Chapter 15

檔案系統內部

章節目標

- 探討作業系統 I/O 子系統的結構
- 討論 I/O 硬體的原則和複雜性
- 解釋 I/O 硬體和軟體方面的效能

圖 15.1　一個典型的儲存器裝置組織

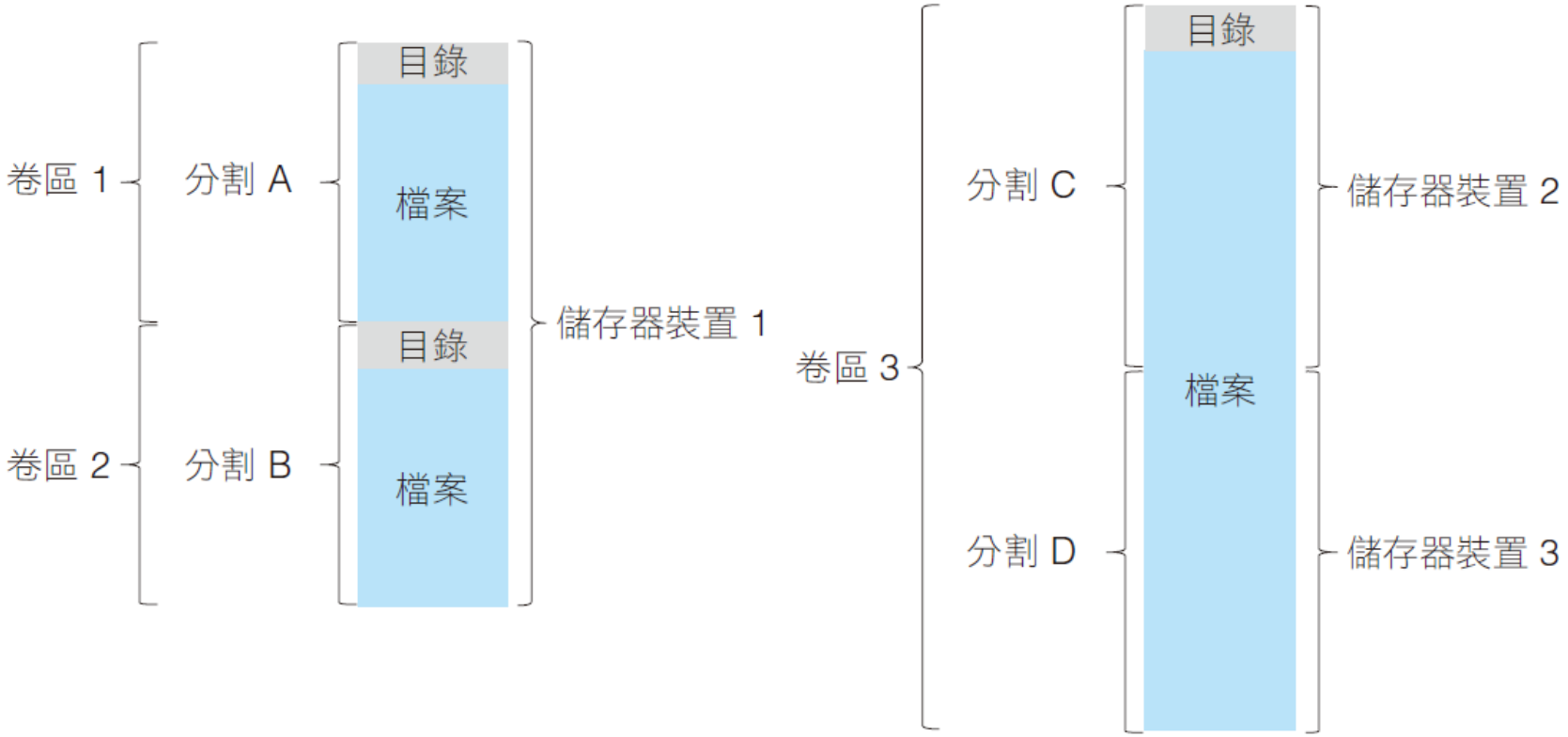


圖 15.2 Solaris 檔案系統

/	ufs
/devices	devfs
/dev	dev
/system/contract	ctfs
/proc	proc
/etc/mnttab	mntfs
/etc/svc/volatile	tmpfs
/system/object	objfs
/lib/libc.so.1	lofs
/dev/fd	fd
/var	ufs
/tmp	tmpfs
/var/run	tmpfs
/opt	ufs
/zpbge	zfs
/zpbge/backup	zfs
/export/home	zfs
/var/mail	zfs
/var/spool/mqueue	zfs
/zpbg	zfs
/zpbg/zones	zfs

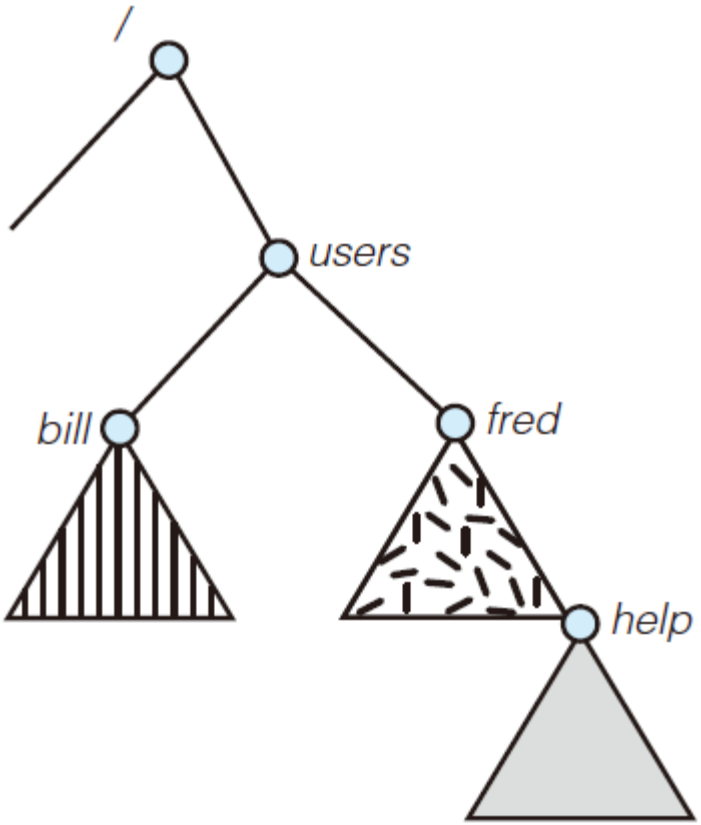
15.1 檔案系統

- tmpfs：一個在揮發性主記憶體產生的“暫時”檔案系統，如果系統重置或毀損時，它的內容就被清除
- objfs：一個“虛擬”的檔案系統(基本上是一個界面到核心，看似檔案系統)，讓偵錯程式存取核心符號
- ctfs：一個維護“合約”資訊以管理系統啟動時那一個行程先開始和操作時那一個行程繼續執行的虛擬檔案系統
- lofs：允許一個檔案系統被存取以取代另一個的“回送”檔案系統
- procfs：一個呈現所有行程資訊為檔案系統的虛擬檔案系統
- ufs、zfs：一般用途檔案系統

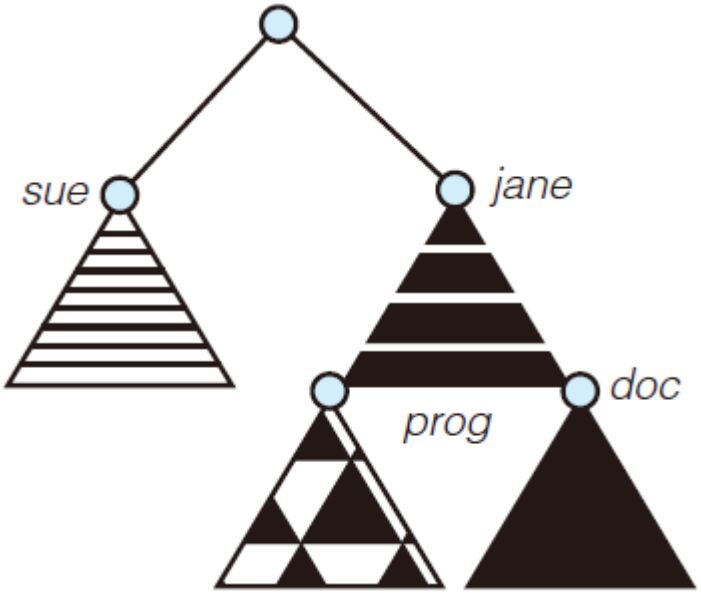
15.2 檔案系統掛載

- 掛載的步驟很直接
- **掛載點** (mount point)：要給予作業系統裝置的名稱，和連結上此檔案系統的檔案結構中位置
  - 有些作業系統需要提供檔案系統型態，而其它系統檢視裝置的結構已決定檔案系統型態
  - 通常掛載點是一個空目錄

圖 15.3 檔案系統

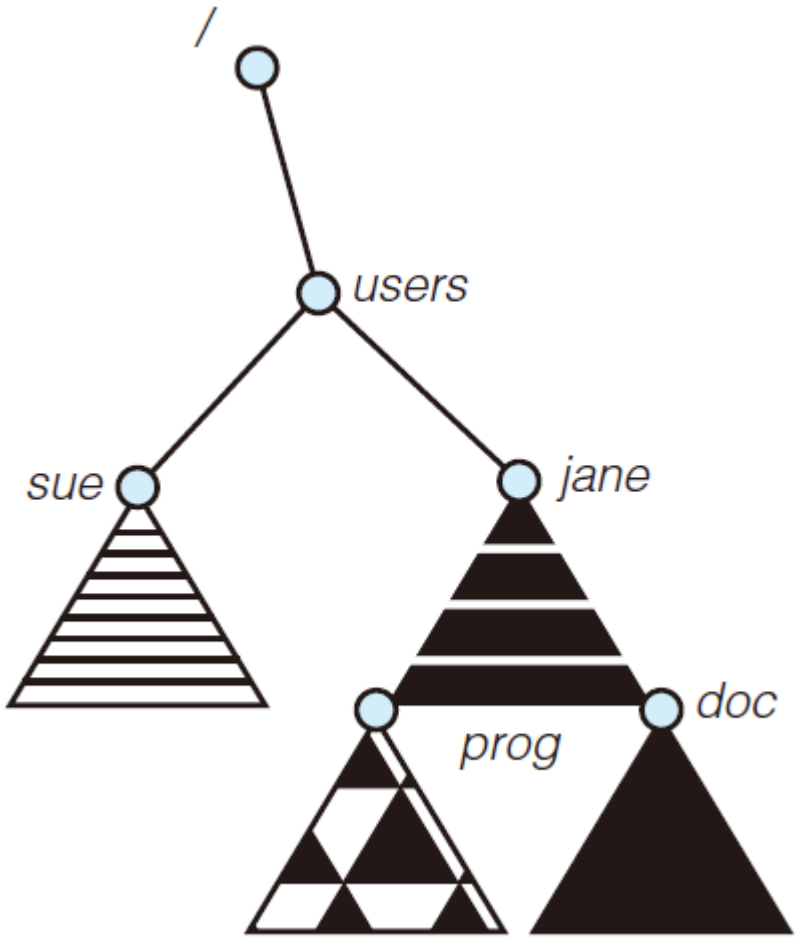


(a) 現存系統



(b) 沒有掛載的卷區

圖 15.4　在 /users 掛載卷區 / 使用者



15.3 分割和掛載

- 每個分割可以是“原始的”(不包含檔案系統)，或是“加工過的”(包含一個檔案系統)
  - **原始磁碟** (raw disk) 被使用在沒有檔案系統的狀況下
- **根分區** (root partition)。其它卷區可以在啟始時自動掛載，也可以之後進行手動掛載，使用哪種方式取決於作業系統
- 作業系統會驗證裝置是否包含有效的檔案系統，這作法是成功進行掛載的功能之一

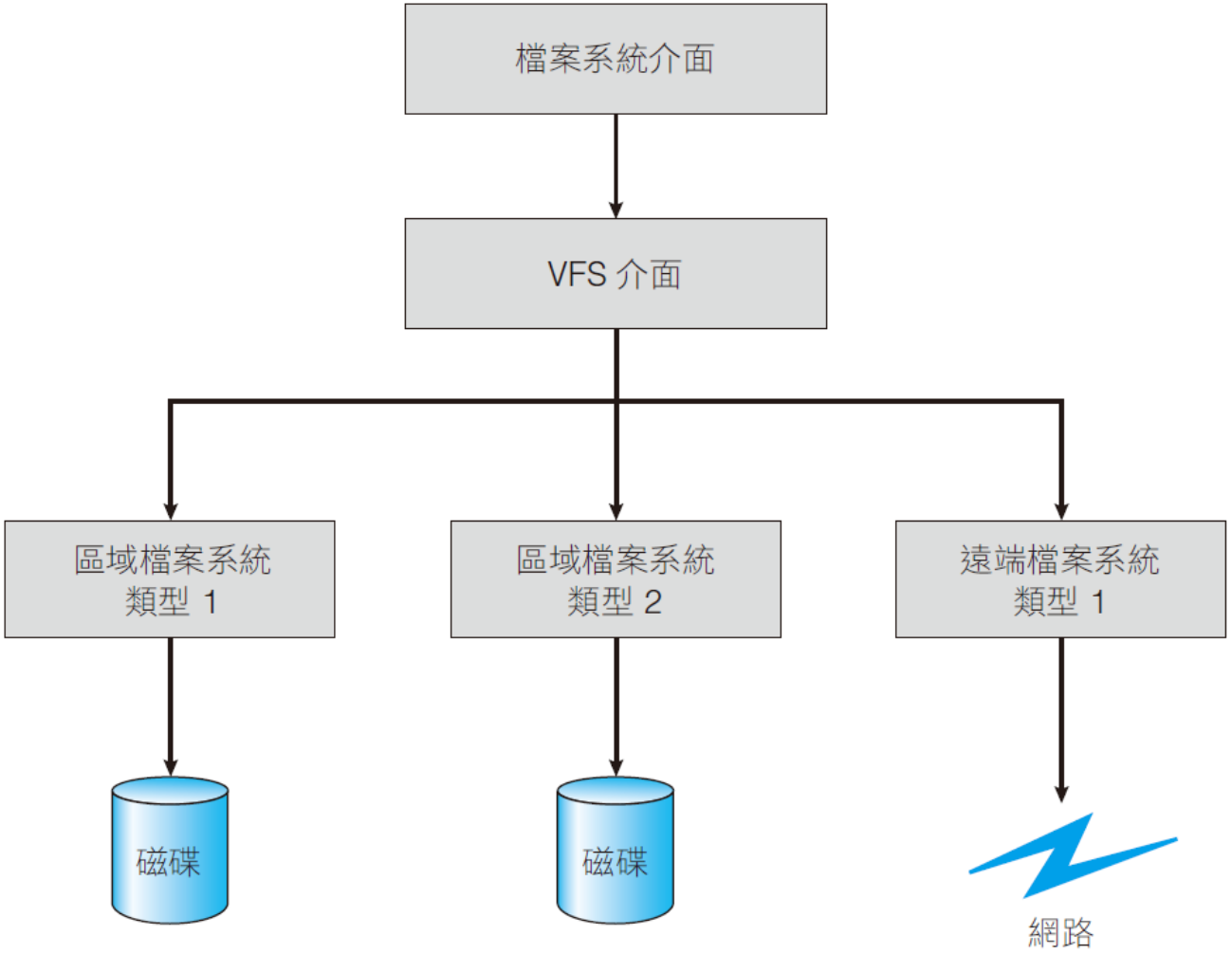
15.4 檔案分享

- 作業系統容納多位使用者時，檔案分享、檔案命名和檔案保護等事項變得很重要
- 為了實作分享和保護，系統必須維護比單一使用者系統更多的檔案和目錄屬性
- 大部份系統已經演化成檔案/目錄**擁有者** (owner)[或**使用者** (user)] 和**群組** (group) 的觀念

15.5 虛擬檔案系統

- 第二層稱為**虛擬檔案系統** (virtual file system, VFS) 層
- VFS 層具有兩個重要功能：
  1. 藉由定義清空的 VFS 介面，將特定檔案系統的操作與其實作分開
  2. 它提供一種在整個網路中唯一表示檔案的機制。VFS 基於稱為 vnode 的檔案表示結構，

圖 15.5　虛擬檔案系統示意圖



15.5 虛擬檔案系統

- 讓我們簡要地研究 Linux 中的 VFS 架構。Linux VFS 定義的四種主要物件類型：
  - **inode 物件** (inode object)：代表個別檔案
  - **檔案物件** (file object)：代表一個開啟的檔案
  - **superblock 物件** (superblock object)：代表整個檔案系統
  - **dentry 物件** (dentry object)：代表個別目錄項目

15.5 虛擬檔案系統

- VFS 定義一組可以完成的實作
- 功能表列出實現該特定物件定義的實作實際功能的位址
  - int open (. . .)—開啟檔案
  - int close (. . .)—關閉已經開啟的檔案
  - ssize_t read (. . .)—從檔案中讀取
  - ssize_t write (. . .)—寫入檔案
  - int mmap (. . .)—記憶體映射檔案

15.6 遠端檔案系統

1. 使用者以人工的方式經由類似 ftp 的程式在機器之間傳送檔案
2. **分散式檔案系統** (distributed file system, DFS) 讓遠端的目錄在本地端可以看得見
3. 全球資訊網 (World Wide Web) 是第一種方法的反轉
   - 網頁瀏覽器需要用來取得遠端檔案的存取權
   - 其它的操作 (主要是 ftp 的包裝版) 則被用來傳輸檔案
   - 雲端運算也正被用在檔案分享

15.6.1 客戶端 - 伺服器模型

- 遠端檔案系統允許電腦從一台或多台遠端機器掛載一個或多個檔案系統，包含檔案的機器
  - **伺服器** (server)
  - 希望存取檔案的機器是**客戶端** (client)
- 客戶可以用它們的網路名稱或其它識別碼 (例如 IP 位址) 來指定，但是這可以被**欺騙** (spoof) 或模仿

15.6.2 分散式資訊系統

- **分散式資訊系統** (distributed information system)，也稱為**分散式命名服務** (distributed naming service)
  - 被發明用來提供遠端運算所需資訊的一致性存取
  - **領域名稱系統** (domain name system, DNS) 提供整體網際網路的主機名稱到網路位址的轉換
- 其它分散式資訊系統對分散式裝置提供使用者名稱/密碼/使用者 ID/群組 ID 的空間
- 產業正朝向**輕型目錄存取協定** (lightweight directory-access protocol, LDAP) 作為安全和分散的命名機制

15.6.3 失效模式

- 區域檔案系統可能因一些不同的原因而失效
  - 存放檔案系統之磁碟的失效
  - 目錄結構
  - 磁碟管理資訊 [統稱為**元資料** (metadata)] 中繼資料的損毀
  - 磁碟控制器的失效
  - 纜線失效
  - 主機轉接器失效
- 製作這種從失效復原的模式，有些種類的**狀態資訊** (state information) 可能在客戶端及伺服器上皆加以維護

15.7 一致性語意

- UNIX 語意
  - 一位使用者對一已開啟檔案進行寫入的動作時，可被其它也開啟該檔案的使用者立即看見
  - 具有共用的模式，在該模式下使用者共同指向檔案目前位置的指標
- 會議語意
  - 一位使用者對一已開啟檔案進行寫入動作時，無法被其它也開啟該檔案的使用者立即看見
  - 一旦關閉一個檔案，其所做之改變只能在下一次會議中看見

15.7 一致性語意

- 不變共用檔案之語意
  - **不變共用檔案** (immutable shared file) 有兩個重要性質：
    - 名稱不能重複使用
    - 內容不可更動

15.8 NFS

- NFS 是經由 LAN (或甚至是經由 WAN) 存取遠端檔案之軟體系統的製作方式及規格
- NFS 製作是 Solaris 作業系統的一部份
  - Solaris 是 UNIX SVR4 修正版本，它使用 TCP 或 UDP/IP 協定
    - 根據連接的網路

NFS 概 論

- NFS 將一組互相連接的工作站視為一組具有獨立檔案系統的獨立機器
  - 應明確要求而言，其目標是要以透明的方式來在這些檔案系統間做到某種程度的共用
    - 在外部要求上
- 此掛載上之目錄看起來就像區域檔案系統下一個完整的子目錄樹一樣，取代從區域目錄傳下來之子目錄樹
  - 區域目錄變成新架設目錄之根目錄
  - 作為掛載操作參數的遠端目錄之規格是以非透明方式完成
    - 必須先提供遠端目錄之地點 (即主機名稱)

NFS 概 論

- 然而，從那時起在 *M1* 機器上的使用者可以完全透明的方式存取在遠端目錄之檔案
- 由於存取權的認定，任何檔案系統或是檔案系統內的一個目錄可以遠端地掛載在任何區域目錄之上
  - 無磁碟的工作站甚至可從伺服器上掛載其本身之根目錄
  - 串接式掛載(cascading mount) 也可以允許在某些NFS上實作

NFS 概 論

- NFS 設計目的之一是，能在不同機器、作業系統及網路架構之異構環境下操作
  - NFS之規格與上述媒介均無關，因而促進其它製作方式
  - 此獨立性是藉由使用建立在外部日期表示 (external data representation, XDR) 協定上的 RPC 運作來達成，XDR 協定是用在兩種與製作方式無關的介面之間
  - 因此若系統由異構的機器與檔案系統組成，並且有適當的NFS 介面，則不同種類的檔案系統便可以區域和遠端的方式掛載

圖 15.6　三個獨立的檔案系統

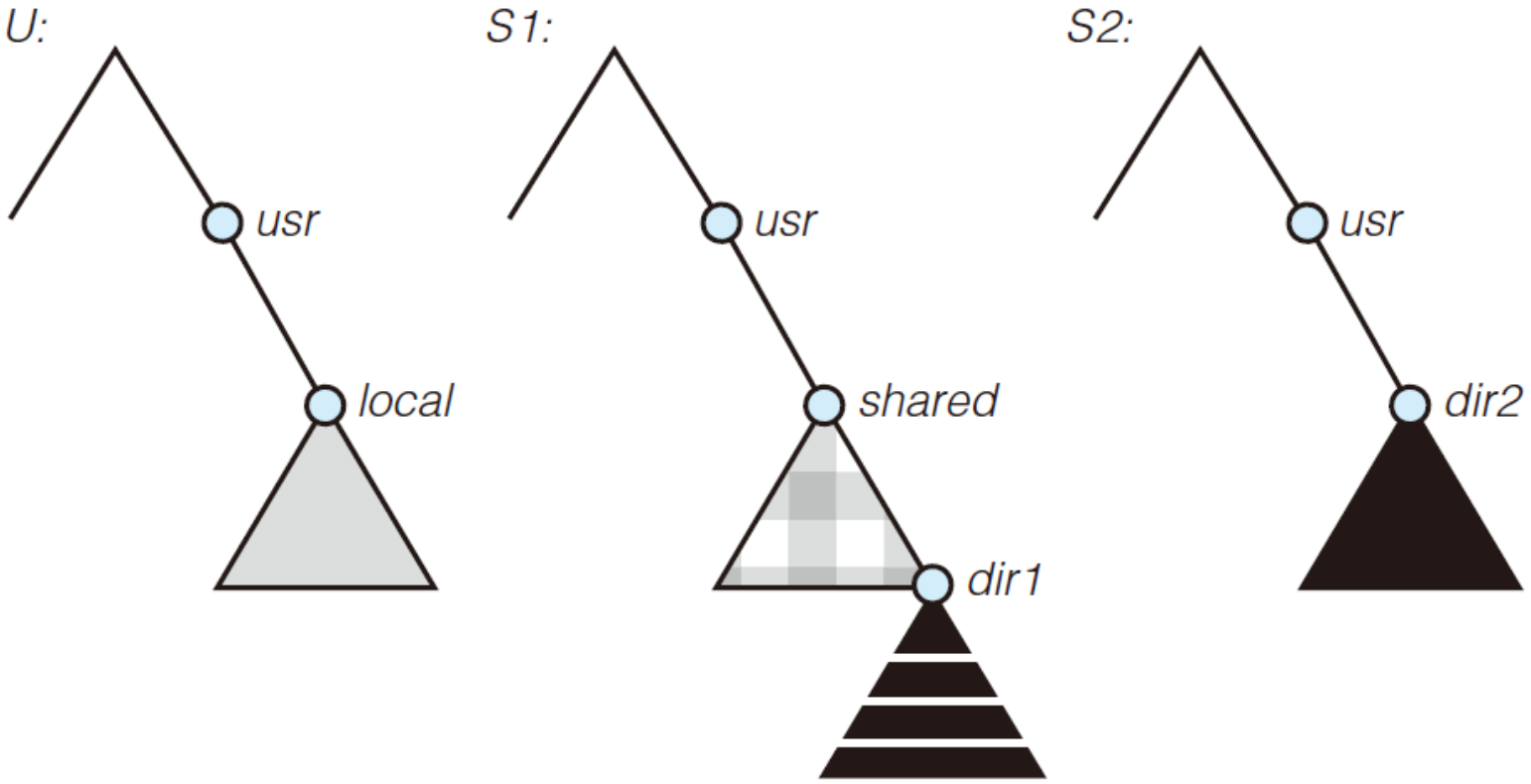
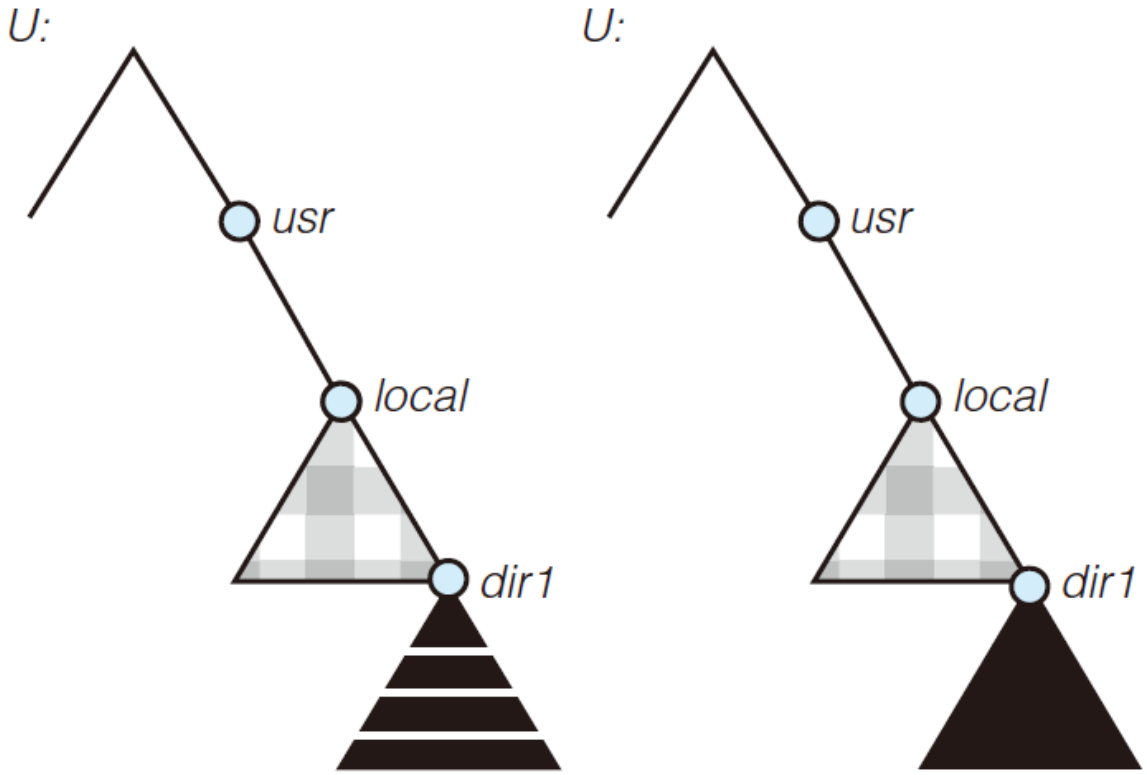


圖 15.7　掛載於 NFS



(a) 掛載　　(b) 串接式掛載

15.8.2 掛載協定

- **掛載協定** (mount protocol) 是用來在伺服器與客戶間建立起始之邏輯連接
- 掛載的操作包括所掛載的遠端目錄名稱及儲存它的伺服器機器名稱
  - 掛載要求被映射至相對應之 RPC，並傳送至在特定伺服器機器上執行之掛載伺服器
  - 伺服器維持一份**輸出串列** (export list) 而這個串列規定要導出掛載的區域檔案系統，以及允許掛載它們的機器名稱
  - 案系統識別碼和 inode 號碼，用以辨識輸出檔案系統的正確掛載目錄

15.8.3 NFS 協定

- NFS 協定提供一組供遠端檔案操作的遠端程序呼叫。此程序提供下列運作：
  - 在一目錄內搜尋一個檔案
  - 讀取一組目錄條目
  - 操作連結及目錄
  - 存取檔案屬性
  - 讀取及寫入檔案

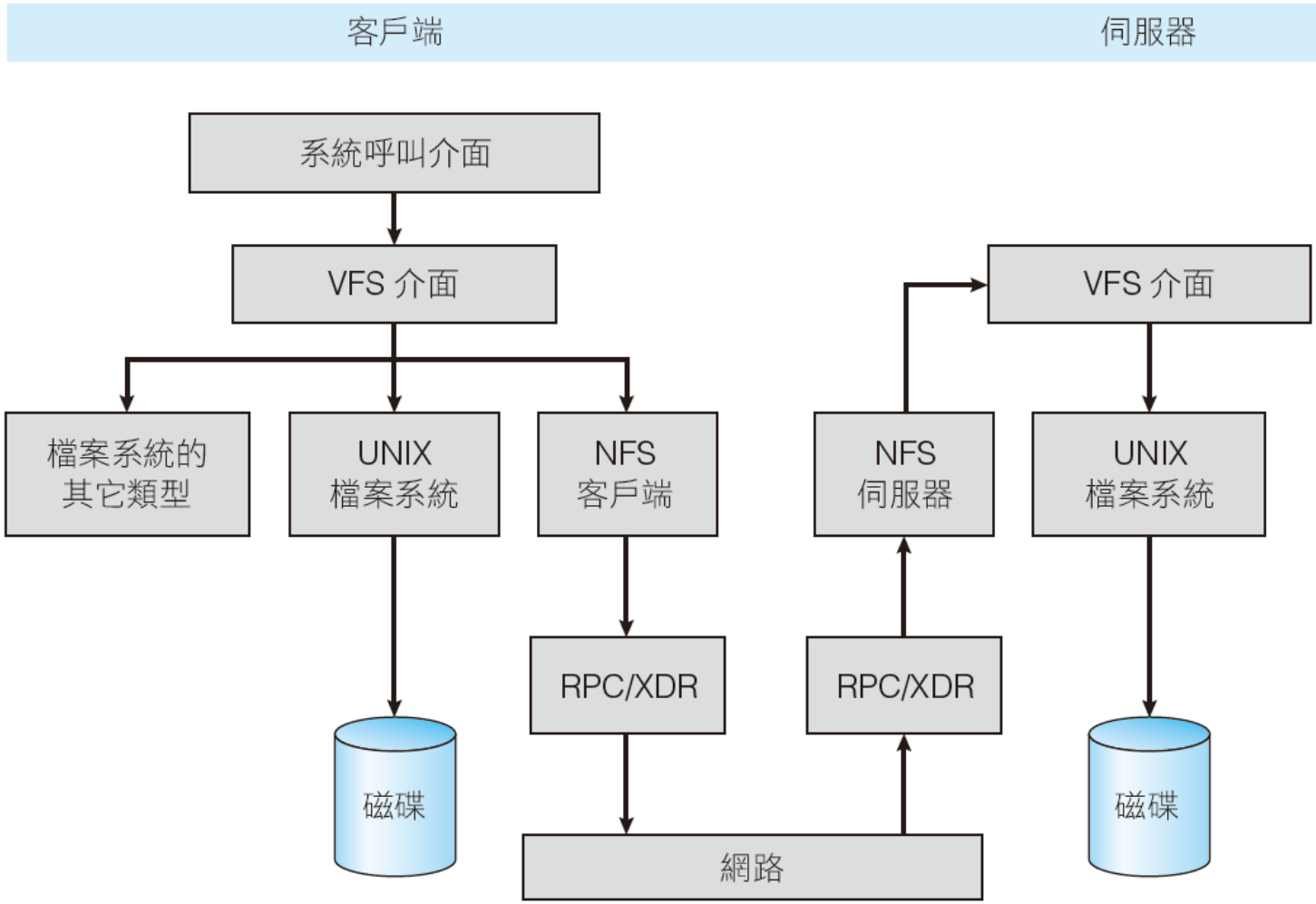
15.8.3 NFS 協定

- NFS 伺服器一項凸顯的特色即為無狀態
  - 伺服器並不需要維持切換存取點時取用之客戶端資訊
  - 在伺服器方面，亦無類似於 UNIX 之開啟檔案表格或是檔案結構
  - 損壞後並不需要採取特別策略來恢復一台伺服器。為了此目的起見，檔案操作必須具相同效力。

15.8.3 NFS 協定

- NFS 經由 VFS 整合到作業系統中
  - 為了說明此架構，讓我們追溯如何處理已打開的遠端檔案上的操作
  - 系統層次將此呼叫映射至適當 vnode 上之 VFS 運作，VFS 將此檔案視為遠端的，並且引發適當的 NFS 程序
  - 在遠端伺服器上對 NFS 服務層做出一項 RPC 呼叫

圖 15.8　NFS 架構示意圖



15.8.4 路徑名稱轉譯

- 安排方式，而該命名空間是由其執行之掛載方式所指揮，因此需要這種高成本的路徑名稱追蹤方式
- 將一個路徑名稱交給伺服器，以及在遇到掛載點時接收一個目標 vnode
- 為了使搜尋更加快速，在客戶方面有一項目錄名稱搜尋快取記憶體，其中為遠端目錄名稱持有 vnode

15.8.5 遠端運作

- 除了開啟及關閉檔案外，在普通檔案運作之 UNIX 系統呼叫與 NFS 協定之 RPC 間，幾乎具有一對一之對應關係
  - 因此一項遠端檔案運作可直接轉譯成對應的 RPC
  - 概念上來說，NFS 附屬於遠端服務模式上，但實際上為了效能之考慮，均採用緩衝及快取之技術
  - 在遠端運作及 RPC 之間並沒有直接對應的關係
  - 相反地，檔案區塊及檔案屬性是由RPC 提取，並且在區域快取記憶體中進行
  - 在某些一致性限制下，更進一步的遠端運作均使用快取記憶體中之資料

15.8.5 遠端運作

- 核心檢查遠端伺服器以決定是提取或使快取屬性再生效
  - 只有在對應的快取屬性是最新時，才會使用快取檔案區塊
  - 每當新的屬性從伺服器抵達時，屬性之快取記憶體便被更新，快取之屬性在 60 秒後就被丟棄
  - 在伺服器及客戶端間使用往前讀取及延遲寫入之技術，客戶端等到伺服器證實資料已被寫入磁碟後才釋放延遲寫入之區塊