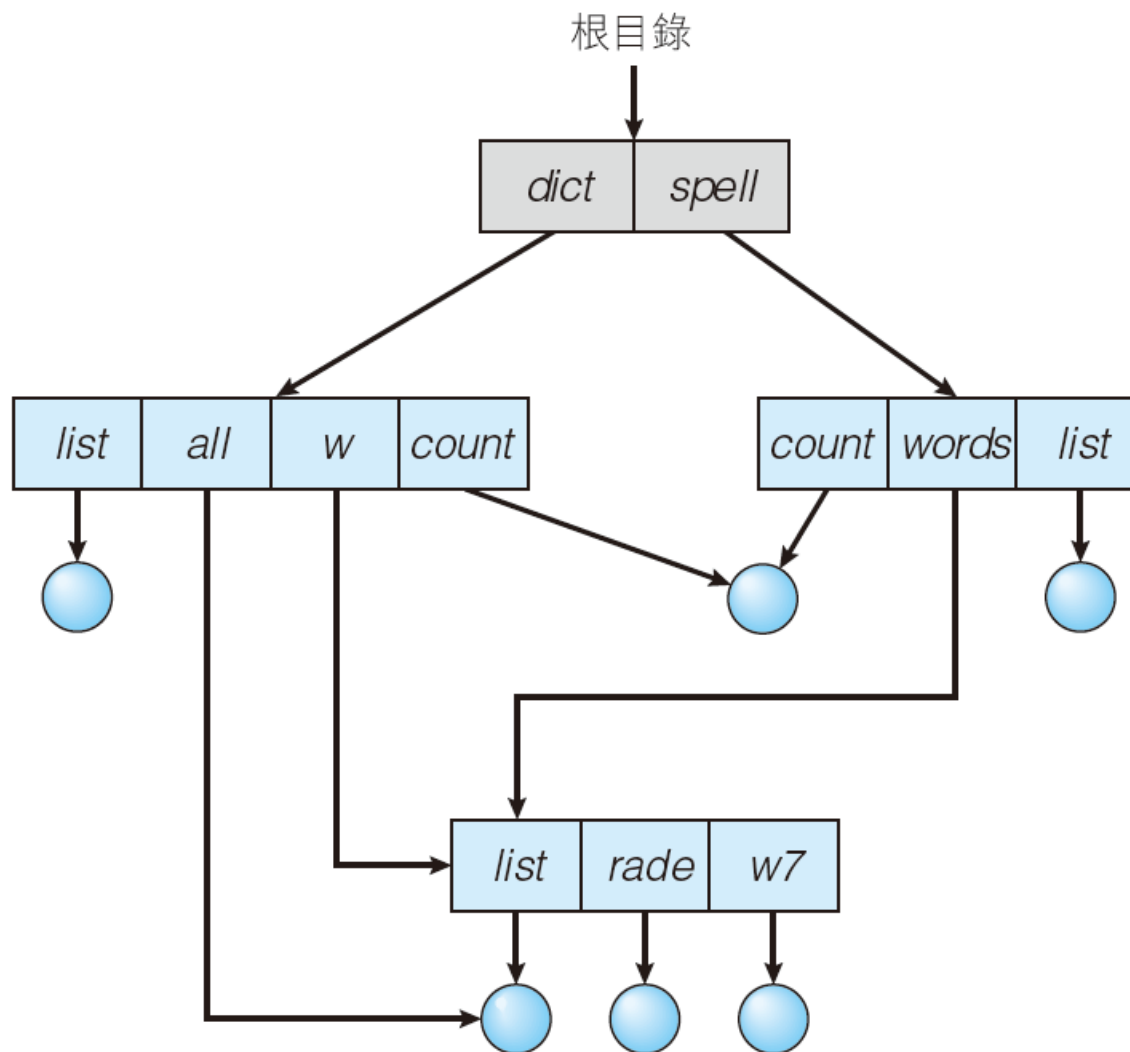


# 13.3.3 樹狀目錄





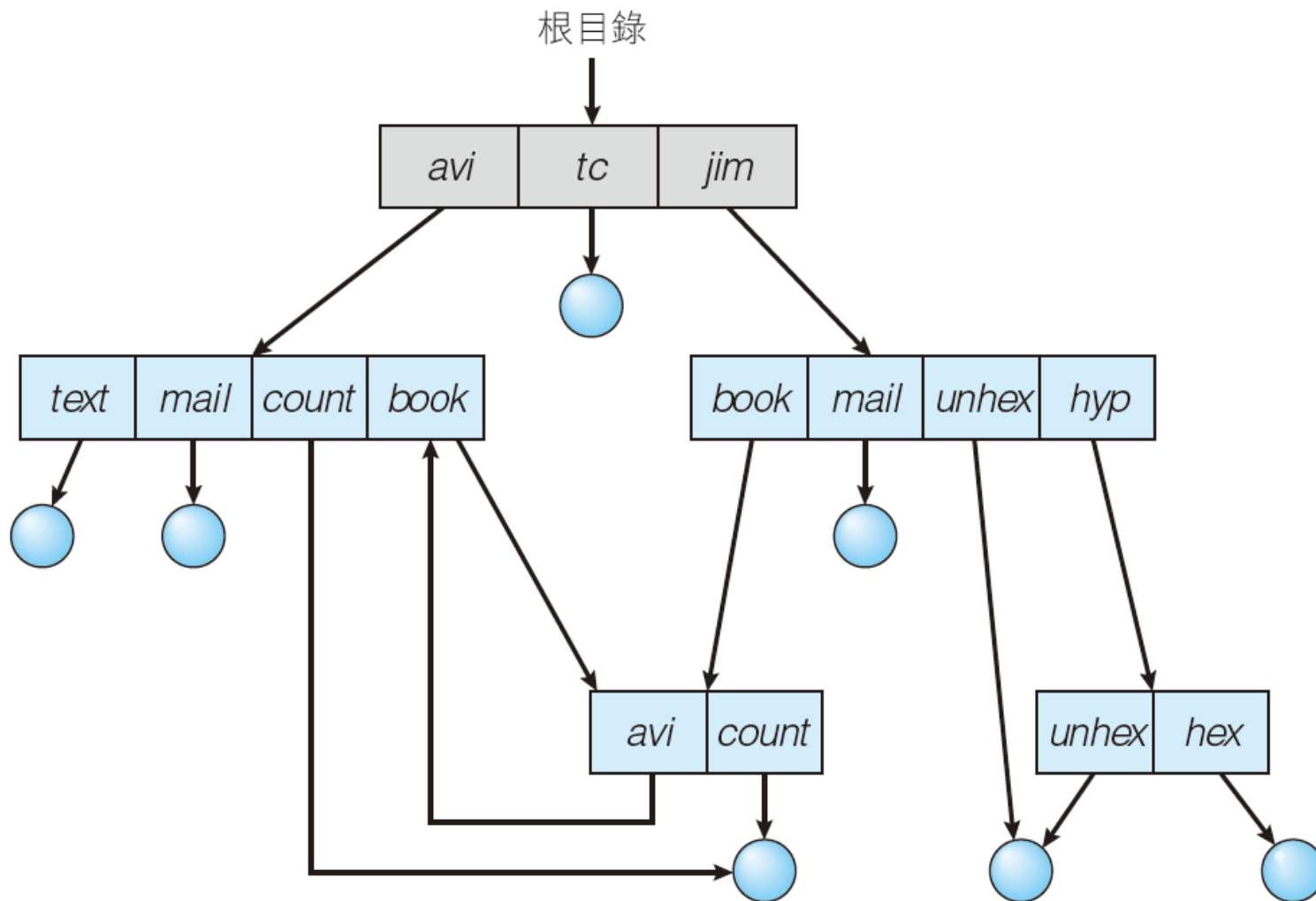
## 13.3.4 非循環圖型目錄







## 13.3.5 一般圖型的目錄





## 13.3.5 一般圖型的目錄

- 廢置空間收集法 (garbage collection)
  - 判斷最後一個參考資料是何時被刪除的，以及磁碟空間可否重新分配
  - 包括檢視檔案系統，標示一切可以存取的東西
    - ◆ 第二次再來將其餘沒有被標示的都收集到可用空間的表列中





# 13.4 保護

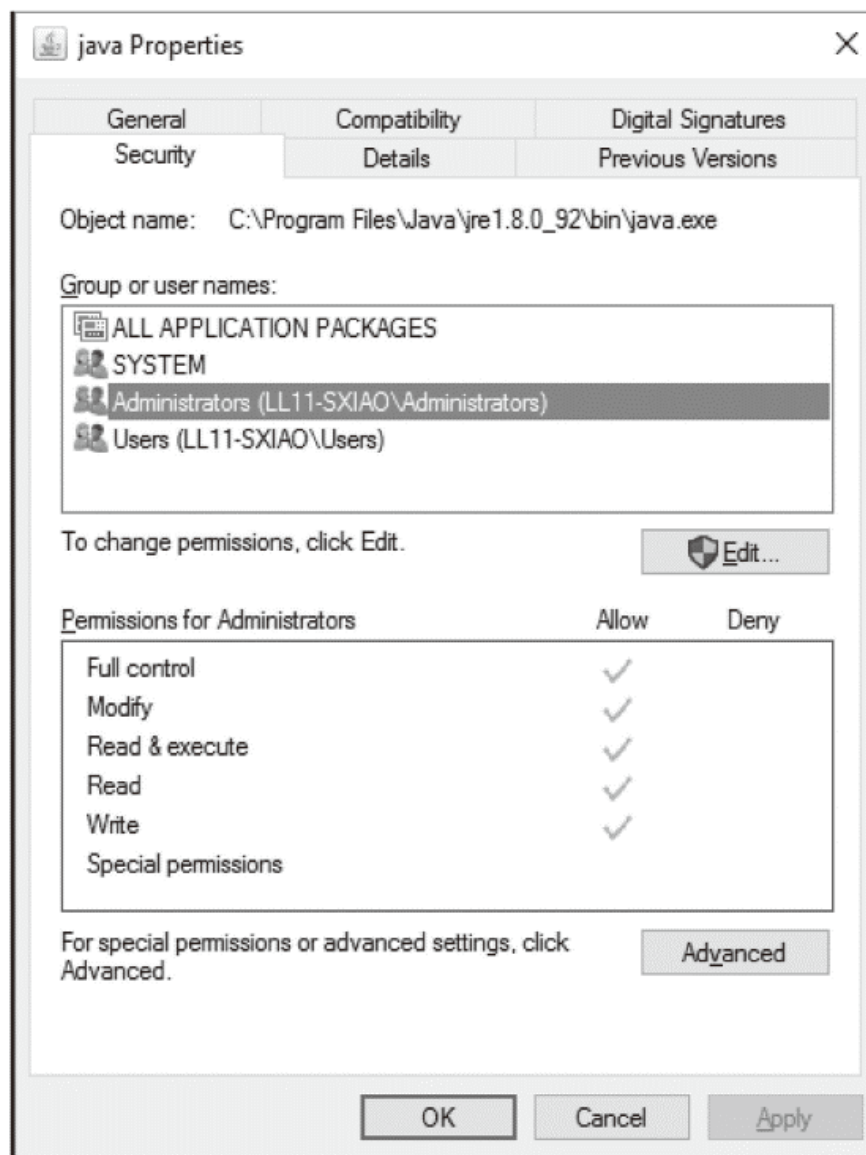


圖 13.12 Windows 10 存取控制列表管理





# UNIX 系統中的權限

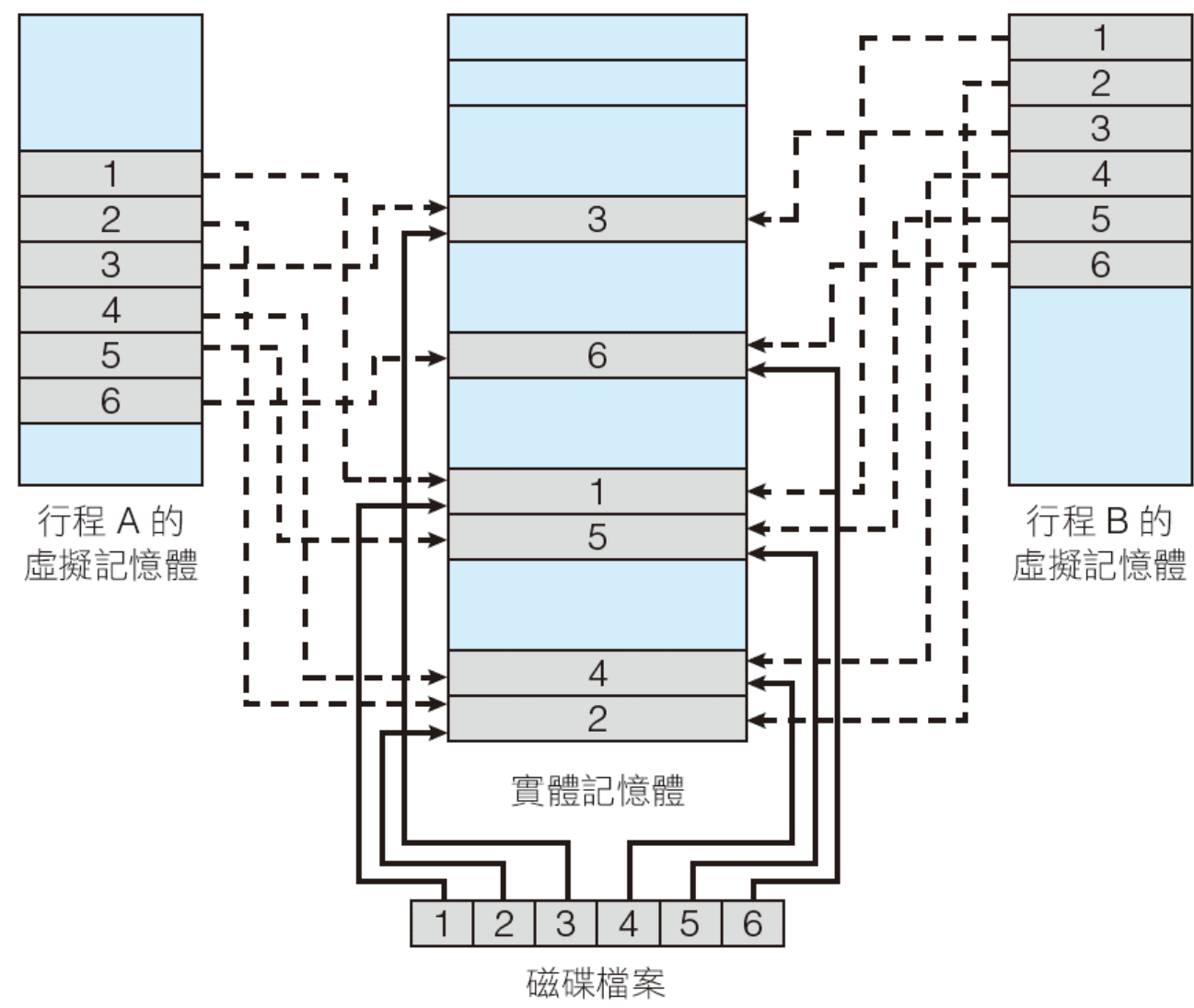
- UNIX 環境中的目錄列表範例如下所示：

-rw-rw-r--	1 pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5 pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2 pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2 jwg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1 pbg	staff	9423	Feb 24 2017	program.c
-rwxr-xr-x	1 pbg	staff	20471	Feb 24 2017	program
drwx--x--x	4 tag	faculty	512	Jul 31 10:31	lib/
drwx-----	3 pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3 pbg	staff	512	Jul 8 09:35	test/





# 13.5 記憶體映射檔案





# 圖 13.14 使用記憶體映射的 I/O 共用記憶體

