第六章 交談層、表現層與應用層

將介紹交談連線之管理與獨立資料格式表示法，並詳述網路作業系統、分散式處理架構、以及 Web-based 網路應用環境。

6-1應用導向服務簡介

在 OSI 參考模式之中，**『交談層』(Session Layer)**、**『表現層』(Presentation Layer)** 和 **『應用層』(Application Layer)** 等三個層次主要是提供一個網路應用程式的發展環境，讓使用者發展各式各樣的應用系統，因此，稱之為**『應用導向服務』(Application-Oriented Service)**。早期 OSI 希望在網路應用方面能將一些共通服務加以分類，而在各個層次上定義許多標準介面，讓使用者利用這些標準介面來開發應用軟體，以增加應用程式之間的共通性。然而 OSI 對網路上的應用期望過高，同時功能分類也太過完美，造成軟體開發的困難度過高，相對阻礙 OSI 網路的發展。另一方面，網路上的應用隨時代變遷，對系統的需求更是變化莫測，早期所定義的標準模式已不符現今系統的需求。再者，標準化的功能介面，雖然能讓使用者節省不少開發的時間，但也相對阻礙了新功能發展的空間。

相反的，人們較偏好連接性高且應用簡單的網路，這種需求剛好符合 ArpaNet 網路的特性，於是造就了 Internet 網路的風行；Internet 網路將網路的應用推到最高峰，同時延伸到社會各角落。然而 ArpaNet 網路上並沒有將應用層再細分為交談層及表現層，有關這兩層的工作都在應用層上完成。也就是說，這兩層的功能並沒有標準介面，隨著應用程式的需求各自開發使用。雖然它減少了應用系統之間的共通性，卻增加各系統發展的空間。正因如此，更加速了網路應用的系統發展，也延伸不少新技術的誕生。本章就這三個層次應具有的功能加以介紹，但較著重於應用層的軟體技術。

6-2交談層功能

『交談層』(Session Layer 或稱為會議層) 負責提供交談服務的介面，主要功能是達成使用者 (或程序) 之間對談 (dialogue) 的連接、同步、以及管理它們之間資料的傳送。交談

層負責根據使用者是否同時傳送或接收資料，來控制使用者傳送或接收的時序，也就是所謂的『**同步交談**』（**Synchronization**）功能，其實這個功能非常類似作業系統中多元程序（Multi-Processes）之間的同步處理。我們可由下面三個主要功能來探討交談層的功能：

- 交談連線之建立及維護
- 交談連線之管理
- 交談連線之復原

6-2-1 交談連線之建立及維護

當傳輸層已完成程序之間的連線（Process-to-Process Connection），表示連線雙方的程序可以利用這一條連線來通訊。但當兩個程序建立連線之後，並不表示雙方隨時在通訊之中，交談層的工作就是針對已建立完成的連線，進行建立對談的程序，管理雙方對談的程序，包括交談的建立、管理、終止與支援檢查點、重新啟動等功能，交談層亦稱為『**對談連線**』（**Dialogue-to-Dialogue Connection**），如圖 6-1 所示。這種現象就如同打電話一樣，傳輸層以下的工作讓我們接通電話（Station-to-Station Connection）、並經過分機號碼找到受話對方（End-to-End Connection），雙方雖然都拿取話機，但要如何來對話？是決定雙方一起說話？還是只允許一方可以發話？或發話當中有哪些資料特別重要必須加以管理等等，這些都是交談層所必須負責的工作。

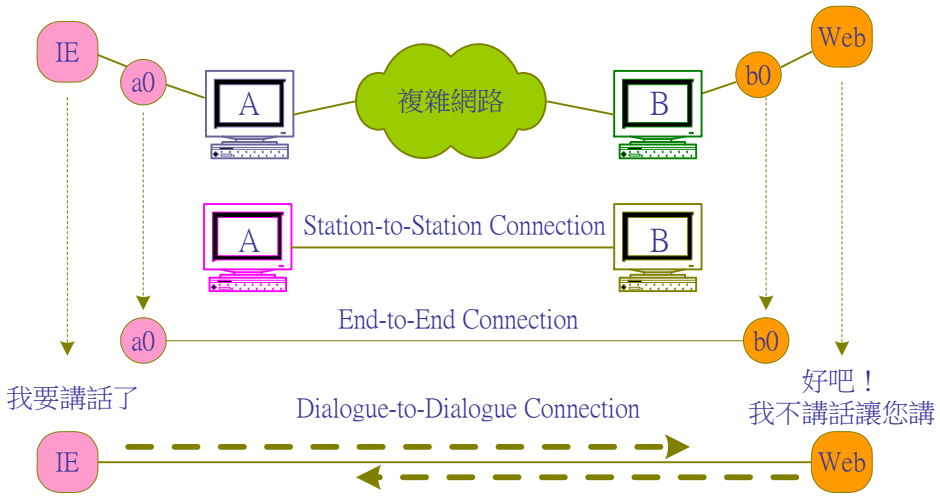

圖 6-1 交談層之對談連接

應用層的兩個通訊軟體之間，必須先建立一個交談（Session），將交談銜接到傳輸層，並商討交談的參數（如資料單元大小）。我們將交談層使用者和通訊軟體之間的交換資料稱為紀錄（Record）。交談層接受紀錄後將其編成**『交談協定資料單元』（Session Protocol Data Unit，SPDU）**，然後透過傳輸層發送資料。傳輸層將 SPDU 分割成若干個**『傳輸協定資料單元』（Transport Protocol Data Unit，TPDU）**，依照順序傳送。對方接收後並組合成原來的 SPDU 再傳給交談層。交談層之間連線的型態，也影響到交談層和傳輸層之間的連線關係，我們以下列幾種連線型態來討論：

(1) 一對一對談關係：兩交談者之間是一對一的對話關係。兩端交談層只要存取到一個傳輸層的通訊鏈路就可以，如圖 6-2 (a) 所示。

(2) 一對多對談關係：一個交談者同時和多個交談者之間的對話（例如三方通訊）。要求連線端的交談層必須連接多個傳輸層的通訊鏈路，每一個通訊鏈路連接一個交談對象。例如一個交談使用者想要同時和五個使用者交談，就必須連接五個傳輸層鏈路，每一條鏈路對應一個交談對象。任何一個傳輸鏈路連接到自己的網路層實體，亦即每一個交談對象可以分散在不同網路上，或不同地區上，如圖 6-2 (b) 所示。

(3) 多對多對談關係：多個交談者同時和多個交談者之間的對話（例如視訊會議）。每一個交談者的交談層必須依照目前所需對話的對象連線，每一個交談對象都必須連接一個傳輸層的通訊鏈路，如圖 6-2 (c) 所示。

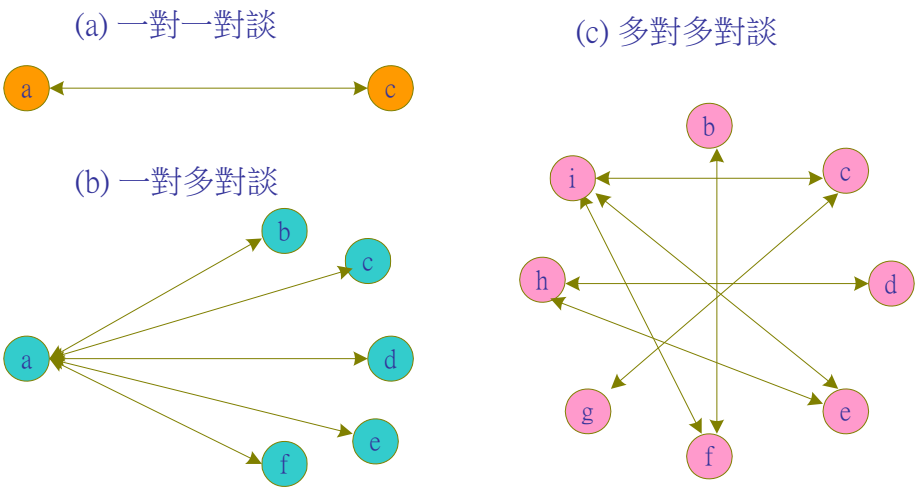

圖 6-2 交談層的連線型態

6-2-2 交談連線之管理

就交談連線管理而言，交談層通訊軟體提供使用者之間相互交談（Interaction）或對話（Dialogue）。對話可能有：雙向同時（Two-way Simultaneous）、雙向交替（Two-way Alternated）和單方向（One-way）等三種對談模式。以下介紹三種對談方式和傳輸層連線之間關係。

(A) 雙向同時對談模式

『雙向同時對談』（Two-way Simultaneous Dialogue）模式是全雙工（Full-Duplex）的交談方式，雙方可同時發送資料。當雙方在交談建立時協議使用此種模式，就不需要特別的對話管理工作，這是最常使用的對話模式。

使用者要求雙向同時交談的情況下，交談層要求連接傳輸層時，必須銜接到兩條傳輸鏈路，一條供傳送使用，另一條則供接收資料使用；連線對方也必須使用兩條傳輸鏈路和傳送端對應，如圖 6-3 (a) 所示。因此雙向同時交談模式成本費用較高。

(B) 單向對談模式

『單方向對談』（One-way Dialogue）模式也不需要特別的對話管理方法，多個使用者之間只能由一個使用者做單向發話，其他使用者只能接收對話。在此需留意，所謂單向對話乃是對傳輸資料而言，在建立交談的過程中，仍需雙方向的交談來達成協議。在資料傳輸過程中，接收方仍需發送確認訊息或其他控制訊息；在某些特殊情況下，接收端也可能發送中斷（Interrupt）訊息，通知發話端暫停傳送資料。一般應用於非交談式的資料傳輸，譬如，遠端列印的情況下，由某一伺服器（如列印伺服器）代為接收列印資料，而只負責接收及儲存資料，並不做資料之處理及回應。當交談層使用者要求單向對談連線時，交談層通訊軟體要求銜接傳輸層時，只要一條鏈路就可以，並指定這條鏈路的傳輸方向。連線對方也是一條鏈路即可，傳輸方向必須由雙方協議而成，如圖 6-3 (b) 所示。

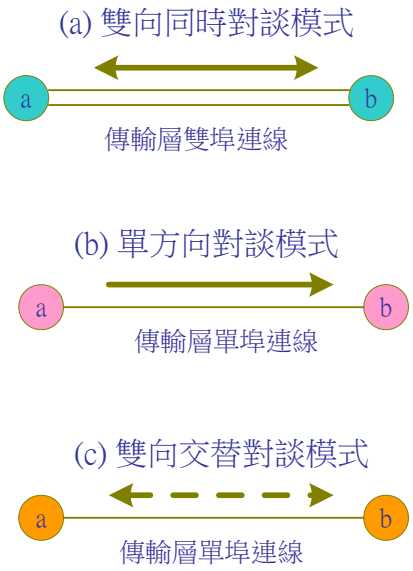

圖 6-3 交談層的對談模式

(C) 雙向交替對談模式

『雙向交替對談』(Two-way Alternated Dialogue) 模式在傳輸層上只要取得一條鏈路即可，對談的雙方輪流使用這條鏈路發話，如圖 6-3 (c) 所示。其運作方式類似無線對講機使用單一頻率，使用者同一時間只能收話或發話，不能同時進行收發。所以雙向交替模式比較複雜，雙方需交替輪流發送資料。

一般我們還是使用**『符記』(Token)** 方式來決定發話的順序，誰取得 Token 就有發話的權利，否則只能收話。運作方式如圖 6-4 所示，我們以下列八個步驟來介紹：

(1) 交易程式 A 要建立對話連線，並表示擁有 Token。

(2) 交易程式 B 回應同意連線，並表示釋放 Token。

(3) 交易程式 A 取得符記，便開始發話 (資料_A)，開始進行傳送。

(4) 交易程式 A 送出釋放 Token 表示發話完畢，將發話權交給對方。

(5) 交易程式 B 取得發話權利 (Token)，並開始發話 (資料_B)

(6) 交易程式 B 將 Token 傳送給 A，表示釋放發話權。

(7) 取得 Token 者 (A) 可要求釋放連線，並將 Token 傳送給對方 (B)

(8) 對方（B）接收到 Token 和釋放連線訊號，便可決定是否同意釋放連線。如沒有資料傳送就發出釋放連線確認訊號；否則繼續發話。

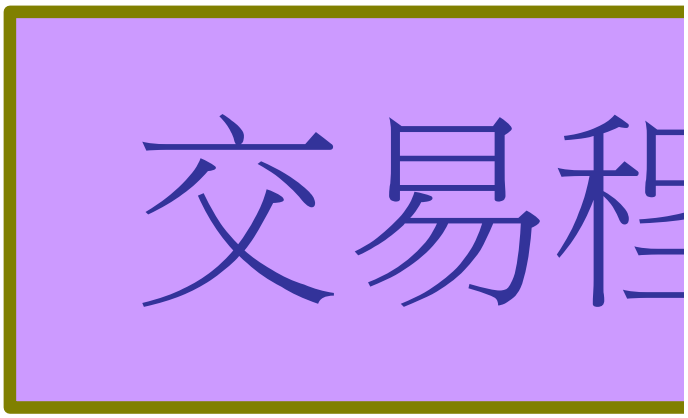

圖 6-4 雙向交替對談模式的運作方式

6-2-3 交談連線之復原

交談層另一個重要的功能是提供**『檢查點』(Checkpoint)**和**『再啟動』(Restart)**的功能來支援**『復原』(Recovery)**的工作。有關復原的工作，我們用一個例子來介紹。假如我們傳送 100 頁的資料到遠端印表機列印時，所有資料在網路中傳送皆正常，但當列印到第 90 頁時發生印表機夾紙，即使我們重新備妥印表機再列印時，如果沒有事先做好適當處理，它將再從第 1 頁開始列印，而非由第 90 頁繼續列印。由這個實例我們可以知道，故障是發生在應用層的交談過程之中，而非網路上傳輸資料時發生錯誤。因此我們有必要在交談過程中插入檢查點要求對方回應，如對方有所回應，則表示已正接收到對話中的檢查點部份。如果發生故障只要回復到最後確認的檢查點即可，而不需要再由起始點開始。

但並非整個對話之中都必須插入檢查點，我們可以將必須特別保護的交談建構成一個**『對話單元』(Dialogue Unit)**，如圖 6-5 (a)，發話端在一個對話單元上分割成若干個**『交談回復單元』(Session Recovery Unit)**。對話開始時插入主同步點，每隔一段交談回復單元後再插入次同步點，又在對話結束時插入一個主同步點。發話端就依照同步點的位置於發話過程中依序插入對話框內。收話端每收到一個同步點，都必須回應確認訊號，表示已安全收話到同步

點的位置。如圖 6-5 (a) 所示，對話中斷在次同步點 t3 和 t4 之間，收話端以回應次同步點 t3，因此重啟動時只要從次同步點 t3 開始對話就可以，不必再由起始點開始。如圖 6-5 (b) 我們可以將一個對話活動分割成若干個對話單元，對於大量資料的傳送效率會比較高。至於遠距離之間的對話，發話端為了提高傳輸效率會連續傳送多個交談回復單元（也就是插入多個次同步點），收話端也並未每一個次同步點都回應確認（類似滑動視窗法運作）。但發話端插入主同步點之後便暫停傳送，等待回應後再傳送；收話端接收到主同步點必須立即回應確認訊息，雙方對話才能繼續。因此對於大量傳送資料，分割成若干個對話單元可保證資料的完整性。

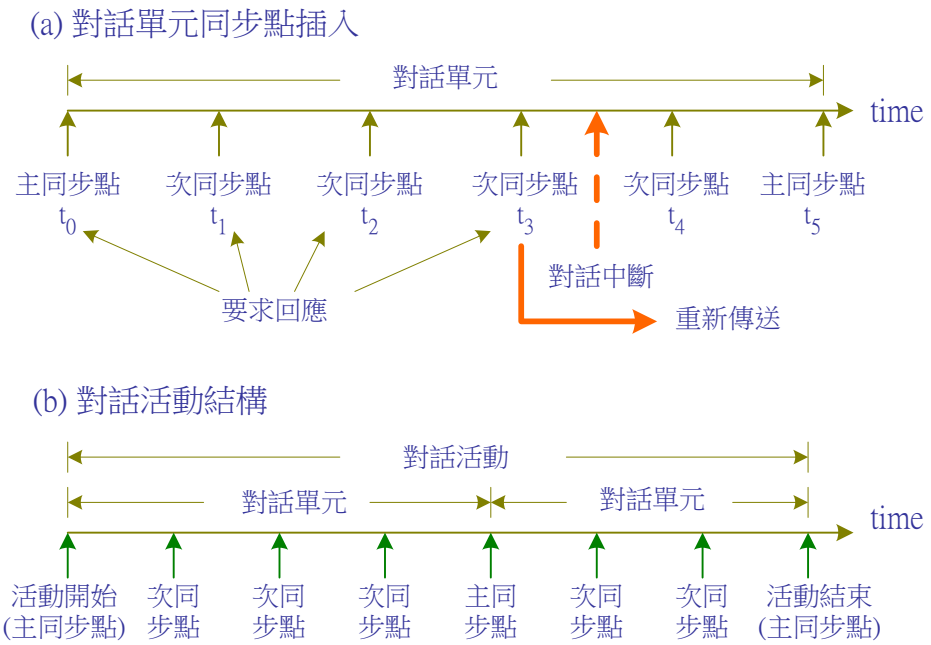

圖 6-5 交談連線的復原程序之處理

6-3表現層簡介

『表現層』(Presentation Layer)的工作是提供應用層獨立的資料格式，包括資料的語法、語意、時序、格式、壓縮（compression）及解壓縮（expansion）、以及資料加密（encryption）及解密（decryption）轉換等功能。使用者兩端必須協議出雙方通訊的資料格式，如圖 6-6 所示。我們用很簡單的敘述來描述表現層的功能如下：傳輸層建立雙方的連線，交談層則於已建立的實體連線上建構交談連線；就如電話雖已接通，但並不表示對話已開始，交談層就如在已接通的電話上建構雙方對話的模式(單工或雙工)和對話連線的種類(單點或多點對話)；表現層就是決定雙方對話中的語言（國語或台語），雙方必須協議出共通的語言，雙方所表示

的含意(應用層)才能被接受。所以表現層的功能就是制定通訊中資料格式的標準，如圖 6-6。本節僅就簡單敘述有關獨立資料格式的觀念，有關資料格式表示法（ASN.1）請參考拙著**『Internet 網路原理與實務』**，至於資料的壓縮和加解密技術，限於篇幅不另敘述，請參考拙著**『Internet 網路安全與實務』**。

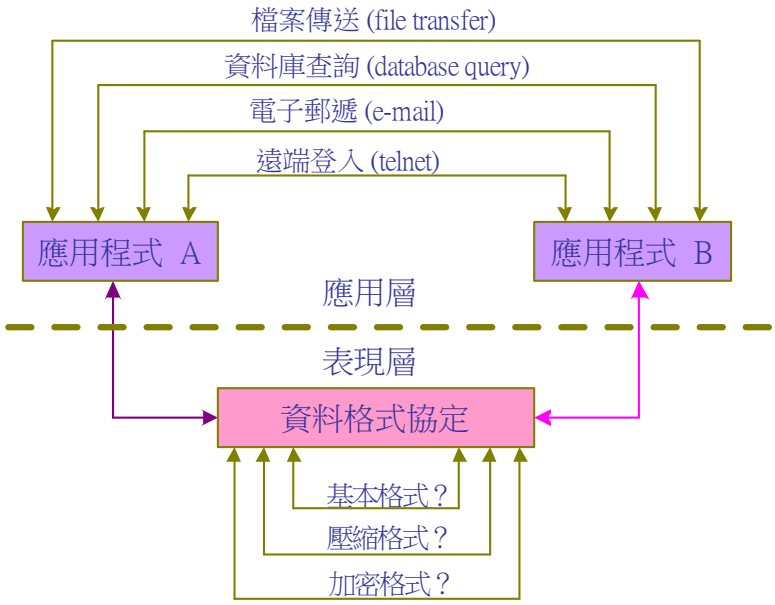

圖 6-6 表現層之功能

並非兩端對談之所有資料格式都必須特殊格式化，而僅對某部份資料作格式化制定，此功能稱之為**『獨立資料格式』**。例如，當您使用電子商務的線上交易機制購買產品，瀏覽網頁上登錄各式樣產品時，已建構完成雙方的對談，這時候用一般格式傳送資料；但當您決定購買某一產品時，且輸入您的信用卡號碼並欲傳送給對方，這時只需要對信用卡號碼加密即可，而非整個對談。獨立資料格式運作如圖 6-7 所示，對談雙方之協議也和通訊連線一樣，於此不再另述。

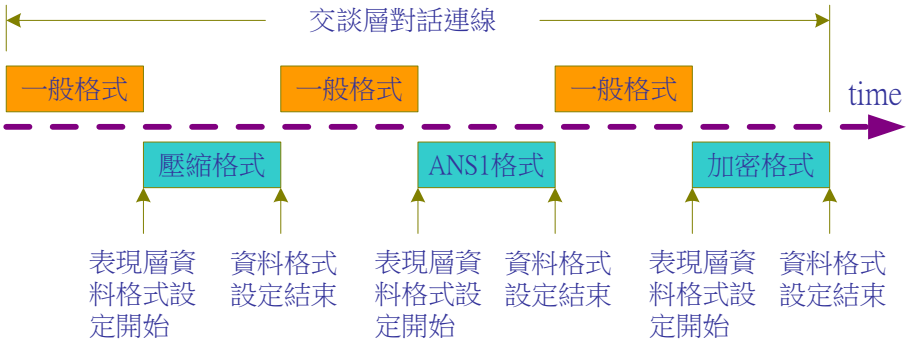

圖 6-7 對談之中的表現層獨立資料格式設定

6-4應用層簡介

『應用層』(Application Layer) 負責提供各種服務給應用程式 （ application processes ），透過網路的連結功能，來達到和其他應用程式交換資料的目的。換言之，應用層提供了使用者（或程式）與網路溝通的介面；有了最高層的應用層，使用者才能存取網路，就像應用軟體所提供的使用者介面一樣。在通訊協定裡，第一到第六層的功能是在敘述如何建構網路，如何透過電腦網路互相通訊，也制定了許多通訊之中應有的技術標準。應用層就是希望藉由在這些網路架構中發展出所需的應用程式，例如檔案傳送通訊協定（ ftp ），及遠端電腦登入（ telnet ）等等。在通訊雙方也需使用同樣的網路應用程式，才可以互相通訊。又如使用者想要遠端登入另一部電腦，雙方必須達成協議，使用相同的登入(telnet)和被登入程式(telneted)，方可正常運作，這也是需要網路層通訊協定的主要原因。

在網路上發展應用程式或者使用應用程式，乃是建構網路最主要的目的，因此應用層扮演了網路上最主要的角色，不論程式設計師或使用者所接觸到的網路環境都是應用層。也就是說，程式設計師透過應用層開發出各式各樣的網路應用程式；使用者透過應用層執行網路應用程式，多采多姿的網路世界就是由應用層開始。以下將介紹應用層的相關技術，讓讀者有一個較完整的概念。

6-5網路作業系統

我們對電腦的『作業系統』(Operating System, OS) 或許有所熟悉，它除了管理該部電腦上所有的硬體外，並提供介面讓使用者操作電腦。如果沒有作業系統，電腦就形同廢物，起不了任何作用。作業系統提供各種程式介面，讓程式設計師可以在電腦上開發應用程式。同樣的，作業系統上可以掛上各公司行號所發展的應用程式，使得作業系統上的資源更加豐富。另一方面，我們必須同時考慮其方便性，因此作業系統必須提供一些非常友善的操作介面，例如視窗介面與多媒體處理介面 (X-windows 或 Windows 系列)。因此，我們可以說一部電腦所提供功能的優劣，完全取決於其使用作業系統為準。

此外，一部電腦通常可安裝不同的作業系統，以目前 Pentium 系列電腦為例，同一部電腦上可以分別安裝 MS-DOS、Windows 98、Windows NT、Windows 2000、Unix、Linux、OS/2 等作業系統，如果安裝 MS-DOS 只能單機單程式使用 (Single-User Single-Task)；安裝

Windows 98 可有視窗介面，圖文並茂，並且是單人/多工系統（Single-User Multi-Task）；安裝 Unix 系統還可達到多人/多工系統（Multi-User Multi-Task）。由此可見作業系統扮演角色的重要性。

同樣的，網路上也必須有一個**『網路作業系統』（Network Operating System, NOS）**，它除了必須具備管理網路上各種通訊行為外，也必須提供各種網路介面，讓程式設計師開發應用程式，以及提供一般使用者的操作介面。所以一個網路的處理能力如何，也是取決於網路作業系統的功能。目前市面上也有許多網路作業系統，例如：Microsoft Network（MSN）、Network File System（NFS）、Netware（Novell）、System Network Architecture（SNA）、ArpaNet。各種網路作業系統提供許多網路應用程式，例如：E-mail、File Server、Web Server、Database Server 等應用。

但是，網路作業系統和電腦作業系統在架構上還是有很大的不同點。一般作業系統管理電腦本身的硬體周邊環境，如磁碟機的存取、螢幕的顯示、鍵盤和滑鼠的輸入等等。當我們在建構網路作業系統時，只要將有關網路的軟硬體裝置在原作業系統上擴充即可。也就是說，網路作業系統是外掛在原作業系統之上，如圖 6-8 所示。

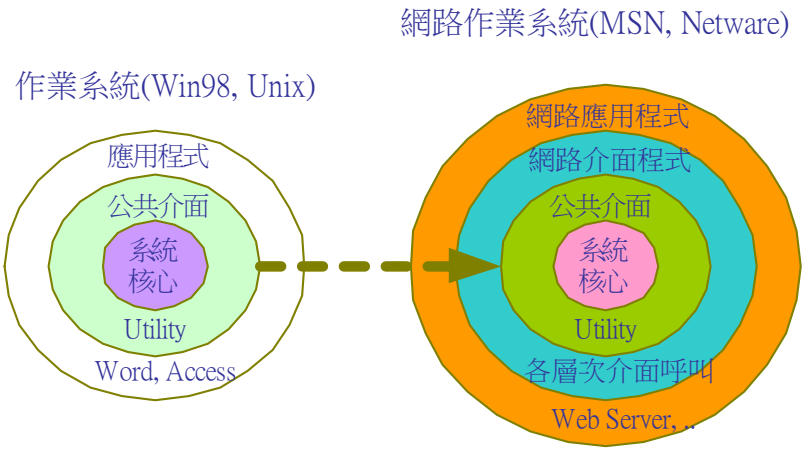

圖 6-8 網路作業系統與電腦作業系統

早期開發網路作業系統也大都採用外掛式的，也就是在現有的作業系統上安裝網路作業系統。至於電腦本身內部的軟硬體處理，還是由本地作業系統（Local Operating System, LOS）處理，對於網路上的通訊行為則由網路作業系統負責。但隨著網路上應用愈來愈複雜，我們需要另一個特殊的作業系統才能符合環境的需求。也有必要發展一個專屬的作業系統來結合

網路作業系統，因此有：Netware File Server、Windows NT server、Windows 2000 Server 的誕生。在這些專屬的作業系統上，當然也可以加掛其它網路作業系統。

一般我們稱 Unix 或 Linux 為網路作業系統，其實它也是一般電腦作業系統，只不過將許多網路處理功能安裝在系統核心（Kernel）上，使對網路的處理能力特別強，所以稱它為網路作業系統。一般我們客戶端（Client）的電腦對網路的處理能力沒有伺服器端那麼複雜，不需要一個專屬的網路作業系統，因此都採用外掛式的網路作業系統，例如，在 Windows 98 上安裝 Microsoft Network 或 TCP/IP 網路（ArpaNet）。

既然網路作業系統是採用外掛式的，不會影響到作業系統本身的處理能力，亦即一部電腦可同時安裝許多套的網路作業系統。譬如，在一部 Windows 98 的電腦上可以安裝多種的網路作業系統，如，Microsoft Network、Novell Netware、SNA、ArpaNet 等等。在同一部電腦上執行不同的應用程式可能會用到不同的網路作業系統，例如，**『網路芳鄰』**是屬於 Microsoft Network；而**『IE 瀏覽器』**則屬於 ArpaNet。

我們瞭解網路作業系統是處理所有通訊行為，也就是說，它包含網路通訊協定的第一層到第七層的通訊協定。但是各家網路廠商所製作的網路作業系統考量到應用的範圍層次不同，也許在各層通訊協定裡所用到的技術也不盡相同。譬如，Microsoft 網路所訴求的對象是範圍較小的區域網路，但希望應用的層次較高；另外，ArpaNet 網路所訴求的功能是較廣域的網路通訊，但網路應用的層次則較簡單。因此，考量到不同的應用，各層次的通訊協定的製作也會不同。在 Microsoft 網路上，第三層和第四層採用 NetBEUI 通訊協定；而 ArpaNet 則採用 TCP/IP 通訊協定。但為了能在不同網路上執行各種網路作業系統，各網路之間也許在某些層次採用相同的通訊協定（如 Ethernet 網路或 ATM 網路），這也是說明通訊協定為何要合乎堆疊性的主要原因。另外，有一個重要的觀念不容忽視，各網路作業系統考慮到應用的層次時，通訊協定裡雖然有七個層次，但並非所有網路作業系統都裝置所有層次。譬如，在 ArpaNet 網路上，就沒有將其功能細分出表現層和交談層，而將其納入應用層之內。

6-5-1 網路介面程式

網路介面程式是提供開發各式各樣的應用程式，程式設計師只要透過介面程式來編寫應用程式，就不需要去考慮網路是如何實現出來的。就像是在一般作業系統上的**『系統呼叫』（System Call）**程式一樣。程式設計者只要呼叫 open() 介面程式，而給予適當的參數

（parameter），它便會指定到磁碟機或其他週邊設備上，並可存取到所需的資料，而無需考慮到磁碟機和其他週邊設備如何驅動馬達和磁軌。

一般網路作業系統為了方便程式開發，除了提供應用層介面程式，也提供其他層次的介面程式。這些介面程式就如 1-6-1 節裡所介紹的上下層介面程式一樣。所以，發展應用程式不一定是由應用層發展。程式設計師可依照自己的需要，呼叫各層次的介面程式，這樣發展的應用程式的變化性就較大，更能滿足不同的需求。

我們用 TCP 的程式介面來說明，如何利用介面程式開發應用系統的概念（有興趣在網路上編寫程式的讀者，請參考拙著**『Internet 網路原理與實務』**）。例如，我們希望開發一個檔案伺服器，讓網路上的使用者可以儲存和讀取檔案。其程式架構如圖 6-9 所示。在這個程式裡，我們會用到下列 TCP 的介面程式（又稱系統呼叫）：

(1) socket()：開啟 TCP 通訊服務點。

(2) bind()：對 socket() 定址，連結至相對應的埠口位址（port number）。

(3) listen()：設定 socket 為等待狀態，等待 Client 端要求連線。

(4) connect()：要求和對方通訊端（socket）連線。

(5) accept()：接受對方（socket）連線要求。

(6) write()：將資料寫入連線中的 socket，傳送到通訊對方。

(7) read()：由連線的 socket() 中讀取資料。

(8) close()：釋放 socket，中斷連線。

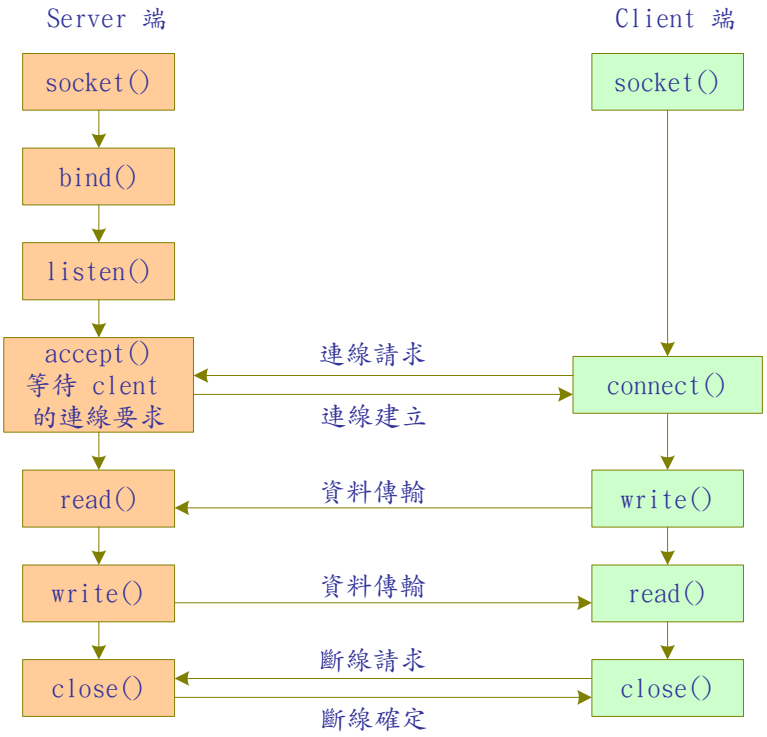

圖 6-9 檔案伺服器程式架構

由上述簡單的例子，我們可以瞭解如何在傳輸層上開發應用層的應用程式，因為它們之間並沒有交談層，所以通訊方式就依照傳輸層介面 (socket()) 所提供的方式 (全雙工方式)；也沒有表現層，所以傳輸之間的資料格式沒有特殊定義和管理，只能用一般資料型態(ASCII) 傳送。如果有一端傳送特殊的資料格式，對方可能無法辨識。但該程式只做資料的存放和讀取，伺服器端並未做資料的處理，所以對資料格式也不必特別處理。由這個簡單的例子，我們也可以瞭解並非所有應用程式都需要七個通訊協定，但第一、二、七層 (實體層、資料連結層、應用層) 缺一不可。

6-5-2 網路命令程式

如同一般作業系統，網路作業系統必須提供一些有關網路管理的命令程式 (command 或 utility)，讓使用者方便管理或操作網路作業系統。我們可以透過命令程式來管理整個網路環境，也可以瞭解網路運作的情況。各種網路作業系統所提供的命令或有不同，以下我們就一些 Internet 網路上命令程式來介紹，如果讀者希望有較完整之命令格式的使用方法請參考相關書籍。

(1) ping：測試網路位置是否正常。

ping 163.15.2.1 （測試 163.15.2.1 電腦位置是否正常）

(2) finger：查詢網路上使用者的相關資料。

finger （顯示在網路上所有的使用者）

finger root@linux-1 （查詢遠端電腦 linux-1 上 root 使用者的資訊）

(3) ifconfig：設定網路介面（interface）環境，如 IP 位置、網路遮罩等等。

ifconfig –a （查閱所有網路介面設定狀態）

ifconfig eth0 163.15.2.62 （設定 eth0 網路介面的 IP 位置）

ifconfig eth0 broadcast 163.15.2.255 （設定 eth0 的廣播位置）

ifconfig eth0 netmask 255.255.0.0 （設定網路遮罩，class B）

(4) route：設定路由器之靜態路徑選擇（static routing）路徑。

route （查閱所有靜態路徑選擇表的設定）

route add –net 163.15.3.0 gw 163.15.2.62（設定網路位置的網路閘出口）

route add default gw 163.15.2.63 （設定 default 的網路閘的位置）

route del 163.15.2.61 （刪除路徑設定）

(5) netstat：查閱網路組態和網路狀態。

netstat –nr （顯示路徑分派的情形）

netstat –ia （顯示網路介面的統計資料）

(6) traceroute：可追蹤路徑選擇所經過的路徑。

traceroute www.nsysu.edu.tw （追蹤到中山大學網站所經過的路徑）

6-5-3 網路應用程式

建構網路的主要目的是要透過網路執行某些應用。不同的網路作業系統所訴求的應用範圍，和其所提供的網路應用程式一不相同。任何人都可以利用網路介面程式開發其所需的網路應用程式，所以網路應用程式的變化性非常的大。一般我們會將專屬的應用安裝在特殊的主機上以提高它的處理能力，因此就有所謂**『主從式架構』(Client/Server Architecture)**的誕生。因此網路應用程式可歸類為下列兩種：

(1) 伺服端 (Server): 在網路上提供服務的程式，又稱為網路伺服器。依照它所提供的服務如下列：(簡列一些實例)

- Mail Server：提供信件服務。
- FTP Server：提供檔案傳送服務。
- Print Server：提供檔案列印的服務。
- Database Server：提供資料庫系統讓使用者查詢的服務。
- Web Server：提供網頁讓使用者瀏覽的服務。

(2) 客戶端 (Client): 在網路上使用伺服器所提供服務之應用程式。

- internet explore (IE)：使用 Web Server 上服務的應用程式(瀏覽器)。
- mail：使用 mail server 上服務的應用程式。
- ftp：使用 ftp server 上服務的應用程式。

就一部電腦而言，執行了 Server 程式時，它就是一部伺服器，當然也可以執行 Client 程式，因此也是客戶端。一般而言，任何一部主機電腦可同時執行若干個 Server 程式，而成為多個伺服器的提供者。但我們為了提高執行效率，都會將較常用的 Server 程式安裝在某一特殊主機電腦上，如圖 6-10 所示。還有一些在 Internet 網路上較常用的應用程式，簡單介紹如下。

- **telnet**：遠端登入程式。
  - # telnet linux-1 (登入到遠端電腦 linux-1 上)
- **ftp**：檔案傳送程式。

- # ftp linux-1.cma.edu.tw （由遠端電腦 linux-1 上作檔案傳輸）

- **rcp**：遠端複製程式。

- **rsh**：執行遠端電腦上程式。
  - # rsh linux-1 cat /etc/passwd （執行遠端電腦 linux-1 上的程式 cat /etc/passwd）

- **mail**：傳送信件到遠端電腦上使用者。
  - # mail tsnien@cc.cma.edu.tw < letter （將 letter 信件傳送給 tsnien 使用者）

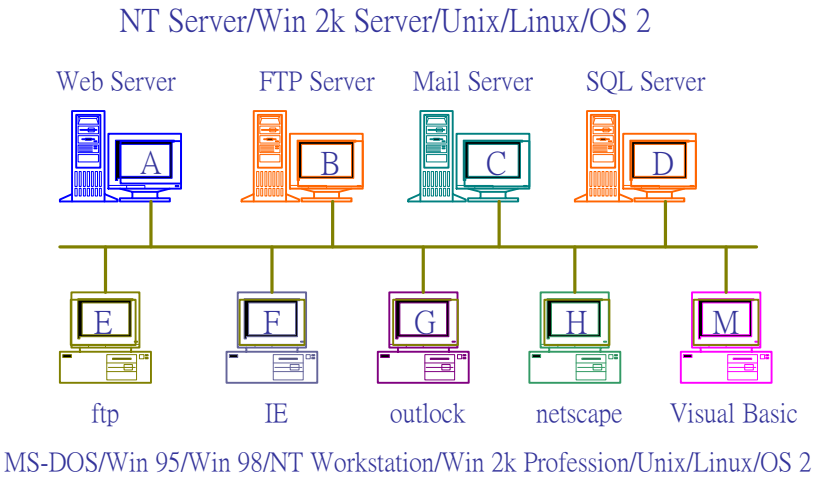

圖 6-10 Client/Server 架構

6-6分散式處理架構

製作網路的目的，除了透過網路達到遠端通訊的目的外，我們最期望能整合多部的電腦來共同完成所交代的工作。早期網路技術尚未成熟時，人們為了提高電腦的處理能力，只能一昧的提高電腦硬體設備，例如，單處理機系統變成多處理機系統、增加記憶體空間、或提高軟體設計能力，以減少執行時間，也因此造成主機電腦愈來愈昂貴。網路技術較成熟之後，人們漸漸思考如何讓多部電腦透過網路，共同來處理工作，不但可以提高執行速度，也可以提高可靠度。尤其近年來微處理機的技術大大提昇，微型電腦的處理能力也不遜於迷你型電腦。如果我們能整合多部的微型電腦，其處理能力或許能高於早期的主機電腦，而且價格也較便宜。倘若這個構想行得通，日後電腦處理能力有所不足時，只需添購微型電腦加入即可，無需重新更換主機電腦，既節省成本又富擴充性，這就是**『分散式處理』**概念的誕生。

『分散式處理』(Distributed Computing) 除了是整合多部電腦共同處理外，我們還有一個基本理念就是：

『在分散式處理系統環境裡，使用者可以在它們之間的任何一部電腦下達所要的命令，使用者將不知道、也不需要知道，到底是哪一部電腦執行該命令；使用者將不知道，也不需知道，他所需的資源是存放在哪一部電腦上』。

為了實現上述的理想，首先，我們將工作性質和所需的資源組織成若干個**『作業環境』(Working Environment)**，再將這些作業環境分配給網路上多部電腦共同來執行，以達到分散式處理的目的。如圖 6-11 表示，一般我們將作業環境區分如下：

(1) 資料檔案：是指資料儲存設備上的資料結構，也就是一般的檔案管理系統。

(2) 資料庫系統：提供邏輯性的資料結構管理系統，也就是所謂資料庫管理系統。如：Informix 或 Oracle 等等。

(3) 應用作業：依照應用環境需要所製作的應用程式。例如：辦公室自動化系統、帳務管理系統、銷售管理系統等等。

(4) 表現作業：提供使用者輸入和輸出的人機介面，以及相關報表的格式製作環境。

一般傳統的系統就是將所有作業環境都由主機電腦執行，客戶端電腦只從事於資料的顯示和輸入而已。亦是，有關資料的儲存、查詢、運算處理的工作都是由主機電腦上執行，如圖 6-11 所示。分散式處理的製作方法就是將上述的作業環境分散到其他電腦上共同處理，我們依照工作分散的情形可區分為不同的分散處理架構，各種處理架構有其優缺點及應用範圍，如圖 6-12 所示。使用者可依照自己環境的需求架設所需的分散式處理架構，以下就各種處理架構的特性加以介紹。

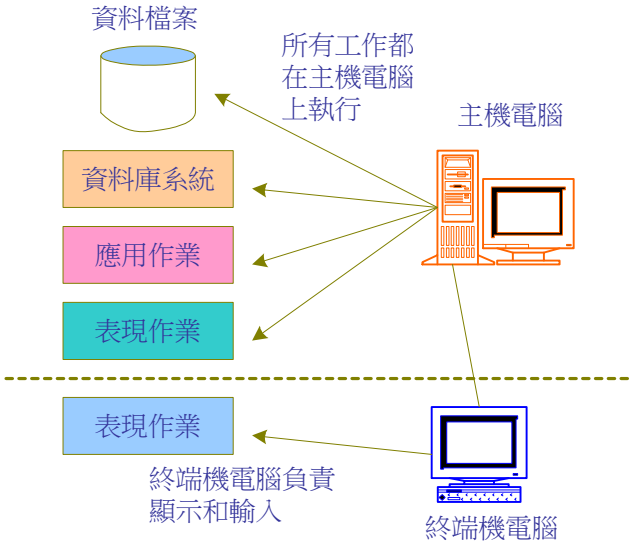

圖 6-11 主機電腦為主的資料處理

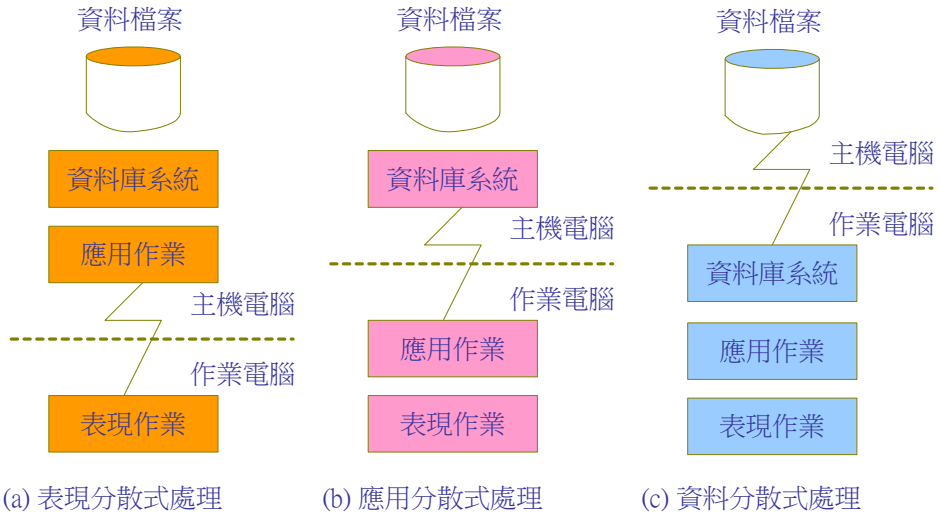

圖 6-12 分散式處理方式

6-6-1 表現分散處理架構

表現分散處理的環境裡，是將所有的應用軟體、資料庫系統、檔案系統、以及磁碟儲存系統都集中在主機電腦上執行，作業端的電腦只處理表現作業，如圖 6-12 (a) 所示。一般輸入和輸出的人機介面處理，例如，輸出入視窗的製作、下拉式命令操作，則交由作業端電腦處理。主機電腦只要將表現的資料傳送給作業電腦，作業電腦就依照自己環境的需求製作輸出介面，將主機電腦的資料顯示出來。因此主機電腦只要定義標準資料格式，再傳送給作業電腦，作業電腦就可以跨越不同的電腦環境下作業。尤其在輸出表格的製作方面，一般來說比較為費時費力，而且依個人需求的變化性較大，則直接交給作業電腦端處理最適合。

既然所有的資料處理和應用程式都在主機電腦上執行，除了資料完整性的管理較佳外，主機電腦可以隨時更新或備份資料，而且應用軟體更新或新購也只要在主機電腦上安裝或修改即可。就整個電腦應用環境而言比較容易管理，架構也較堅固，因此，目前各組織單位、或公司行號以此類系統架構為大宗。譬如，一般的醫院管理系統、銀行管理系統、銷售管理系統等等，都是這種架構。然而此架構也存在某些缺點，因為大部分的工作都集中在主機電腦上執行，作業端電腦上只執行管理資料無關的表現作業，無形中加重主機電腦的負荷，當工作量或上線人數持續增加時，主機電腦可能應付不來，處理速度會降得很厲害，如此非擴充或更換主機電腦不可，但礙於主機電腦的擴充性及成本考量，恐怕不是那麼輕易可以達到。

6-6-2 應用分散處理架構

時下作業端（client）的電腦處理能力愈來愈強（例如 Pentium 系列電腦），而且一般工作人員對電腦的處理能力也增加，因此希望使用者自行開發系統（End User Computing, EUC）。資料庫系統也提供許多程式設計的工具，例如第四代語言（4 Generation Language, 4GL）的開發工具。因此，在種種環境因素的變遷下，我們希望降低主機電腦的工作負荷，逐步將許多工作轉移到作業端電腦上執行，於是有所謂應用分散處理架構的誕生，如圖 6-12 (b) 所示。

應用分散處理架構就是目前時尚非常流行的**『主從式架構』（Client/Server architecture）**。資料庫系統安裝於主機電腦上，所有相關資料都儲存於資料系統裡，稱之為**『資料庫伺服器』（Database Server）**，使用者可以在作業電腦上開發應用程式，透過標準介面到資料庫伺服器上存取資料。這種作業方式不但可以解決資料一致性的問題，也可以提高執行效率。例如，在一個銀行系統裡，所有客戶資料有數十萬筆，在作業電腦上想要查詢某一客戶的資料，於是傳送一個 SQL 查詢命令給資料庫伺服器，資料庫伺服器查詢後，便將該客戶資料傳送給作業電腦。因此網路上只傳送查詢得到的結果，而不需要將整個客戶資料（數十萬筆）傳給作業電腦，對整個處理速度來講可提高許多。

客戶端（client）可以依照自己的需求開發應用程式並執行之，如此可以減低主機電腦的負荷。如圖 6-13 所示，客戶端電腦不一定要固定到某一資料庫伺服器上存取資料，而可以同時連接到其他伺服器上存取資料。如果從資料管理的觀點來看，表示對整個資料的存放不需要儲存在固定一部伺服器上，而可以依照資料的特性，分散儲存於不同的地方。對客戶端

電腦而言，存取本機電腦上資料或其他伺服器上資料都沒特殊不同的地方，因此可以達到分散式處理的精神：**『使用者不知道、也不需知道，他所需的資源是存放在哪一部電腦上』**，這就是**『分散式資料庫系統』（Distributed Database System）**的基本架構。目前大部分的資料庫系統廠商都有提供資料庫伺服器（如 SQL server）的產品，也提供相關的發展軟體（如 Visual Basic）讓使用者能快速開發應用程式，因此有許多應用系統都使用這種環境發展。

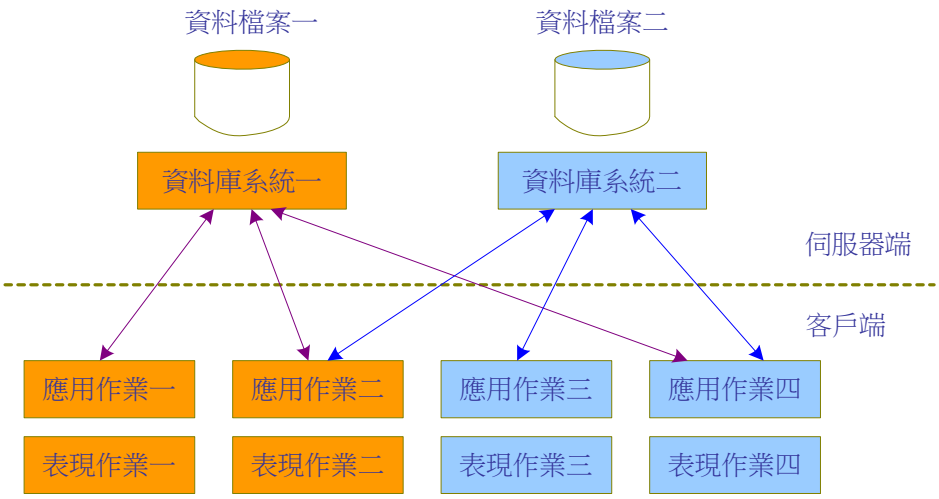

圖 6-13 分散式資料庫系統架構圖

6-6-3 資料分散處理架構

如果我們更進一步將資料檔案放置於主機電腦（或稱伺服器端），而將其他資料處理環境安裝在作業電腦（或稱客戶端）上，就是所謂的**『資料分散處理架構』**，如圖 6-12 (c) 所示。此架構的伺服器端就稱之為**『檔案伺服器』（File Server）**，檔案伺服器提供邏輯性的檔案系統，讓不同電腦的客戶端存取，使用者不必去考慮檔案系統真正儲存在磁碟機上的資料結構。當存取資料時，不論檔案是來自本地電腦或來自遠端檔案伺服器，對使用者而言並無不同。這種系統架構是網路作業系統所提供的基本應用，如 Novell Netware 的檔案伺服器、Microsoft Network 上的網路芳鄰、Sun 的網路檔案系統（Network File System, NFS）都是以資料分散處理為基本架構。

檔案伺服器提供跨越不同平台的檔案儲存結構，也就是說，檔案伺服器上所提供的檔案服務可以讓不同作業系統之電腦存取。例如一部檔案伺服器，可以讓 MS-DOS、Windows 98、Unix 電腦存取檔案。我們也可以在客戶端安裝資料庫系統，對資料庫系統而言，它的檔案結構可以儲存在不同的檔案伺服器上，也很容易整合各地區的資料檔案。

在客戶端安裝資料庫系統，而資料檔案儲存於檔案伺服器上的架構，雖然早期網路上應用大多是這種架構，但它的處理效率似乎非常不理想。我們以圖 6-14 (b) 來說明其情況，在一個銀行系統裡有數十萬個客戶，使用者在客戶端電腦（電腦 B）上輸入查詢某一個客戶資料，因為所有資料檔案都儲存在檔案伺服器（電腦 A）上，所以檔案伺服器必須將數十萬筆的資料透過網路傳送到客戶端（電腦 B），再由客戶端上的資料庫查詢功能，尋找到所需要的資料。

因此，網路上傳送資料的傳輸量就非常高，但如果有多位操作者同時輸入查詢客戶資料，那整個系統的處理效率會變得非常差。而且檔案伺服器上架設資料庫系統，管理者和資料儲存在不同工作平台上，對資料一致性的問題也很難克服。同樣的情況下，如果以資料庫伺服器架構（圖 6-14 (a)），只需傳回 5 筆資料即可，因其查詢工作是由資料庫伺服器（主機 A）負責，而且對於資料的一致性問題也比較容易控制。所以，檔案伺服器只提供跨越不同平台的檔案共享服務，不提供其他特殊資料的處理。

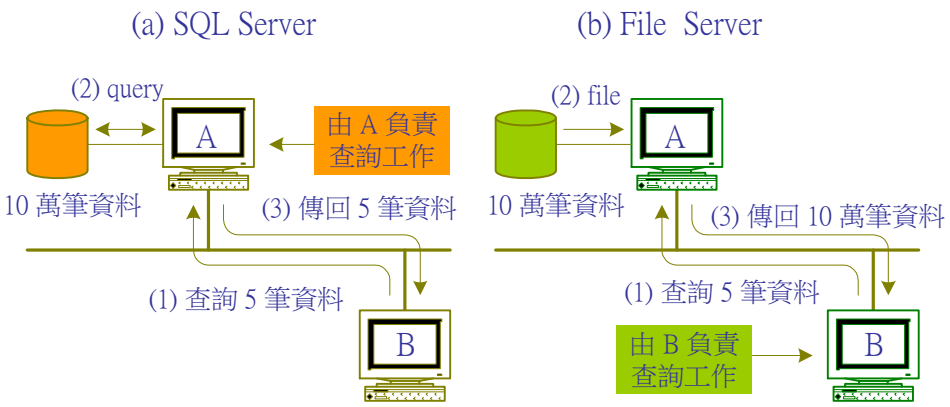

圖 6-14 SQL Server 與 File Server 資料查詢的比較

6-6-4 磁碟分散處理架構

主機電腦（或稱伺服器）上提供磁碟機實際儲存（Physical Storage）的空間讓客戶端電腦儲存資料，稱之為**『磁碟分散處理架構』**，該伺服器也稱為**『磁碟伺服器』（Disk Server）**，如圖 6-15 所示。磁碟伺服器上只提供磁碟空間，並不提供檔案系統的資料結構管理服務，該工作必須由客戶端上電腦自行處理，因此伺服器端的處理工作就比較簡單。早期磁碟機非常昂貴，並非所有電腦上都裝有磁碟機，我們可以購買一部磁碟機讓多部電腦共用，以節省費用。而且在客戶端上，可安裝遠端啟動程式（Remote Booting），它的作業系統程式可放置在磁碟伺服器上，和本地磁碟機的操作沒有兩樣。如圖 6-15 中， 磁碟伺服器（主機 A）提供

磁碟空間讓主機 B（MS-DOS）、C（Unix-1）、D（Unix-2）、E（Unix-3）使用，客戶端電腦都沒有裝設磁碟機，它們的系統啟動之作業系統，以及有關的程式和資料都儲存於磁碟伺服器上（主機 A）。

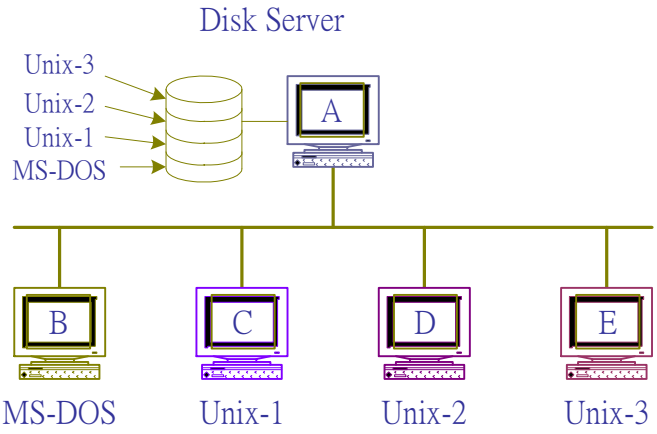

圖 6-15 磁碟分散處理架構

其實，由上述四種分散處理架構，也僅能達到一半的分散式處理功能，而另一半：**『使用者可以在它們之間的任何一部電腦下達所要的命令，使用者將不知道、也不需要知道，到底是哪一部電腦執行該命令』**並沒有真正達成。至於如何達成這方面的功能，讀者可參考有關**『分散式作業系統』（Distributed Operating System）**書籍，本書限於篇幅無法一一介紹，也請見諒。

6-7 Web-based 網路應用

由上一節我們可以大略瞭解，在電腦網路上比較高層次的應用，還是以應用分散處理環境（如圖 6-12 (b)）中的主從式架構較適當。也就是說，資料庫伺服器所提供的服務，最合乎各種應用需求。然而在應用分散環境裡，應用程式存放於客戶端（client）電腦，也在客戶端電腦上執行。提供服務者只能控制資料部分，無法控制客戶端上的程式。如果提供服務者有意要修改或提供另一種服務，必須到所有客戶端安裝新的程式，對於較大的應用環境裡，可能不容易做到。如果，我們將應用程式以檔案伺服器的方法，存放於伺服器上，讓客戶端隨時下載安裝或更新。這也許會出現下列問題：

- 所下載的程式必定是執行檔案，無法跨越不同電腦上執行。
- 每部電腦上環境不一定相同，因此下載的程式不一定可以執行。

- 一般執行檔案都非常龐大，下載不容易。
- 客戶端也許不知道要下載新程式，而執行舊程式可能發生錯誤。

由上述的分析，在主從式的架構裡，將應用程式儲存於檔案伺服器上讓客戶端下載是不大可行的辦法。

如果，我們能將原始程式下載到客戶端，讓客戶端電腦重新翻譯再執行，也許是一種解決的辦法。但是我們要設計一種程式讓不同工作平台都能翻譯執行也是不簡單。所以，我們若在各種不同的工作平台上安裝一套共通軟體，這套軟體專門負責接收伺服器傳來的原始檔案，並且可以翻譯執行，就能解決所有的問題，這套軟體就是目前最流行的『**瀏覽器**』(**Browser**)。其實瀏覽器是一個非常複雜的程式，一個瀏覽器有不同的版本，可以安裝在不同電腦上，但可以接收相同的原始程式並翻譯執行。早期瀏覽器只能接收顯示文字格式 (如 Mosaic)，目前瀏覽器大多能顯示聲音、影像、和文字 (如 Netscape、Microsoft IE)，成為多采多姿的世界。另外，我們也必須制定一個標準的程式語言，這就是 HTML 語言 (HyperText Markup Language)，而伺服器與客戶端電腦之間的傳輸協定是 HTTP 協定 (HyperText Transport Protocol)。儲存這原始檔案讓客戶端電腦下載執行的主機電腦，稱為網頁伺服器(Web server)，神秘又美妙的全球資訊網 (World Wide Web, WWW) 之網路世界，就是由 Web-based 網路應用架構開始。以下我們來看看 WWW 發展的過程：

- 1989 年瑞士日內瓦的歐洲粒子物理實驗室(European Laboratory for Particle Physics) 建立整合性資訊系統 (CERN)，提出 WWW 的構想。
- 1990 年 11 月在 NeXT 系統上開發出 WWW 雛型，12 月開發出文字模式的瀏覽程式。
- 1993 年 4 月 NCSA (The National Center for Supercomputing Applications)發表 Mosaic (文字模式)。
- 1994 年起，WWW 開始了第二次多媒體革命，導入多媒體技術。
- 1994 年 7 月 Netscape 進入 WWW 市場 (多媒體模式)。
- 1995 年 6 月 Sun-soft Java Applet 增添文字以外的動態美感。

- 1995 年 12 月 Oracle 開發出 Web 伺服器連接資料庫技術。
- 1996 年 3 月 Microsoft 發表 Active X 架構。

我們可以發現網際網路在短短的十年功夫，就改變了整個網路世界，也成功了連接數千萬台電腦，和電話連結演進的時效相比，真是不可思議。而且網際網路的應用又不停的進步，除了電子商務外，我們相信他將會進入更深入的應用（如 Intranet 或 Extranet 的辦公室自動化環境）。本節裡我們就來介紹 Web-based 所應用到的相關技術。

6-7-1 全球資訊網（WWW）簡介

圖 6-16 表示**『全球資訊網』（World Wide Web, WWW）**的簡介。WWW 也是主從式架構，伺服器端（Web server）提供資源（HTML 文件）讓客戶端（瀏覽器）使用。它們之間傳輸使用的是 HTTP 通訊協定。伺服器端的定址方法是 URL 定址，客戶端可以依照 URL 位址找到所要的網站，所以這個地址又稱為**『網址』**。我們分別簡述其功能如下：

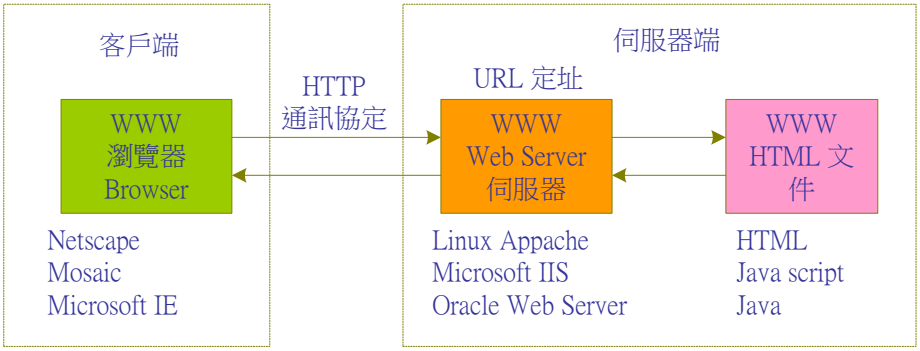

圖 6-16 全球資訊網簡介

(1) 客戶端：客戶端就是瀏覽器（如 IE 或 Netscape），接收伺服器端的 HTML 文件再執行。這個文件型態就稱為頁（page），也稱之為**『網頁』**。所以，客戶端以顯示大量文件（或網頁）為主要工作。每一網頁上的文字或圖樣可以指向其他相關頁之間的連結，頁和頁之間的連結可以無止境的延伸，此連結方法就稱為**『超鏈結文件』（HyperText）**。不僅可以連結網頁，還可以在網頁上任何文字或圖樣設定連結到其他網站，稱為**『超連結』（Hyperlink）**。因此，在客戶端上可以行走全世界任何一個網站，觀看網站上的網頁，所以稱之為**『瀏覽器』（Browser）**。

基本上瀏覽器只能觀賞文字或圖形，如此並不能滿足使用者的需求，我們希望能隨使用者的喜好插入（plug-in）其他程式，以增加瀏覽器的功能。例如，可插入 Java applet 程式，在瀏覽器上即可以執行 Java 所開發的程式，不但可以整合多媒體（文字、聲音、影像），還可以表現動畫，提高網站的可看性。

(2) 伺服器端：伺服器是儲存 HTML 文件，讓瀏覽器下載執行。將 HTML 文件顯示在瀏覽器上就稱為網頁（page），因此該伺服器就稱為**『網頁伺服器』（Web Server）**。因為它和客戶端之間是以 HTTP 通訊協定溝通，又稱為**『HTTP 伺服器』（HTTP server）**。HTTP 伺服器大多架設在 TCP 上的第 80 埠口。

(3) URL：『一致資源定址』（Uniform Resource Locators, URL）是指對每一個網頁能完全定義出它的位址和使用的通訊協定。當我們使用超連結時，設定某一文字片段或圖形指到其他網站上的網頁時，必須標明三件事情：(a) 連接該網站使用何種通訊協定（http 或 ftp）；(b) 網站位址在哪裡（主機的 DNS 名稱）；(c) 該網頁的檔案名稱。例如：

http://www.tsnien.idv.tw/index.html

這 URL 的三個部分是：通訊協定（http）、主機位址（www.cma.edu.tw/）、網頁的檔案名稱（index.html）。通訊協定有：http（超連結文件，HTML）、ftp（FTP 檔案傳輸協定）、file（本地檔案）、news（新聞文章）、gopher（Gopher 文件協定）、mailto（傳送郵件協定）。

6-7-2 HTTP 傳輸協定

『超文字傳輸協定』（HyperText Transfer Protocol, HTTP）是針對 Web 設計的傳輸協定。伺服器和客戶端之間交談都是由 ASCII 方式請求，和 RFC 1341 MIME（Multipurpose Internet Mail Extensions）相似的回應所組成。一般是架設在 TCP 協定上的應用程式。ASCII 方式請求類似一般指令（Command）格式得下達命令，每一個命令格式有其存取方法。MIME 是將網頁（或程式）定義標準化格式，網頁的包裝格式分為表頭（Header）和主體（Body）。一般表頭包括網頁格式（HTML）、網頁頁頭、網頁標題、網頁主題及粗體文字等等（依照 HTML 定義）。主體是指某一個表頭的內容。ASCII 命令配合 MIME 的網頁格式，這就是 HTTP 傳輸協定存取的方法，它包含下列幾種基本命令：

- **GET**：請求讀取網頁。瀏覽器以 GET 命令，請求伺服器送出網頁，以 MIME 方式編碼。
- **HEAD**：請求讀取網頁的檔頭（header）。客戶端僅請求訊息檔頭，而非實際網頁。這方法可取得網頁最後修改時間，可用在建立或測試 URL 的有效性。
- **PUT**：請求儲存網頁。客戶端寫入網頁，提供客戶端建立網頁的功能。
- **POST**：附加一個名稱資源。客戶端將資料附加到某一資源的資料之後。
- **DELETE**：刪除網頁。客戶端要求刪除某一網頁。
- **LINK**：建立超連結。客戶端要求加入超連結。
- **UNLINK**：刪除超連結。客戶端要求刪除超連結。

客戶端以上述的命令要求伺服器端工作，每次請求都有回應執行狀態，狀態代碼有：200（執行正確）、304（執行錯誤）、400（錯誤請求）、403（請求禁止）。

6-7-3 HTML 語言

早期設計網頁瀏覽器的觀念非常簡單，僅希望整合不同文書處理的工具，並於共同的瀏覽器上顯示。不同的文書處理有：MS-Word、vi、PE2 等等，各種文書處理的文件都有其資料表示方式。例如，在 MS-Word 上的文書儲存方式，對於某一段文字的大小或粗細體（如 14 號字及粗體），會在這一段文字的頭尾加入標記符號。文章被開啟時，再依照標記符號將文字型態顯示在螢幕上。網頁的設計就是利用這種觀念，希望在文件上，以標準的標記符號來標示，使其能在不同工作平台上的瀏覽器開啟。這個標準文件標記語言就是**『超文字標記語言』（HyperText Markup Language, HTML）**。

HTML 為 ISO 8879，SGML（Standard Generalized Markup Language）的一種應用，設計使用於超連結文件並為 Web 所採用。如前所述，HTML 是一種標記（markup）語言，即是用於描述文件格式的語言。**『標記』（markup）**是放置文件之中，告知排版軟體顯示文件的格式。例如，在 HTML 中，<B> 表示開始粗體模式，</B> 表示離開粗體模式。以記號語言所寫的文件和 WYSIWYG(What You See Is What You Get)文書處理器產生的文件非常類似，如 MS-Word 一樣。一些較常用的標記如表 6-1 所示。

表 6-1 HTML 語言之標記

標　　記	描　　述
<HTML> ….. </HTML>	宣告網頁將以 HTML 編寫
<HEAD> …. </HEAD>	定義網頁的檔頭
<TITLE> … </TITLE>	定義標題（並不在網頁上顯示）
<BODY> … </BODY>	框註內為網頁主體
<Hn> … </Hn>	n=1~6，框註內六個階層的標題字大小
<B> … </B>	設定框註內文字為粗體
<I> …. </I>	設定框註內文字為斜體字
<UL> … </UL>	框註內為無序串列（註標式）
<OL> … </OL>	框註內為編號串列
<MENU> … </MENU>	框註內<LI>項目行成選單
<LI>	串列項目的開始（並無 </LI>）
 	強迫分離
<P>	區段開始
<HR>	水平線
<PRE> … </PRE>	已格式化文字
<IMG SRC= “…”>	在此載入影像圖形
<A HREF=”…”> … </A>	定義超連結

我們用圖 6-17 的例子來說明 HTML 程式的編寫方式。HTML 編寫程式相當類似於 MS-Word 的文件儲存格式，瀏覽器再依照文件上的標記來顯示。反過來，如果我們使用類似 MS-Word 的文件編輯方式，將於 MS-Word 上編輯 (如 WYSIWYG 方式) 的文件，儲存時再依照標記文件方式儲存。藉由此概念，使用一般文件編輯方式來設計網頁，儲存時再依照 HTML 語言型態儲存，就可節省需多程式設計的時間，這個網頁編輯工具就是類似 FrontPage 的網頁開發工具。

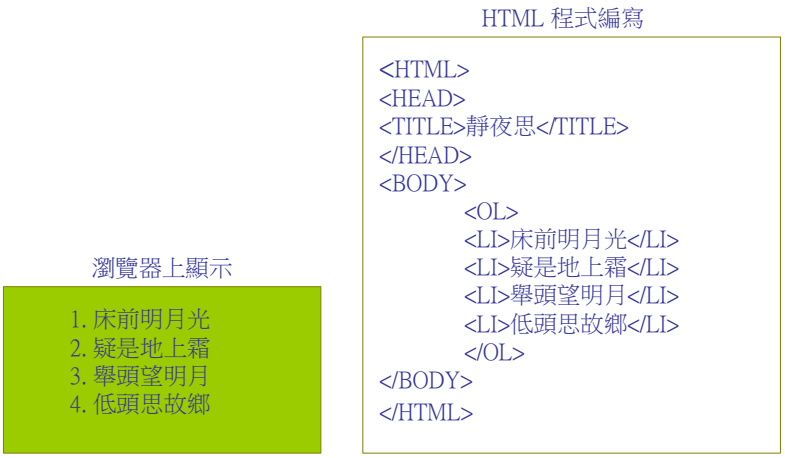

圖 6-17 HTML 編寫方式

6-8Web-based 網路應用架構

隨著網站伺服器的流行，Web-based 網路應用架構也隨著環境的需求，不斷進步與演變，以下我們就目前發展的三個基本架構來加以說明。

6-8-1 靜態網頁伺服器

『靜態網頁伺服器』(Static Web Server) 只提供標準文件格式，讓不同工作平台上的瀏覽器顯示網頁。這網頁可以包含文字、影像、聲音的多媒體，甚至可以達到動畫或播放視訊、音訊等功能。然而，並非所有瀏覽器都具備這些功能，而是依照使用者的需求，在網站上下載 (Plug-in) 各種播放程式 (如，ActiveX、Flash、Medial Play、MP3) 以增加瀏覽器的功能。基本上，靜態網頁伺服器系統是將網頁內容置放於網站內，客戶端透過網路將網頁內容下載到瀏覽器上執行並顯示出來，顯示內容可能是文件、音訊或視訊等等。它們之間的動作是單方向的，客戶端無法傳遞任何訊息給伺服器端，因此稱之為**『靜態網頁伺服器』**，如圖 6-18 所示。

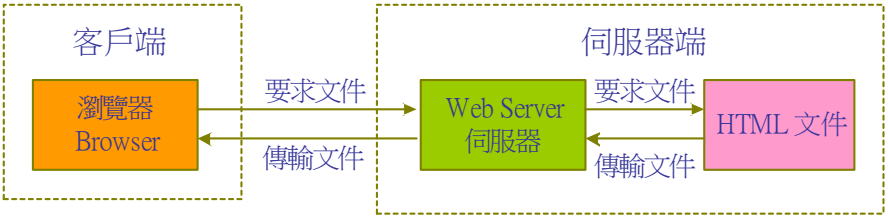

圖 6-18 靜態網頁伺服器架構

6-8-2 動態網頁伺服器

隨著網站伺服器的盛行，單方向的瀏覽網站上的網頁內容，並不能夠滿足環境的需求。我們需要類似資料庫伺服器一樣有效率的架構，讓客戶端和伺服器端能相互交換訊息，伺服器端能提供資料庫環境讓客戶端可以查詢使用。所以有所謂的**『三層發展架構』(Three-Tire Architecture)** 的產生，可分為客戶端（Client）、Web 伺服器（Web Server）、與資料庫伺服器（Database Server）三個層次。資料庫伺服器負責儲存應用程式所需的資料，Web 伺服器則負責伺服功能的運作與網頁的建立，而客戶端便利用瀏覽器和伺服器連接，透過網頁的各種呈現方式讓使用者和伺服器端、資料庫端做單/雙向的溝通，此架構稱之為**『動態網頁伺服器』(Dynamic Web Server)**。

程式設計師利用 HTML 語言在網站設計各種表格（Form），讓使用者在瀏覽器上填入。這些表格可以是：(1) 提供線上目錄及訂購產品；(2) 使用者登記基本資料；(3) 使用者填問答卷等等。在使用者輸入之後，表格內容會進行組合及轉換，轉換成資料庫可以辨識的格式或相容的查詢語言，資料庫才能執行查詢（query）的工作。

在許多的資料庫系統裡，我們必須定義一個標準介面來結合資料庫和 HTML 之間的連接，以期不同的資料庫系統都能提供標準介面的連接方式，使用 HTML 所製作的程式就很容易連結到各種資料庫系統，甚至程式設計師不需要去考慮到底使用何種資料庫統，皆可使用標準模式發展應用程式。這個標準介面稱之為**『共通閘門介面』(Common Gateway Interface, CGI)**，凡是具有 CGI 介面的程式也稱為 CGI 程式。CGI 程式可由 Perl、C 、或 Java 等程式語言發展。當瀏覽器執行到 CGI 程式時，它會透過 CGI 介面查詢資料庫系統，如圖 6-19 所示。

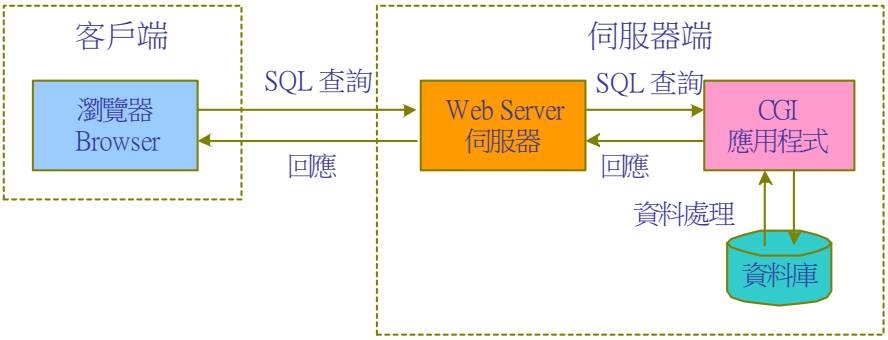

圖 6-19 動態網頁伺服器架構

6-8-3 Server/Daemon 動態網頁伺服器

為提高網站的執行效率，我們希望使用連結常駐記憶體中的『多重執行緒』(Multi-Thread) 來提高效率，而不再使用呼叫程序 (Procedure Call) 方式。兩者最大的不同點是，程序必須經過呼叫才會載入記憶體中執行 (原來不存在)，執行完畢便釋放記憶體 (也不再存在)，下一個程式有用到該程序時再載入記憶體；而執行緒是常駐在記憶體內，應用程式使用到某一個執行緒，便連結到該執行緒，執行完後，便切斷彼此之間連線，該執行緒還是停留在記憶體內。一般執行緒可由多個應用程式連結，稱之為『多重執行緒』。不像一般程序(procedure)，如被三個應用程式呼叫，就有三份程式被載入記憶體內。程式的設計以手搞 (Scripting) 方式為主，取代了傳統以 C、Perl 等所寫的 CGI 程式，使用簡單的 Script 語言做動態網頁設計，其工具如：IBM 公司的 Net.Data、Sun 公司的 JSP (Java Server Page) 和 Microsoft ASP (Active Server Page) 等等。其架構如圖 6-20 所示。

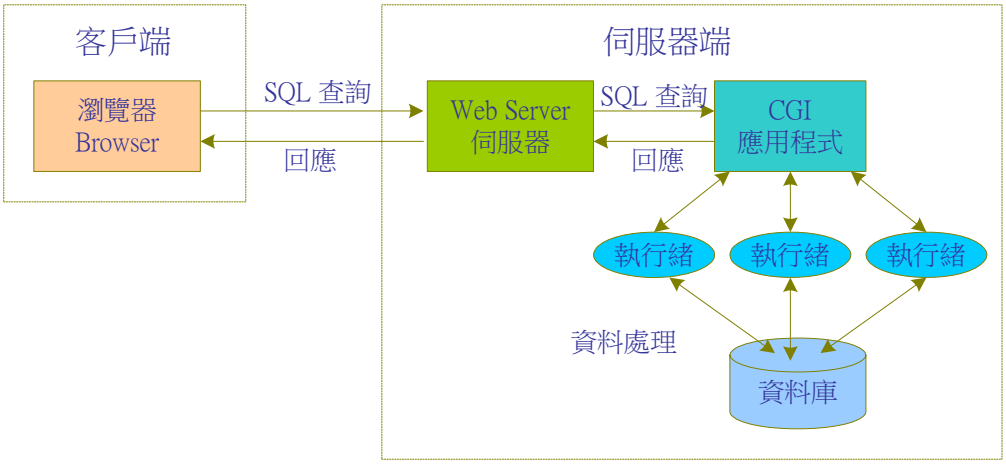

圖 6-20 Server/Daemon 動態網頁伺服器架構

習題

1. 請簡述交談層（Session Layer）的功能。
2. 交談層中對談有哪三種方式？請分別敘述其特性。
3. 交談層中交談模式有哪三種？請分別敘述其特性。
4. 請利用虛擬碼(Pseudo-code)寫出雙向交替對談模式(圖 6-4)的演算法。訊號方式如下：

   Allocate（要求連線，並要求 Token）、Receive_and_Wait（釋放 Token 等待接收）、Confirm（要求確認）、Confirmed（回應確認）、Send_Data（傳送資料）、Deallocate（要求結束）。
5. 請利用虛擬碼（Pseudo-code）寫出以滑動視窗法，作同步點的插入與回覆演算法。
6. 請簡述表現層（Presentation Layer）的功能。
7. 何謂獨立資料格式？請說明其特性。
8. 請簡述應用層（Application Layer）的功能。
9. 何謂網路作業系統（Network Operating System, NOS）？與一般作業系統有何差別？
10. 請至少列出五種網路作業系統的通訊協定的堆疊。
11. 請問您是否能尋找出，關於 NetBEUI 介面程式的相關資料，來製作一個簡單的檔案伺服器。
12. 同上題，請在 LLC 介面程式上製作一個簡單的檔案伺服器。
13. 請在您的電腦上執行下列程式，並寫出其結果，以及當時描述網路狀態。

    # ping www.nsysu.edu.tw
    # finger
    # route
    # traceroute www.nsysu.edu.tw
14. 何謂主從式架構？並舉一應用實例說明其特性。
15. 分散式處理的基本理念是什麼？應如何去實現它的理想。
16. 請至少舉出三個實用例子來說明表現分散處理的特性。
17. 何謂分散式資料庫系統？您是否可以找出一種該特性的銷售產品，並列舉實用的例子。
18. 使用於資料庫管理系統中，採用應用分散處理架構與資料分散架構之間有何差異？

19. 如果以檔案伺服器來作共享資料庫的記錄（Record）存取時，為何非常困難解決資料一致性的問題？您有解決辦法嗎？

20. 何謂磁碟伺服器？請說明其應用範圍。

21. 請簡述全球資訊網（WWW）的基本架構。

22. 請說明 HTML 語言的特性。

23. 請說明靜態網頁伺服器架構的特性。

24. 請說明動態網頁伺服器架構的特性。

25. 請說明 ASP（Active Server Page）的特性。