CHƯƠNG 6: Các dịch vụ mạng máy tính

6.0. Những quan niệm về dịch vụ mạng máy tính (computer network)

Ngày nay, có những máy tính cở lớn có thể phục vụ đến vài ngàn người và cũng có những máy tính chỉ phục vụ cho một vài người sử dụng. Nhưng, để có thể sử dụng dịch vụ hệ thống rộng rãi và để có thể dùng chung các nguồn tài nguyên do nhu cầu của các công ty, các trường học, các xí nghiệp và nói chung do những nhu cầu trao đổi thông tin rộng rãi của toàn xã hội, các máy tính nói trên được kết nối thành mạng.

Đến nay, kiểu phân bổ chức năng khách-chủ là tiện lợi nhất: Những máy tính chuyên dụng chứa đựng nguồn tài nguyên phong phí được gọi là các máy chủ (*file server)* và tạo nên các chức năng bổ sung cho các thành viên của nhóm công tác *(Working group)*. Với các mạng máy tính hiện hành, máy chủ đã tạo cho người sử dụng những chức năng dưới sau đây:

✧ *Chia sẽ tệp tin (file sharingo):* Mọi người sử dụng (chủ và khách) có thể cùng nhau tạo lập và cùng nhau sử dụng các tài liệu và các dữ liệu.

✧ *Thư điện từ (elctronic mail)*: các thông tin điện tử gọi là thư điện tử được dịch vụ như phương tiện thong tin, cụ thể đó là các phiếu cập nhập thanh toán hoặc ghi chép giữa các người dụng trên mạng vi tính

✧ *Chia sẽ máy in (printerscharing):* Cả nhóm công tác có thể dùng chung một máy in, do đó, việc in ấn các bản vẽ hay các tài liệu được thực hiện trên một loại máy in nào đó ở trong mạng, phương pháp này gọi là cách quảng (*remote priting),* đã tạo điều kiện giảm thiểu đáng kể phí tổn nếu phải trang bị nhiều máy in.

✧ *Điều hành công việc (job management)*: Qua việc phân bổ các nhiệm vụ riêng lẻ trên các máy tính, những thành viên khác nhau của nhóm có thể xử lý công việc nhanh hơn, rút ngắn thời gian tính toán và thời gian thực hiện chương trình.

Sự khác nhay giữa các máy tính riêng lẻ và mạng máy tính đối với người sử dụng thì khó nhìn thấy, khi chúng ta nói về hệ thống máy tính phân bổ. Các chức năng được kể ở trên đạt được nhờ một sự trình diễn có mục đích của nhiều thành phần hệ điều hành trên các máy tính khác nhau. Do đó, việc mở rộng mạng máy tính có thể coi như việc mở rộng hệ điều hành.

Những ưu điểm được mô tả ở trên là có thể thực hiện trên mạng máy tính; mặc dù vậy, chúng tồn tại nhiều khó khă trở ngại bởi nhiều kiể máy tính được lắp đặt vào mạng, nhiều loại hệ điều hành và nhiều loại ngôn ngữ lập trình: sự hợp tác của máy tính cho phép những tiêu chuẩn mạng khác nhau; những tiêu chuẩn này tồn tại trong cả phần cứng và trong cả phần mềm. Vì vậy, chúng ta mong muốn đón nhận trong chương này vai trò quan trọng của hệ điều hành; đồng thời, chúng ta tiếp tục nghiên cứu các nhiệm vụ , các kiểu chức năng cà các giải pháp một cách đầy đủ hơn, mà một sự kết nối mạng sẽ mang chúng lại cho hệ điều hành.

Để thống nhất hoá một phạm vi rộng lớn, liên hiệp các nhà sản xuất máy tính đã giới thiệu một thử nghiêmh quan trọng về môi trường máy tính phân bổ (*distributed compting environment: DCE;)* nó là cơ sở các phần mềm mở (*open software fundation:OSF)* chứa đựng các giải pháp khác nhau về quản lý công việc vủa hệ thống *client/server* (khách/ chủ) và về quản lý tệp tin cũng như các cơ chế bảo vệ.

6.1 Kết nối mạng máy tính

Với phương hướng thưa nhất để các máy tính làm việc độc lập với nhau trong mạng, có một bộ điều khiển được dẫn vào cho việc nối mạng, giống như một bộ móc thiết bị phải được lắp vào nhờ bộ kích tạo ở trong nhân hệ điều hành.

Đối vớikiểu kết nối logic của các thông tin, chúng ta có thể áp dụng trở lại các sơ đồ được nêu ra trong mục 2.4.1 ở đầu chương 2: Đầu tiên, chúng ta tạo ra một sự kết nối; hoặc là, chúng ta sử dụng địa chỉ người nhận để gởi thông tin, và do đó, chúng ta đạt được một sự trao đổi thông tin không kết nối. Đối với việc thực hiện trao đổi hướng kết nối logic, người ta có thể áp dụng hai ý kiến: Một cách vật lý, chúng ta có thể tạo ra một surk liên kết cố đinh; sau đó, chúng ta có thể gởi thông tin qua đường dây điện thoại cố định;hay chúng ta có thể bắt đầu cách nối tiếp vật lý nhờ các thông tin đặc biệt qua mạng máy tính, tiếp đến, gởi các thông tin trên con đường vừa chuẩn bị. Việc kết nốicác máy tính tới vavs máy tính riêng lẻ thì không chỉ tồn tại một dạng vật lý; thực ra, nhiều kết nối như thế có thể sử dụng đường dây dẫn, mà không hề có trở ngại gì. Đối với việc thực hiện hai ý kiến trên ở mạng máy tính thì có thể đạt được do việc phân xẻ thông tin thành các gói tin (*daten package)* và do việc chuyển liên tục các gói tin tới các địa chỉ người nhận ở trong mạng

Một cách bình thường, ở kết nối mạng máy tính, thì không tồn tại kiểu kết nối vật lý điểm tới điểm, mà nhiều máy tính được kết nối với nhau bằng cáp đồng trục. Để thích hợp cho một máy xác định, trên cáp này, không phải chỉ có các dữ liệu, mà cả những thông tin địa chỉ cũng truyền tải. Những thông tin quản lý này (chẳng hạn chiều dài thông tin, tổng ngang để kiểm tra lỗi truyền đat...) được liên kết với các dữ liệu trong một gói tin. Nhiệm vụ của bộ điều khiển điện tử là thực hiện việc chuyển đổi giữa các tín hiệu điện tử trong cá và khuôn khổ logic của các gói tin đối với việc đọc khi nhận, và đối với việc viết khi gởi. Với kiểu dịch vụ này, những chức năng tiếp theo như việc điều khiển gơie thông tin, tạo lập môtn kết nối thông tin logic với các máy tính khác phải được tạo lập trong sự trợ giúp của một dãy tuần tự các gói tin. Kết quả các bước trao đổi thông tin để đạt được một mục đích định trước gọi là một giao thức (*protocol),* nó cũng chính là sơ lược các quy tắc trao đổi thông tin *(communicatin).*

6.1.1. Các lớp công việc

Việc thiết lập các dịch vụ cao hơn phù hợp với cấu trúc hiện hữu nhờ các cơ cấu ảo nối ở chương 1 và nó được tiêu chuẩn hoá kiểu các lớp OSI, gọi là *hệ thống mở kết nối với nhau (Open Sytem Interconnect)* của *tổ chức tiêu chuẩn hoá quốc tế* ISO *(Internetional Standards Oganization)*, xem hình 6.1 ở dưới đây.

HÇNH 6.1 TRANG 215

Các lớp khác nhau được đánh số từ 7 tới 1, có các nhiệm vụ sau đây:

7. Tại lớp áp dụng, các dịch vụ do người sử dụng được chỉ dẫn như các ứng dụng đồ hoạ, kiểm tra an toàn, trao đổi thông tin (như E-mail)...
6. Tại lớp trình báo, các dữ liệu được tạo dạng và khẳng định việc mã hóa hay khuôn dạng nén cũng như việc phân nhóm các dữ liệu, thí dụ ở kiểu tạo dạng bản ghi thù phụ thuộc người sử dụng
5. trên lớp toạ đàm, nó được khẳng định: ai là người gởi và ai là người nhận; các lỗi được xem xét như thế nào khi trao đổi thông tin... Nói chung, lớp này thường được bỏ qua
4. Lớp chuyển vận làm thay đổi các dong dữ liệu ở các gói tin, nó lưu ý khi đánh số và khi nhận thì nhận đúng gói tin. Ở đây, lần đầu tiên, một sự lưu ý đã được chấp nhận trên phần cứng thuộc cái đó. Các yêu cầu khác nhau của sai số lỗi được lựa chọn ngay ở đó. Một đại diện kiểu giao thức này là giao thức TCP.
3. Trên lớp mạng, tất cả các câu hỏi và các vấn đề còn tồn tại được trao đổi có lưu ý tới các quan hệ về địa hình và khoảng cách của mạng máy tính. Sự chuyển hướng của thông tin ở trạng thái chuyển vận trên đoạn đường xác định, cấu hình thiết bị giới hạn bởi bề rộng băng từ và chiều dài cáp đồng trục, tóm lại, tất cả các điều kiện ký luật của mạng máy tính được xem xét kỹ càng ở đây. đối với việc trao đổi thông tin không có kết nối, hầu như giao thức IP (*internet protocol)* hay một ấn bản khác của loại đó UDP *(uniform datagram protocol)* được sử dụng; tuy nhiên, với cái đó, dãy tuần tự các gói tin trên đường đến người nhận có thể tự mình tu chỉnh. Ngược lại với cái đó, giao thức nổi tiếng X.25 đã đảm bảo dãy tuần tự, vì giao thức định hướng kết nối này sử dụng một kiểu kết nối được tạo lập trước đó cho tất cả các gói tin một cách chặt chẽ.
2. Lớp liên kết dữ liệu đã làm phân đoạn các tập dữ liệu thông tin (có đọ lớn không đều đặn) thành các gói tin nhỏ riêng lẻ có độ lớn xác định, gởi chúng đi và lặp lại việc gởi, nếu không có thông báo trở lại hay nếu việc tổng kiểm tra lỗi ở bên nhận không có vấn đề gì. Bởi lẻ, nó được làm đầy chức năng bảo vệ dữ liệu.
1. Các gói dữ liệu sẽ được biến đổi khi dẫn tới việc gởi trên lớp kết nối vật lý thành các Bit nhị phân, mà những Bit nhị phân này được mang theo như các xung điện hay các xung quang trên môi trường chuyển vận.

Việc thực hiện và áp dụng kiểu phân lớp này vẫn còn có sự khác nhau giữa các nhad chuyên môn, và do đó, vấn đề này vẫn còn tiếp tục trãi qua thực nghiệm...

Trong trường hợp đơn giản, người ta có thể tạo lập một sự kết nối kiểu *điểm tới điểm* giữa hai máy tính; trong đó, người ta kết nối chúng bằng dây cáp đồng trục; sự kết nối này thích hợp với thiết bị tốc độ chậm như máy in. Khi đó, các chíp vi mạch với các giao diện nối tiếp sẽ thực hiện các chức năng của lớp kết nối dữ liệu; bộ kích tạo lối ra thì đang ở bước thứ nhất. Đối với một sự trao đổi thông tin, người ta có thể áp dụng một chương trình KERMIT thông thường. Chương trình này do trường Đại học Columbia đề xướng, nó chứa đựng nội dung các lệnh để gởi dữ liệu (lớp 7) như là cơ chế mã hoá thành các gói tin thông tin (lớp 6). Ở đây, các lớp 5,4 và 3 được bỏ qua một bên, vì nó là kiểu kết nối tận hiến *điểm tới điểm*.

Ở các mạng máy tính với các yêu cầu cao hơn như E-mail và cùng nhau sử dụng các tệp tin, các lớp còn phức tạp hơn nhiều. Do vậy, điều đó đã đảm bảo rằng, bộ điều khiển không chỉ chứa các chíp, mà chúng thiên luận các lớp 1 và lớp 2; đặc biệt, một hệ thống vi xử lý riêng lẻ với bộ nhớ và một chương trình cố định trong ROM sẽ làm giảm tải với một môi trường truyền tải đặc biệt (thí dụ Ethenet) hay với một giao thức đặc biệt (thí dụ một sự kết hợp đặc biệt các giao thức TCP/IP). Những bộ điều khiển *thông minh* này có thể cùng làm việc với các bộ kích tạo của hệ điều hành trên mức đọ cao hơn.

Đối với hệ điều hành, điều quan trọng cần phải hiểu là, những lớp nào và những dịch vụ nào sẽ tồn tại trong bộ điều khiển và cần được xem xét tốt hơn.

Các lớp trao đổi thông tin ở trong Unix:

Ở trong hệ điều hành Unix, các lớp riêng lẻ thay thế nhau thì rất khác nhau. Thực ra, lúc đầu Unix không phải thiết kế cho mạng. Vì ngày từ đầu, mã nguồn của hệ điều hành này được dụng không mất tiền ở các trường đại học, cho nên, nó đã nhận được sự tham gia rộng rãi và đem lại các ấn bản phát triển cũng như các ấn bản thiết kế quan trọng đối với mạng máy tính và đã được thực nghiêmh thành công mỹ mãn trên hệ điều hành Uinx.

Một bản thiết kế mong mỏi với Unix System V đã được dẫn tới việc sắp xếp các bộ kích tạo thiết bị các lớp. Ý tưởng cơ bản của bản thiết kế này là tạo ra dòng lư thông (*stream)* các dữ liệu xuất-nhập của một kết nối nhờ các bậc xủ lý khác nhau của bộ kích tạo, xem hình 5.2 ở trên. Hệ thông sdòng lưu thông này tạo ra trên một giao diện đã được chuẩn hoá, để di dịch các bậc xử lý bất kỳ hay để đón nhận dòng lưu thông. Do đó, nó có thể giảm nhẹ việc trao đổi các lớp của giao thức và việc lựa chọn liên hiệp các giao thức TCP/IP. Hình 6.2 chỉ ra một lớp chứa đựng các dịch vụ được nhắc đến ở trong mục 6.2.2 ở dưới đây.

7 Ap dụng	Named pipes, rlogin,...
6. Trình báo	XDS
	Giao diện HĐH: socket
5. Toạ đàm	Ports, IP address
4. Chuyển vận	TCP/IP

3. Mạng 2. Liên kết dữ liệu	
1. Kết nối vật lý	Ethnet

Hình 6.2. Các lớp của các giao thức thường hay sử dụng ở trong Unix

Gọi hệ thống các dịch vụ vận chuyển ở trong nhân hệ điều hành xảy ra nhờ các gọi hệ thống đặc biệt. Một áp dụng có thể hoặc là sử dụng trên lớp cao hơn như named pipes; hoặc là nó có thể thiết đặt trên các dịch vụ hệ điều hành socket() một cách sâu sắc, do đó đạt được một sự trao đổi thông tin trên mạng.

Các lớp trao đổi thông tin ở trong Windows NT:

Ở hệ điều hành Windows NT, nhiều dịch vụ mạng khác nhau được đem lại. Những dịch vụ này cho phép tương thích với nhiều giao thức mạng riêng biệt thuộc hệ điều hành MS-DOS; thí dụ giao thức *khối thông điệp máy chủ SMB (server message block),* giao thức *hệ thống mạng xuất-nhập cơ bản NetBIOS (Network Báic Inpit-Output System)* đều tương thích với các giao thức dịch vụ của các nhà sản xuất hệ thống khác. Cấu trúc này được sơ đồ hoá trong hình 6.3 chỉ cho chúng ta thấy rằng, việc phân lớp theo kiểu IOS-OSI được dẫn ra tương đối đầy đủ.

<table>
<tr><td>7</td><td>Ap dụng</td><td colspan="4">File, Named pipes, mail slots</td></tr>
<tr><td>6.</td><td>Đề xuất</td><td colspan="4">Hệ thống con</td></tr>
<tr><td></td><td></td><td colspan="4">Dịch vụ lặp</td></tr>
<tr><td>5.</td><td>Toạ đàm</td><td colspan="2">NetBIOS</td><td>NBT</td><td>Windows-socket</td></tr>
<tr><td>4.</td><td>Chuyển vận</td><td rowspan="3">Net BEUI</td><td rowspan="3">IPX/ SPX</td><td rowspan="3" colspan="2">TCP/IP</td></tr>
<tr><td>3.</td><td>Mạng</td></tr>
<tr><td>2.</td><td>Liên kết dữ liệu
Giao thức NDIS</td></tr>
<tr><td colspan="2"></td><td colspan="4">Bộ kích tạo NDIS</td></tr>
<tr><td colspan="2">1. Kết nối vật lý</td><td colspan="4"></td></tr>
</table>

Hình 6.3. Mô hình OSI và các thành phần mạng của Windows NT

Lớp chuyển vận được che phủ bởi gia thức đầy quyết định NetBEUI, gọi là giao thức liên hiệp của *giao diện người sử dụng nânng cao kiểu hệ thống xuất-nhập trên mạng (NetBIOS extended user interface)* hau liên hiệp các giao thức IPX/SPX của công ty Novell hay bởi giao thức thông dụng TCP/IP. Ngoài ra, các lớp này còn mang trên mình giao diện chuẩn cho bộ điều khiển mạng NDIS *(Network driver interface specification)* của hãng Microsoft

Các lớp cao hơn 5,6,7 cũng đã được che phủ bởi lớp SMB, mà lớp này tạo ra một cách trực tiếp vào các trạng thái sockets theo giao thức NBT và NetBIOS, do đó đạt được sự trao đổi thông tin.

6.1.2 Các hệ điều hành phân bổ

Một hệ điều hành được kết nối mạng và được tồn tại một cách toàn vẹn thì được biểu thị là hệ điều hành mạng máy tính. Bây giờ, người ta có thể phân bổ nhiêmh vụ, chẳng hạn dẫn các tệp tin tới các máy tính ở trong mạng. Liên quan tới nhiệm vụ này, người ta nói về một hệ phân bổ, cụ thể đó là hệ thống tệp tin phân bổ. Các chức năng của hệ thống phân bổ bị giới hạn trong trường hợp của một hệ điều hành mạng máy tính trên các dịch vụ cao và trên các hệ thống chương trình được chuyên môn hoá cao của người sử dụng.

HÇNH 6.4 TRANG 218

Ngược lại, mỗi thành phần của một hệ điều hành phân bổ tồn tại chỉ một lần riêng lẻ trên một máy tính. Do đó, tất cả các thành phssn cũng như máy tính phải làm việc cùng nhau; tất nhiên, chỉ có hệ điều hành là chung cho toàn mạng.

Một hệ điều hành như vậy chỉ sử dụng những lớp thấp nhất của giao thức chuyển vận, do đó, nó có thể thích ứng với các máy tính khác nhau với dịch vụ phân bổ nhanh nhạy. Hình 6.4 chỉ ra việc phân lớp đối với nhân của một hệ thống.

Phần nhân của hệ điều hành tồn tại trên mỗi máy là phần nhân tối thiểu (*microkernel).* Phần nhân tối thiểu này chỉ chứa đựng những dịch vụ cần thiết để thực hiện việc trao đối thông tin và các dịch vụ cơ bản cho việc quản lý bộ nhớ và chuyển đổi tiến trình. Tất cả các dịch vụ khác như quản lý hệ thống tệp tin *(file server)*, quản lý in ấn *(printer server),* xây dựng cây thư mục *(director server),* điều hành công việc (*process server)...*được định vị trên các máy tính chuyên môn hoá. Kiểu hệ điều hành này có những ưu điểm sau đây:

Tính linh hoạt:

Các dịch vụ tiếp theo (như dịch vụ tính toán...) có thể được lấy đi hay được đưa ra thêm vào yêu cầu khi máy tính hoạt động; hệ thông smáy tính có thể được mở rộng tăng dần.

Tính biểu trưng:

Với tính biểu trưng của hệ điều hành, các dịch vụ ở trong mạng được mang lại mà không cần người sử dụng biết điều đó xảy ra ở đâu.

Độ sai số lỗi:

Một cách nguyên tắc, sai số lỗi là điều có thể; do đó, hệ thống máy tính có thể tái tạo dạng ở bên trong mà không cần người sử dụng biết điều đó.

Việc gia tăng hiệu suất:

Vì tất cả các dịch vụ có thể được dẫn tới sang hành; do đó, nhờ các máy tính bổ sung, người ta có thể đạt được việc thực thi cao hơn.

Tuy nhiên, nói đúng ra, người ta cũng nhận thấy rằng, các hệ điều hành phân bổ cũng có nhược điểm vì chi phí phần cứng lớn, thí dụ, nếu một dịch vụ chỉ tồn tại có một lần và ngay sau đó, loại khỏi máy tính nay. Tóm lại có thể nói rằng, hoạt động của toàn mạng thì quan hệ với hiệu suất của dịch vụ.

Vẫn đề cơ bản của nhân hệ điều hành phân bổ là sự tổn hao thời lượng đối với một số dịch vụ qua trao đổi thông tin. Đối với hệ thống hiệu suất, nó thì có lợi để xử lý những gói tin lớn ngoài khả năng; đa số các hoạt động nhỏ tồn tại trong nhân hệ điều hành sẽ được thực hiện nhanh hơn trên hệ thống vi xử lý tương ứng.

Đối với các dịch vụ mạng, người ta còn lưu ý một vấn đề, khi các nhân hệ điều hành được sắp đoạn hay được tái tạo, mà chúng có thể nhận được các chức năng hệ điều hành một cách tự chủ, khi đó sẽ đem lại những dịch vụ chuyên dụng cao hơn (thí dụ việc điều khiển truy cập mạng), mà chức năng của chúng được phân bổ. Vì các dịch vụ cơ bản này được tính toán một phần thuộc hệ điều hành và nó được chứa đựng trong phạm vi cung cấp của hệ điều hành; do đó, nó được nói về các hệ điều hành phân bổ nhiều hơn hay ít hơn mà không cần phải làm chủ một nhân tối thiểu.

Trên cơ sở này, điều cần thiết là phải xem xét sự hoài nghi, liệu, một hệ điều hành tập trung *(main frame)* hay một hệ điều hành phân bổ *(client server)* thì tốt hơn: Hầu hết các hệ điều hành là một sự pha trộn từ một hệ điều hành mạng thuần khiết, mà hệ điều hành này tự làm tất cả hay được kết nối một cách rời rạc với một hệ thống các máy tính khác nhau, chẳng hạn với một hệ điều hành phân bổ thuần khiết, nó dẫn tới tất cả các dịch vụ trên máy tính được chuyên môn hoá. Phạm vi của nhân chỉ ra trạng thái quá độ giữa hai điểm cực trị.

Ở mục 6.4.2, chủ đề này được nhắc tới một lần nữa trong thì dụ về máy trên mạng.

6.2. Trao đổi thông tin trên mạng

Nếu chúng ta mở một kết nối trao đổi thống tin *điểm tới điểm* và nếu mỗi máy tính quan hệ với một máy tính khác; do đó, đối với một kết nối có mục đích tới một máy riêng lẻ, người ta phải làm thích hợp máy tính này với một cái tên hay một địa chỉ.

6.2.1. Tên mở rộng trọng mạng

Có những quy ước tên mở rộng khác nhau ở trong mạng; các quy ước này được thay đổi thêm mỗi khi sử dụng. Do đó, chúng ta có thể phân biệt một cách thô thiển giữa tên được sử dụng rộng rãi ở trong mạng lớn (đặc biệt trọng mạng Internet) và các tên trong các mạng cục bộ diện hẹp.

Các tên trong mạng diện rộng:

Trong sự lưu thông giữa máy tính diện rộng của mạng Internet, một quy ước tên được công nhận; nó được phát triển theo lịch sử. Do đó, tất cả các cách tổ chức ở Hoa Kỳ được phân chia thành các nhóm khác nhau:

Com	cho các công ty
Edu	cho các trường đại học và trường phổ thông;
Gov và mit	cho chính phủ và quân đội;
Net	cho người công tác trên mạng;
Ong	cho tất cả các tổ chức cùng sử dụng

Trong một nhóm (*top level domain)*, mỗi thành viên của nhóm *(domain)* được phân bổ một cái tên; thí dụ đối với trường đại học Berkeley với nhóm top lever domain thì tên edu được bổ túc thêm berkeley. Tên đầy đủ cho một mảy tính của trường đại học được cấu thành tên vùng *(domain name)* và tên máy tính *(computername),* các tên được tách biệt nhau bằng dấu chấm.

Thí dụ về tên máy tính:

Tên đầy đủ của một máy tính Okeeffe của trường đại học Bekeley gọi là: okeeffe.berkeley.de

Ở đây, việc viết chữ lớn hay chữ nhỏ không cần lưu ý. Đối với phần mở rộng, có một sự thoả thuận khác được đưa ra. Thay vì viết tên nhóm, có thể dẫn ra tên viết tắt của quốc gia. Thí dụ một máy tính của bộ môn có tên là diokles, nó được xác định bởi địa chỉ: diokles.informatic.uni-danang.vn

Một địa chỉ tượng trưng như vậy thì quá dài. Do đó, nó tồn tại đối với mỗi máy tính một số logic; con số này bao gồm 4 con số đứng liền kề nhau (địa chỉ Internet là 32 Bit).

Thí dụ về địa chỉ trên mạng Internet:

Một máy tính như nói ở trên có địa chỉ số: 141.2.1.2.

Hai số đầu (141.2) phù hợp với nhóm vùng (*domain)*, đó là uni-danang.vn; và nó được một cơ quan (ở Mỹ do trung tâm thông tin mạng NIC) trao cho chính xác như tên vùng *(domain name)*; ở đây, các con số khác cho thấy: con số thứ 3 (.1.) chỉ mạng con *(subnet)* và con số thứ 4 (.2.) chỉ máy tính riêng lẻ của một tổ chức vùng nào đó kết nối vào mạng Internet.

Từ các số logic, người ta chọn ở trong mỗi bộ điều khiển của giao thức TCP/IP có một con số cố định tương ứng, đó là con số duy nhất đối vơi sbộ điều khiển này, nó được tạo lập một cách cố định và không thể được thay đổi (khác với các số logic của địa chỉ IP). Khi tạo lập một sự kết nối; đầu tiên, con số vật lú này được thạm chiếu, tiếp đến tham chiếu các đặcđiểm logic.

Sự sắp xếp địa chỉ logic tới địc chỉ ảo bằng các chữ cái phải được giữ lại với sự trợ giúp của một bảng. Trong các hệ thống Unix, chúng được tìm thấy dưới đường dẫn /ect/hosts; ở các hệ thống khác nó là một ngân hàng dữ liệu thực thụ.

Một cái tên trong mạg Internet như vậy có thể được sử dụng để yêu cầu một dịch vụ trên một máy tính

Thí dụ về các dịch vụ trên mạng Internet:

Trên trang WEB (worrd wide web:WWW), để có một dịch vụ trình diễn text (*hypertext presentserive),* một giao thức URL (*uniform resource locator)* được cấu thành như là một địa chỉ bởi ba thành phần :tên của giao thức dịch vụ (thí dụ:*http://*), tên nối mạng của máy tính và tên tệp tin cục bộ. Đối với dịch vụ sao chép tệp tin, giao thức URL dẫn tới cho máy tính vùng với uni-danang.vn và cho tệp tin với /public/Text.dat. Như vậy, ta có đường dẫn đầy đủ:

ftp://ftp.infomatic.uni-danang.vn/public/Text.dat

Người ta lưu ý rằng, việc mô tả tên bằng chữ cái boá hay chữ cái thường ở trong mạng Internet thì không quan trọn; nhưng ở tên tệp tin hay đường dẫn trong mạng cục bộ thì điều đò không thể được. Máy tính vùng uni-danang.vn cũng có thể được tìm thấy theo cách viết Uni-Danang.VN , nhưng tệp tin public/Text.dat thì không thể viết Public/ext.dat.

Việc sắp xếp tên theo địa chỉ IP logic thì không thực hiện tự động, mà nó phải được cho phép theo một danh sách tên. Ở mạng cục bộ, điều này được hoàn thiện bởi máy tính chủ quản lý cây thư mục; máy tính này dừng lại ở địa chỉ mong muốn trực tiếp trong một danh sách tất cả các máy tính nó quản lý; hoặc nó phải hỏi lại một máy tính khác. Tên máy tính và các dịch vụ toàn cùng của chúng (*all domain)* được dẫn ra trong một máy chủ (*domain name server)* quản lý hệ thống tệp tin toàn vùng; khiđó, máy tính quản lý vùng này sẽ nhận biết các máy tính kết nối trực tiếp cũng như các máy tính kết nối trực tiếp cũng như các máy tính quản lý vùng tiếp theo; tại đây, nó dẫn tiếp các thăm dò tới các máy tình còn chưa biết rõ.

Cây thư mục trong mạng cục bộ:

Ơ trong mạng Internet, chúng ta đã biết cách tìm tên một máy tính. Ở mạng cục bộ Lan *(local area network)*, thí dụ với rất nhiều máy tính tồn tai trong một phòng lớn hay được phân bổ thành nhiều phòng theo từng nhóm công tác; việc quy ước tên rõ ràng thì thật khó khăn. Để chuyển vận một tệp tin từ một máy tính tới một máy tính khác, người ta phải chỉ ra hai tên máy tính cũng như đường dẫn của hệ thống tệp tin cục bộ. Tính mềm dẻo này thực ra không cần người sử dụng quan tâm; vì nhóm công tác của anh ta luôn luôn tồn tại như nhau.

Do đó, mục đích là, để thiết đặt các cơ chế cho hệ điều hành. Mà với cơ chế này, có thể cho phép truy cập lên một hệ thống tệp tin của một máy tính khác một cách thông suốt. Nếu có nhiều máy tính cùng làm việc đồng thời trong mạng cục

bộ; do đó, người ta có thể liên kết với nhau các hệ thống tệp tin của các máy tính (thí dụ trong hình 6.5, đó là các máy tính vùng Hera và Cronos) thành một cây thư mục. Người ta ký hiệu dấu hhai sọc chéo "//" cho thư mục gốc nối ảo của hệ thống tệp tin; nó phục vụ như là một khoá cho việc dò hỏi tệp tin.

HÇNH 6.5 TRANG 223

Một phương pháp như vậy đã dẫn tới một hệ thống tệp tin trên hai máy tính tương tự nhau; tất cả các người sử dụng của các máy tính có thể được tham chiếu cây thư mục như nhau.

Tuy nhiên, còn có những phương pháp khác nữa; thật vậy, chúng ta nhận thấy rằng, việc tạo ra chỉ kết nối một phía; do đó, ở ví dụ của chúng ta, máy tình Cronos đã được kết nối với thư mục NewDepartment. Tất cả việc dò hỏi trên máy tính Hera được dẫn tới máy tính Cronos, được xử lý ở đó hay được đưa trở lại máy tính Hera. Đối với người sử dụng của máy tính chủ Hera thì được dâng hiến toàn cảnh hệ thống tệp tin, như đã chỉ trong hình 6.6 dưới đây.

HÇNH 6.6 TRANG 223

Đối với người sử dụng của máy chủ Cronos thì cũng không có gì thay đổi; anh ta không cần ghi nhớ dịch vụ bổ sung hoặc việc máy tính Cronos thình cầu đối với máy chủ Hera; anh ta chỉ cần lưu ý tới hệ thống tệp tin được giới hạn; do kiểu kết nối không đồng đều các hệ thống tệp tin trong mạng cục bộ.

Thí dụ về hệ thống tệp tin phân bổ trong Unix:

Hệ thống tệp tin kiểu AFS (*anderew file system)*, do trường đại học Acrnegie-Mellon đề xướng, đã trợ giúp một hình dạng tệp tin đồng đều toàn cục và được bổ trợ bởi hệ thống tệp tin phan bổ DFFS (*distribute gile system)* cho ccs gói tin DCE. Đối với mỗi tệp tin, điều được quyết định là; liệu nó tồn tại một cách cục bộ hay khoảng cách (?) và sau đó, nó được tổ chức truy cập như thế nào (?). Do vậy, các cờ hiệu thay thế được sử dụng để trao đổi dữ liệu và trao đổi giao thức giữa các máy tính (xem hình 6.17).

Thí dụ về hệ thống tệp tin mạng ở Unix:

Hệ thống tệp tin mạng NFT (*network file system)* được công ty máy tính SUN phát triển cho hệ điều hành Unix và được phổ biến rộng rãi ở các ấn bản khác nhay của Unix, phần mở rộng của nó được khẳng định hướng theo kiểu không đồng nhất. Xuất phát điểm là, một hay nhiều máy tính chủ dịch vụ tệp tin tồn tại trong mạng, chúng sẽ dùng chung một hệ thống tệp tin. Nếu một máy tính muốn sử dụng một lệnh đặc biệt muont() để phân tán cây thư mục của máy chủ cới một thư mục xác định.

Vì điều đó có thể xẩy ra tại mỗi máy tính,do đó, cây thư mục thì có thể khác nhau trên mỗi máy. Cách phô diễn này của hệ thống NFS thì rất cần thiết cho việc bố trí cây thư mục.

Thí dụ về cây thư mục trong mạng ở Windows NT:

Ở hệ điều hành Windows NT,có hai cơ chế khác nhau để truy cập các tệp tin của các máy tính khác nhau. Cơ chế thứ nhất dùng dịch vụ kết nối ảo (*symbilic links),* mà điều này đã được trình bày trong mục 4.2.2 ở trên. Trong cây thư mục có các thiết bị có đường dẫn \Device, một bộ kích tạo đặc biệt được treo vào cho mỗi loại tệp tin mạng (thí dụ: hệ thống tệp tin kiểu giao thức lặp tại MS, hệ thống tệp tin mạng Novell...); mà phương pháp phân tích của chúng (*parse methode)* dẫn tới việc tạo lập một kết nối mạng hay dẫn tới việc đọc chọn hệ thống tệp tin mạng. Nếu bây giờ, người ta tạo lập một tên gọi cho thiết bị (chảng hạn V:) ở trên bộ kích tạo mạng này; do đó, nó sẽ được làm thích ứng một cách tự động khi mở một tệp tin. Ơ hình 6.7, người ta nhìn thấy kết quả của tên tệp tin được chuyển hướng kiểu như thế.

HÇNH 6.7 TRANG 224

Cơ chế thứ hau sử dụng khuôn khổ cây thư mục của Microsoft UNC (*Universal naming convention)*; trong đó, tất cả các tên thư mục của mạng được bắt đầu bằng ký tự hai dấu gạch xiên phải "\\". Nếu hệ thống con Windows32 nhận biết rằng, một cái tên như vậy được sử dụng; do đó, nhờ thiết bị DOS, thư mục "\\" đã được thay thế bởi thư mục "UNC:" một cách tự động. Do đó, một kết nối ảo /9*symbolic link)* được dẫn tới cho bộ kích tạo tên MUP, gọi là bộ đa lo liệu *(multi provide).* Bây giờ bộ này dò hỏi với một giao thức IRP tại bộ quản lý đăng ký tệp tin mạng, tiệu tên đường dẫn phù hợp đã nhận biết chưa(?). Nếu đó là trường hợp được nhận thấy ở khắp nơi, thì do đó, dãy tuần tự quyết định cho việc đăng ký của bộ quản lý tệp tin ở trong ngân hàng dữ liệu được đăng ký.

Giải pháp tên tệp tin với thư mục UNC:

Đầu tiên, chúng ta muốn mở tên tệp tin \textserver\public\text.doc. Sau đó, việc dò hỏi này được chuyển tới UNC:\textserver\public\text.doc. Bây giờ, từ bộ kích tạo thiết bị MUP, chuỗi ký tự tin textserver\public\text.doc sẽ được chuyển tiếp tục tới bộ kích tạo về *gởi lặp (redirector) hệ thống tệp tin mạng...* và cũng như phương pháp phân tích giao dịch thông điệp được gọi tới. Mỗi bộ kích tạo sẽ dò hỏi về việc kết nối mạng của nó tại máy chủ quản lý tệp tin, liệu tệp tin này có còn tồn tại không? Nếu có nó xẽ mở cho cái đó một sự kết nối.

Ở cây thư mục của Windows NT cũng như ở tại hệ thống tệp tin NFS, không có hình dạng thống nhất được đảm bảo ở các hệ thống tệp tin. Đầu tiên, các việc

kết nối mạng được tạo lập giống như ở hệ thống NFS và chúng mới bảo đảm mối quan hệ giữa máy chủ và máy khách.

6.2.2 Kết nối trao đổi thông tin

Để thực thi việc trao đổi thông tin trong mạng, các kiểu quy ước tên trong mạng phải sử dụng tới nhiều cơ chế khác nhau. Sau đây chúng ta sẽ lần lượt khảo sát các cơ chế đó.

Trao đổi thông tin kiểu nối cổng (ports)

Ở đây ý tưởng về các điểm trao đổi thông tin đã chấp nhận: mỗi máy tính chiếm một điểm cuối của trao đổi thông tin mạng tại hệ thống có nhiều kết nối trao đổi thông tin. Ở giao diện tới lớp TCP/IP, đó là những cổng *(ports)* kết nối; những cổng này đã được đáng số, gọi là địa chỉ số. Vài trong các số cổng 16 Bit được sắp xếp cho một kết nối một dịch vụ đặc biệt, gọi là Well known oprt number, mà người ta có thể chuyển thông tin tới đó. Hình 6.8 chỉ ra danh sách số cổng của các dịch vụ, với sự sắp xếp này, người ta tìm thấy ở trong Unix các dịch vụ tệp tin /ect/services.

Người ta có thể so sánh một điểm trao đổi thông tin như vậy với một hộp thư *(mailbox),* nó cũng giống như việc trao đổi thông tin được dẫn ra ở trong mục 2.4.1 ở trên. Đối với dịch vụ hộp thư, hệ điều hành phải tạo nên hai hàng đợi: một cho các thông tin kết nối và một cho các thông tin gởi tới. Ngoài ra, mỗi hàng đợi phải được đảm bảo với một cờ hiệu.

Sự tổng hợp các thư mục này và các cấu trúc dữ liệu tạo thành một lớp, mà lớp này được ghi lại trên các cổng và nó có thể xem xét như là điểm cuối của việc trao đổi thông tin giữa các tiến trình. Một sự kết nối trao đổi thông tin giữa tiến trình A và tiến trình B được mô tả như sau:

hình 6.8 trang 226

Thí dụ về trao đổi thông tin kiểu kết nối TLI:

Với hệ điều hành Unix System V, ấn bản số 3 dẫn ra một giao diện TLI (*transport layer interface)*. Giao diện này bao gồm một tệp tin thư viện (libnsl.a), gọi là thư viện dịch vụ mạng *(network service library:libnsl);* đồng thời, nó cũng phủ lên cả lớp vận chuyển. Một kiểu dễ dàng và tốt hơn được một uỷe ban tiêu chuẩn hoá X/Open chấp nhận như là một giao diện chuyển vận nâng cao ETI *(extended transport interface).* Hình 6.9 mô tả một sự trao đổi thông tin kiểu này. Trong giao diện TLI, một tiến trình có thể trao đổi thông tin đồng bộ hay không đồng bộ tuỳ theo cách chọn; ở giao diện này, việc trao đổi thông tin chỉ có thể là đồng bộ. Vì lớp chuyển vận xảy ra thật rõ ràng, cho nên, người ta có thể không cần

gây ảnh hưởng lên các tham số khác nhau, thí dụ lên giao thức TCP/IP. Điều này có thể gây nên những bất lợi mà sau này người ta cần phải xem xét tới.

hình 6.9 trang 226

Trao đổi thông tin kiểu hốc (socket's communication):

Một kiểu trao đổi thông tin dạng logic tiếp theo được gọi trao đổi thông tin kiểu hốc *(socket model)*. Kiểu này được tổ chức như một kiểu trao đổi thông tin định hướng điểm theo hàng đợi. Nó có sự khác nhau giữa máy chủ *(server)* và máy khác (*client)*: máy chủ thính cầu sự trao đổi thông tin với khả năng của mình và chờ đợi khách hàng; còn khách hàng muốn tạo một sự trao đổi thông tin vơi máy chủ. Đầu tiên, cả hai muốn tạo ra sự trao đổi socket với gọi hệ thống socket(). Các thủ tục tiếp theo cho phép kết nối trao đổi thông tin tới một cổng. Tên của tiến trình gọi được ghi với một gọi hệ thống bind() vào trong hệ thống. Khi đó, tiến trình server chờ đợi với gọi hệ thống listen(), cho tới khi, với gọi hệ thống connet(), một tiến trình client yêu cầu một kết nối tới nó. Nếu client được đáp ứng, do đó, tiến trình server sẽ làm việc kết nối với gọi hệ thống accep() và sự trao đổi thông tin thực sự bắt đầu. Hình 6.10 lad diễn biến của giao thức trao đổi thông tin kiểu socket.

HÇNH 6.10 TRANG 227

Kết thúc trao đổi thông tin kiểu socket với gọi hệ thống thông dụng close(), điều có thể dẫn tới là, liệu các sữ liệu được gởi theo hàng đợi hay được dập tắt(?).

Những kết nối trao đổi thông tin kiểu socket giữa các tiến trình trên các máy tính mạng cục bộ là có thể, và do đó, chúng đem lại một khả năng trao đổi thông tin giữa các tiến trình; quá trình trao đổi này thì độc lập vơi slai lịch tạo lập của tiến trình.

Trao đổi thông tin kiểu định tên đường ống (named pipe):

Một kiểu trao đổi thông tin quan trọng nữa được dẫn tới; đó là kiểu định tên đường ống (named pipe), cũng còn gọi kiểu đường ống (*pipesline):* Trong Unix, MS-DOS và các hệ điều hành tương ứng, mọt đường ống kiểu xây hai hay nhiều tệp tin có thể đọc viế lại với nhau, mà trên đó, một nhóm các tiến trình có thể truy cập đươc. Thật vậy, nếu một tệp tin kiểu như thế được tạo nên một máy tính kết nối mạng, do đó, người ta có thể truy cập hệ thống tệp tin mạng, và vì vậy, một sự trao đổi thông tin giữa các tiến trình của các máy trong mạng được thực hiện.

Thí dụ về trao đổi thông tin kiểu định tên đường ống ở Unix:

Khả năng trao đổi thông tin kiểu pipesline ở trong mạng máy tính thì không dẫn tới với tệp tin NFS (*network file system)*, vì các xâu tệp tin kiểu pipesline ở trong Unix được tạo nên như là những thiết bị đặc biệt bởi gọi hệ thống mknode(). Thuộc vào các tệp tin kiểu pipesline có một bộ kích tạo, mà thay vì sử dụng bộ nhớ đĩa cứng, nó sử dụng bộ đệm của hệ điều hành để lưu trữ trung gian và tạo lập các hàng đợi dữ liệu kiểu FIFO.

Hầu hết các xâu tệp tin liêut pipesline được sử dụng để trao đổi thông tin giữa các tiến trình, khi đó, tất cả các tiến trình được tham gia thì đều ở trên cùng một máy tính. khác với điều này, ở Unix system V, dòng lưu trữ thông tin của các xâu tệp tin kiểu pipesline cho phép một kết nối trao đổi thông tin tới một tiến trình trên một máy tính khác.

Thí dụ về trao đổi thông tin kiểu đường ống (pipeslien) ở Windows NT)

Ở hệ điều hành Windows NT, gọi hệ thống CrêâtNmePipe() được dùng để tạo lập một kết nối trao đổi thông tin kiểu pipesline nhờ một bộ kích tạo đặc biệt. Khác với hệ thống tệp tin NFS, việc truy cập trên đối tượng được phân bổ theo cây thư mục toàn cục ở trên mạng thì có thể được, vì rằng, các tiến trình có thể được trao đổi trên các máy tình khác nhau nhờ các tệp tin kiểu pipesline với các gọi hệ thống Readfile () và WriteFile(). Nếu xâu tệp tin kiểu pipesline được tạo ra trên máy chủ, do đó, tất cả các khách hàng có thể sử dụng nó cho một dịch vụ nào đó.

Một xâu các tệp tin pipéline được mở như một tệp tin bình thường; tuy nhiên, khi đó, một đường dẫn kiểu UNC phải sử dụng có dạng: \\ComputerName\PIPE\PipeName.

Các pipes cục bộ sử dụng ký tự một dấu chấm "." Thay cho tên của máy tính (*Computer name)*

Cấu trúc xâu pipes ở hệ điều hành Windows NT thì phụ thuộc vào giao thức chuyển vận cụ thể được áp dụng. Tuy nhiên, điều này chỉ đúng trong điều kiện các máy tính với hệ điều hành Windows NT hay OS/2; khác với cấu trúc socket, cấu trúc này chỉ tương thích với kiểu socket Unix. Đối với kiểu kết nốipipes thuộc hệ điều hành Unix, ở đó, một tiến trình đặc biệt (thí dụ: bộ điều hành Unix cho mạng LAN) được tạo lập.

Các dịch vụ hộp thư (mailbox):

Khác với kiểu trao đổi thông tin hai chiều *điểm tới điểm* đã đươc nghiên cứu cho tới nay, các dịch vụ hộp thư nói chung cho phép gởi thông tin của một tiến trình tới nhiều tiến trình khác, gọi là phương pháp đa tung *(mutlicast)*, hay phương pháp tung rộng *(brroadcast)*. Kiểu trao đổi thông tin này phù hợp cho việc gởi thư tín (kể cả các tệp tin âm thanh và hình ảnh cũng được chuyển đi) tới người nhận: vấn đề cơ bản của việc nhận là sự tồn tại một hộp thư ở bên nhận; nếu bên gởi không nhân được đáp thư nào gởi tới thì điều đó chứng tỏ, bức thư đã được chuyển tới hộp thư của người nhận. Tuy nhiên, việc nhận những thông tin của máy

chủ vẫn bị trói buộc bởi nhiều giới hạn, mà các giới hạn này thì phụ thuộc mạnh mẽ vào giao thức chuyển vận *(tranaport protocol)*, bởi các lẻ sau đây:

- ✧ Dãy tuần tự các tông tin ở bên gởi thì không giống như bên nhận
- ✧ Việc nhận một thông tin thì chưa hoàn toàn được đảm bảo.

Thí dụ về các lô thư (mails lots) ở trong Winsdows NT:

Ở đây, hộp thư thích hợp cho một lô thư ở tại máy chủ, khi máy chủ được tạo một gọi hệ thống CreateMailslot(). Việc gởi các thông tin sẽ được chuyển đi khi tại các máy khách có gọi hệ thống Writèile() và việc trao đổi thông tin sẽ được kết thúc bởi gọi hệ thống CloseFlie(). Do đó, tên của một lô thư tín với gọi hệ thống Crềatile() có đường dẫn: \\ComputerName\mailslot\MailboxName

Nếu kí tự dấu chấm được dùng cho tên máy tính (*ComputerName)*, do đó, một lô thư tín cục bộ ở trên máy tính được thích hợp với tên MailboxName; còn nếu, tên máy tính được thay thế bằng một cái tên khác, do đó, máy tính này sẽ được viết tên trực tiếp. Nếu tên máy tính là tên máy tỉnh vùng *(domâinNme)*, do đó, các thông tin được gởi đến tất cả các máy tính trong vùng, khi dùng ký tự "□" cho tên máy tính, thì có một trao đỏi thông tin kiểu broadcast (tung rộng) được thực hiện trên mạng; ở đó, tất cả các máy tính được cảm ứng, chúng đều có lô thư tín với tên MailboxName. Còn lại, đó là tên của một dịch vụ; dịch vụ trống đầu tiên sẽ được thông báo tiếp đó.

Vì ở Windows NT, chỉ các dịch vụ biểu đồ lưu thông dữ liệu *(data flow diagram)* không đích xác được sử dụng cho trao đổi thông tin kiểu lô thư, do đó, các giới hạn nói ở trên của hộp thư có giá trị.

Ngoài ra, trong Windows NT, chiều dài lớn nhất của thông tin thì phụ thuộc mạnh mã vào giao thức chuyển vận được sử dụng. Chiều dài này được xác định bởi các gọi hệ thống SendFile() và ReadFile(). Ở giao thức NetDEUI, chiều dài của thông tin trao đổi kiểu điểm tới điểm đạt giá trị lớn nhất 64 kByte, còn kiểu *brroadcast* đạt giá trị max khoảng 400 Byte.

Nói chung, kiểu *mailbox* khó đạt được việc trao đổi thông tin như thiết kể. Dĩ nhiê, nếu người ta giả định có một lớp chuyên vận định hướng kết nối bền vững và có các dịch vụ kiểu biểu đồ lưu thông dữ liệu *(data flow diagram)* đảm bảo cho giao thức chuyển vận; do đó, người ta phải dẫn tới các quy tắc bổ sung, cũng gọi là các giao thức bổ sung: Thông tin có ý nghĩa gì? Lúc nào có một hộp thư có thể được xoá? Và, với các thông tin chưa được trả lời cần giải quyết như thế nào? Từ lý do này, người ta có thể xem các dịch vụ hộp thư như là một trung gian đối với các dịch vụ cao hơn.

Các gọi thủ tục cách quảng (remote procedure calls: RPC)

Trong mạng máy tính, có các dịch vụ mạng diện rộng khác nhau như thư điện tử hoặc in ấn các tệp tin ở trong mạng... Một trong các dịch vụ quan trọng được

nhiều chương trình trợ giúp, đó là dịch vụ gọi thủ tục từ máy chủ qua máy khách. Kiểu gọi tủ tục cách quảng hay kiểu gọi hàm cách quảng *(remote procedure calls: RPC)* hoạt động như một gọi thủ tục cục bộ đơn giản đối với tiến trình gọi; việc thực thi một nhiệm vụ được thực hiện tại đó, nhiệm vụ này đi qua mạng tới một máy tính khác và các kết qủa của nó trở lại qua mạng tới tiến trình gọi và tiếp tục quay vòng.

Việc thiết kế một gọi thủ tục rất quan trọng; nhờ mức độ tư duy cao, kiểu này có thể được bổ sung một cách độc lập với cấu trúc và thiết bị, nó đảm bảo như là một cơ chế vạn năng cho những ứng dụng của hệ thống client/server; cơ chế này cũng hoạt động trong các mạng không đều.

Việc thực thi một gọi hệ thống RPC được sử dụng những thủ tục có tên giống như tên gốc, gọi là thủ tục gốc *(stubprocedure)*, và vì vậy, các thông tin được bó lại và gởi tới máy chủ qua một lớp chuyển vận. Ngoài ra, đối với việc thực thi này, bằng con đường gọi một RPC thì tên của một thủ tục mong muốn được sử dụng cho nó sẽ giống như một tham số.

Tại server, các thông tin được gộp lại và được chuyển cho các thủ tục với một gọi thủ tục bình thường. Các kết quả sẽ di chuyển qua các trạm theo hướng ngược lại. Hình 6.11 chỉ ra một sơ đồ cơ bản cho một gọi RPC.

HÇNH 6.11 TRANG 230

Cũng như các thủ tục gọi thông thường, gọi RPC có hai ấn bản: các gọi RPC bộ ngăn hãm tiến trình gọi, cho tới khi kết quả mong muốn được đem lại; còn các gọi RPC không đồng bộ thông báo tiến trình gọi khi các kết quả RPC dược bày ra. Hình 6.12 mô tả quá trình gọi một RPC đồng bộ.

HÇNH 6.12 TRANG 231

Tại một thủ tục, các khuôn dnạg của số liệu phần mềm và phàn cứng khác nhau cũng tạo ra một vấn đề cần quan tâm; những khuôn dạng này được sử dụng ở các hệ thống máy tính khác nhau. Nếu chúng ta bắt đầu với phần cứng, giả sử có một số nguyên 32 Bit (bao gồm 4 Byte), chúng ta sẽ thực hiện kiểu 1 với các Byte giá trị cao thì bố trí địa chỉ Byte thấp, còn kiểu 2 với các Byte giá trị thấp thì bố trí địa chỉ Byte cao. Hình 6.13 chỉ ra dãy các Byte trên cấu trúc 2 bộ vi xử lý khác nhau.

HÇNH 6.13 TRANG 231

Kích cỡ các số dấu phẩy trôi (theo chuẩn IEEE 1985: phần định trị 23 Bit, số mũ 8 Bit, ký hiệu 1 Bit) rất khác nhau, nó giống như việc mã hoá các chữ cái (mã ASCII và mã EBCDIC).

Lớp để dẫn tới ccs gói tin xuất/nhập và kết quả phải trù tính các nghivấn và phải biến há ccs dữ liệu giữa hình thái lệ thuộc hình dạng ổ đĩa và đại diện của nó khi trao đổi thông tin qua lại.

Một nhiệm vụ tiếp theo là làm thích hợp các dữ liệu khi xếp hàng để tạo trình biên dịch ở trên địa chỉ bộ nhớ, các bộ vi xử lý như thế sẽ từ chối việc truy cập lên một số, nếu con số này không bắt đầu bởi một địa chỉ đúng hay không bắt đàu bằng các chu trình cần thiết. Từ lý do này, các trình biên dịch được xử lý, nếu chúng muốn lưu trữ một số ở một bản ghi theo một chữ cái trước một ký tự trống.

Thí dụ về gọi thủ tục cách quảng ở Unix:

Kiểu gọi RPC ở trong Unix được thực hiện với sự trợ giúo của các thư viện đặc biệt và nó bao gồm 2 lớp: lớp thứ nhất là những cơ chế gọi RPC đối với các thủ tục gốc *(stubprrocedure)*, lớp thứ hai bao gồm việc đóng gói/tháo gói các thông tin với sự trợ giúp của *lớp đại diện dữ liệu nâng cao (extended data representive layer)*

Các thủ tục RPC được gởi trở lại ở các gọi hệ thống cao hơn hay thấp hơn. Lớp cao nhất thì bao gồm gọi hệ thống registerrpc(), mà với gọi hệ thống này, tại server, một dịch vụ (chẳng hạn một thủ tục) được thông báo: cho thủ tục svr_run(), mà với thủ tục này, tiến trình của server bị hãm và chờ đợi gọi hệ thống RPC; cho thủ tục callrpe(), mà với thủ tục này, tiến trình client gọi một thủ tục mong muốn ở server.

Lớp giữa được tạo lập bởi các thủ tục đối với client và server, nhằm điều chỉnh các thông số của giao thức chuyển vận hay điều chỉnh các quyền hạn cho phép.

Gọi hệ thống RPC được thực hiện ở Unix nhờ hệ thống tệp tin mạng; nhiều dịch vụ về NFC sử dụng gọi RPC cho việc thực thi các dịch vụ này.

Trong các gói phần mềm DCE, hệ thống gọi RPC đã đen lại một phương hướng giải quyết khác; ở các gói tin này, các dịch vụ ở trong ngông ngữ giao diện *(interface defintion language:* IDL) được mô tả một cách trừu tượng. Một trình biên dịch chuyển đổi các gọi thủ tục thành các gọi thủ tục gốc và cùng nhau điều chỉnh nhiệm vụ tại một dịch vụ bổ sung của tất cả các server đưa vào sử dụng; do đó, khách hàng hay người lập trình không phải biết trước đó ở máy tính nào, các dịch vụ cần thiết được sử dụng trong mạng (dịch vụ các tệp tin hay dịch vụ toán...).

Thí dụ về gọi thủ thủ tục ở Windows NT:

A.Sinha cho nhận xét (1996): hệ điều hành Windows NT, luôn luôn cần tới mọt công cụ rộng rãi các gọi đồng bộ và không đồng bộ RPCs. Bởi lẻ, các gọi RPCs có thể truy cập tới một tiến trình trên một máy tramh hay trên một máy chủ xác định; lúc đó, hệ thống này được gọi là hệ thống định hướng thiết bị; ngời ra, chúng cũng có thể truy cập tới một thiết bịi xác định trên một máy chủ nào đó; lúc đó, hệ thống này được gọi là hệ thống không có kết nối. Do đó, các cơ chế mức thấp và các giao thức chuyển vận khác nhau sẽ được sử dụng (xem hình 6.41)

hình 6.14 trang 233

Việc tạo dạng cũng như việc đóng gói thông tin đối với gọi RPC được đảm bảo bởi các thủ tục gốc; các thủ tục này sẽ chuyển vận dữ liệu theo *kiểu đại diện dữ liệu mạng (network data representation)*. Các thủ tục gốc không phải do người lập trình tự tạo nên; một trình biên dịch MIDL (*Microsoft RPC IDL computer)* tạo ra mã bổ sung từ cấu hình các tệp tin; mã này được biên dịch và được kết nối với chương trình của máy chủ cũng như máy khách; và thích ứng với các thư viện chuyên dụng trong thời gian xảy ra. Nhờ đó, việc kết nối mạng đối với người lập trình được thông suốt đầy đủ. Những giao thức đặc biệt có thể được lựa chọn qua tiền tố ở trước tên đường dẫn của các dịch vụ mong muốn. Thí dụ để yêu cầu một dịch vụ tại máy tính Myerver ở cổng (port) 2004, giao thức TPC/IP được sử dụng với dòng lệnh: ncacn_ip_cp: Myerver[2004]

6.3. Hệ thống các tệp tin ở trên mạng : (Network Files System: NFS)

Việc truy cập lên các tệp tin dùng chung với sự trợ giúp của mạng là cơ sở quan trọng đối với nhóm công tác và do vậy, đó là một trong các chức năng thiết yếu của kết nối mạng. Do đó, các phần tiếp theo sẽ đem lại một cáci nhìn tổng quát về các vấn đề của hệ thống các tệp tin phân bổ và kể cả các quan hệ của chúng sẽ được nghiên cứu ở trong hệ điều hành Unix và Windows NT. Các hệ thống tệp tin khác không trình bày ở đây.

6.3.1. Ngữ cảnh truy cập *(access sematic)*

Nếu chúng ta khảo sát một tiến trình đơn lẻ ở trong mạng, chúng ta thấy hệ thống tệp tin mạng tương đối đơn giản: để đọc/viết hay để đặt một tệp tin trên một hệ thống tệp tin cục bộ; do đó, điều này sẽ đặt được cả trong dạng các nhiêmh vụ ở một hệ thống khác, khi đó nó thực hiện các nhiệm vụ nay. Và cứ thế tiếp tục tiếp diễn tốt đẹp.

Mặt khác, điều này cho thấy, nếu chúng ta cho phép nhiều tiến trình ở trong hệ thống làm việc trên cùng một tệp tin: sau đó, cái gì sẽ xảy ra? Với trường hợp này có nhiều phương cách để thực thi trong các hệ thống tệp tin mạng khác nhau và được biểu thị là ngữ cảnh truy cập/ chúng ta có thể phân biệt các trường hợp sau đây:

✧ *Tệp tin chỉ đọc:*

Tệp tin có thể chỉ đọc trong mạng. Trường hợp, tất cả các tiến trình chứa đựng các bản sao tệp tin; các bản sao này có thể được nạp vào bộ đệm hay được lưu trữ vào bộ nhớ một cách tuỳ ý, mà không có vấn đề gì.

✧ *Ngữ cảnh tác vụ (operation sematic)*

Các tác vụ để thực hiện các tiến trình ở tệp tin sẽ làm thay đổi dãy các tác vụ đã được thực hiện một cách tuần tự. Nếu gọi hệ thống read() của tiến trình A là gọi hệ thống Write() của tiến trình B; và đồng thời, gọi hệ thống read() của tiến trình A bắt đầu, do đó, tiến trình A sẽ trải qua sự thay đổi ở gọi hệ thống thứ hai, mà sự thay đổi này đã được tiến trình tiếp nhận trên tệp tin. Vì vậy, ở trong hệ điều hành Unix, điều này đã được thực thi, tức là chúng đã tham chiếu như là ngữ cảnh của Unix.

✧ *Ngữ cảnh hội đàm (session sematic):*

Nhiều tiến trình làm việc trên tệp tin, do đó, đầu tiên, tất cả các tiến trình nhận một bản copy của tệp tin, mà bản sao này có thể được các tiến trùnh làm thay đổi. Đầu tiên, nếu một tiến trình đóng lại một tệp tin, do đó, các bản sao được viết trở lại. Điều này có ý nghĩa rằng, ấn bản của tiến trình sau cùng (tức tiến trình đóng tệp tin lại) sẽ xoá và đè lên tất cả những ấn bản khác

✧ *Ngữ cảnh biến động (trấnction sematic):*

Bán phát thảo về hoạt động (mà không thể phân nhỏ được nữa) được sử dụng ở đây, nó có thể tồn tại một cách đầy đủ ở trong dãy tuần tự được chuyên dungnjn hoặc hoàn tàon không tồn tại (xem biến cố nhân tử ở mục 2.4). Nếu chúng ta mở và tu chỉnh một tệp tin, do đó, hệ thống tệp tin mạng sẽ đảm bảo rằng, điều đó sẽ xảy ra một cách độc lập với tất cả các tiến trình khác ở trong quá trình làm việc riêng lẻ. Trong các quá trình này, tệp tin được ngăn hãm việc truy cập nhờ một tiến trình khác.

Quá trình làm việc cụ thể ở một nhóm ngữ cảnh thì phụ thuộc mạnh mẽ vào kiểu nào được đem thực thi. Do đó, dẫn tới vấn đề, những tác dụng cụ thể của một kiểu nào đó còn phụ thuộc vào các bộ đệm và giao thức thực thi.

Nếu chúng ta khảo sát thí dụ ngữ cảnh tác vụ; chúng ta thấy rằng, tiến trình B viết một cái gì đó lên tệp tin, còn tiến trình A đọc tệp tin. Tại thời điểm cả hai nhiệm vụ đạt được máy chủ (quản lý files) theo một dãy tuần tự, mà dãy tuần tự này phụ thuộc vào kiểu mạng được sử dụng,vào giao thức mạng và vào sự chịu tải của ca máy tínhâthm gia. Giả sử, tiến trình A đọc các dữ liệu hiện hành được viết bởi tiến trình B hay phụ thuộc vào việc thực thi, mà không phụ thuộc vào ngữ cảnh hội đàm này.

Vấn đề còn lại là, nếu một ngữ cảnh này được thực thi một cách cục bộ trong một hệ điều hành, thì một ngữ cảnh khác được thực thi trong mạng. Nếu ccs tiến trình tham gia được khoanh vùng một phần lên các máy tính này và một phần lên các máy tính khác ở trong mạng và chúng đều làm việc trên các tệp tin, do đó, người ta phải đặt kế hoạch liên hợp các tiến trình của một hệ thống một cách thận trọng để loại trừ được các chức năng lỗi.

6.3.2. Máy chủ có trạng thái và không có trạng thái

Nói một cách rộng ra, tại các kết nối trao đổi thông tin, chúng ta phải phân biệt, chúng được dẫn tới nhờ một giao thức kết nối (gọi là trao đổi thông tin không có kết nối). Chúng ta có thể tạo ra một sự phân biệt như thế khi truy cập lên hệ thống tệp tin ở trong mạng. Nếu một têph tin được mở, nhằm truy cập lên nó, thì do đó, các cấu trúc tệp tin đặt biệt sẽ được tạo ra khi trao đổi thông tin được định hướng kết nối ở trong máy chủ, các quyền truy cập sẽ được kiểm tra và việc truy cập sẽ được đăng ký. Đối với khách hàng, sự nhận biết này sẽ được chuyển giao, do vậy, tất cả các tác vụ tiếp theo read() và write() với sự nhậnbiết này có thể được xử lý ngay.

Qua đó, máy chủ đã tạo nên một trạng thái xác định; đầu tiên, một sự kết thúc việc truy cập với close() có tác dụng xoá thông tin quản lý và có tác dụng trả lại tự do cho bộ nhớ đã bị chiếm chỗ. Một sự diến biến chứa đựng trạng thái như thế có các ưu điểm sau đây:

✧ *Truy cập nhanh hơn:*

Việc truy cập tiếp theo lên các tệp tin xảy ra nhanh, vì không có các sự chỉ dẫn của địa chỉ người sử dụng tới một trong các nhiệm vụ để đọc/viết, cho nên, chúng sảy ra ngắn hơn và nhanh hơn để chuyển vận qua mạng, và cũng cần thiết phải kiểm tra các chỉ dẫn tới các nhiệm vụ khác, trước khi có thể tiến hành truy cập tệp tin.

✧ *Bộ đệm Cache hiệu nghiệm hơn:*

Vì tất cả các chỉ dẫn về trạng thái truy cập lên máy chủ được dẫn ra, cho nên, các chiến lược về bộ nhớ đệm Cache hiệu nghiệm đối với tệp tin nhằm tăng tốc độ việc truy cập.

✧ *Tránh được các bản sao nhiệm vụ:*

Nhờ việc dẫn tới một số thống tin đối với một tệp tin, các bản sao một nhiệm vụ như thế có thể được thu xếp, mà các bản sao này được tạo trong các náy tính trung gian

✧ *Điều kiện ngăn hãm tệp tin:*

Đối với ngữ cảnh biến động, các tệp tin có thể chấp nhận trạng thái *ngăn hãm*. Đối với ngân hàng dữ liệu, điều đó rất quan trọng.

Tuy nhiên, các máy chủ có bao hàm trạng thái cũng còn tồn tại nhiều vấn đề, mà những vẫn đề này ở trường hợp máy chủ không bao hàm trạng thái lại cho phép:

✧ *Rụng máy tính (client crash)*

Nếu một náy khách bị rụng và với cái đó, chìa khoá truy cập tệp tin bị lác quên, do vậy, tệp tin luôn luôn bỏ nhỏ và các thông tin quản lý chiếm một chỗ không cần thiết.

✧ *Rụng máy chủ (server crash):*

Nếu một máy chủ từ chối, tức là, trạng thái chung của việc xử lý tệp tin bị xoá. Như vậy, máy khách phải nhận biết, tệp tin đang ở trạng thái nào và phải thực hiện tiếp công việc xử lý ở vị trí nào? Điều đó, đòi hỏi những cơ chế khác nhau của phía máy khách và đòi hỏi cả chi phí bổ sung.

✧ *Số lượng các tệp tin được sử dụng đồng thời hay bị giới hạn:*

Việc che phủ bộ nhớ tại máy chủ đối với các thông tin trạng thái đã làm hạn chế số lượng các têph tin được mở đồng thời.

Một máy chủ không có trạng thái là dựa theo nguyên tắc của bộ dung thứ lỗi và không đòi hỏi chi phí để tạo tập một sự kết nối trao đổi thông tin; tại các dịch vụ định hướng trạng thái, nó còn tồn tại nhiều vấn đề như việc ngăn hãm một tệp tin hay việc loại bỏ các bản sao nhiệm vụ...

Thí dụ về máy chủ quản lý hệ thống tệp tin mangnj và bộ điều hành khoá mạng ở Unix:

Máy chủ quản lý hệ thống tập tin NFS thường hay được sử dụng, nhất là đối với máy chủ không trnạg thái, mà nó có thể vừa làm nhiệm vụ máy chủ vừa đồng thời làm máy khác. Gọi open() để mở một têph tin mạng sẽ được dịch ra lênh lookup()ở trên máychủ; lệnh này chọn ra một tham chiếu 32 Byte nhằm giảm nhẹ việc truy cập đối với tệp tin và nạp trên máy khách. Tất cả các trạng thái (như trạng thái hiện hành của tệp tin...) được nạp trên máy khách. Thật vậy, bên cạnh các dự liệu, các gọi hệ thống read() và write() còn gới đi tham chiếu tệp tin máy chủ, trạng thái tệp tin... Đối với việc thực thi ngữ cảnh biến cố, một khả năng ngăn hãm tệp tin phải được tồn tại *(looking)*, và cài gì sẽ không tồn tại ở máy chủ không trạng thái nữa(?). Cho nên, công ty SUN đã quyết định, sau khi xuất hiện hệ thống NFS, cứ 3 năm lại dẫn raviệc khoá tệp tin cho hệ thống NFS. Vì giao thức NFS là kiểu không trạng tháu *(unstate)*, nên dịch vụ này được làm đầy bởi dịch vụ RPC, và được gọi chung là bộ điều hành khoá mạng NML (*network looking manager)*, dịch vụ này đòi hỏi một sự tác dụng tổng hợp với nhiều giao thức và nhiều dịch

vụ. Việc cùng tác dụng này được phân biệt trên các hệ thống Unix khác nhau, vì thỉnh thoảng các cơ chế khoá cục bộ vẫn tồn tại.

Ở hệ điều hành Unix Berkeley, tiến trình lookd đảm nhận phẩm quyền đối với tiến trình *file looking* (khoá tệp tin) tiến trình này tồn tại trong một máy tính và thực hiện các dịch vụ nhờ việc trao đổi thông tin. Hình 6.15 chỉ ra một quá trình như vậy.

hình 6.15 trang 237

Quá trình này bao gồm các bước sau đây :

(1). Chương trình này sử dụng trên máy tính A làm trì hoãn một gọi hệ thống để ngăn hãm phạm vi một tệp tin.
(2). Khoá trình diễn của máy tính A nhận của môđun nhân (của hệ điều hành NFS Client) một nhiệm vụ khoá cho máy tính B.
(3). Khoá trình diễn của máy tính A đăng ký miệm vụ tại một bộ kiểm tra trạng thái của máy tính A bằng một thủ tục RPC (gọi thủ tuc cách quãng)
(4). Bộ kiểm tra trạng thái máy A báo cho bộ kiểm tra trạng thái giao dịch của máy tính B . Máy này nạp tên máy A vào 1 tệp tin /etc/sm
(5). Bây giờ ,nhiệm vụ thực thụ ở tại các khoá trình diễn của máy B được chuyển đi bằng thủ tục RPC.
(6). Khoá trình diễn máy B đăng kí sự ngăn hãm ở bộ kiểm tra máy B.
(7).Cuối cùng khoá trình diễn B ngăn hãm được tẹp tin mong muốn. Tiếp theo, việc thực thi được chỉ ra cho bộ cảm biến nhiệm vụ.

Việc đăng kí bổ sung nhiệm vụ khoá tại bộ trạng thái kiểm tra trạng thái đã tránh khỏi sự nguy hiểm :khi sụp hệ thống, 1 tệp tin vẫn tồn tại trang thái tự do được khoá. Tại thời điểm sụp serve, dịch vụ khoá sẽ thử ngiệm 1 cách nguyên tắc:phải tạo lập trở lại các bảng tồn tại trước đó và dĩ nhiên trên Client và Server đều cùng biện pháp.

Một tình huống khác cũng cần được đề cập, nếu nhiều client đều muốn ngăn hãm 1 tẹp tin, lúc đó, chỉ có 1 client là đạt mỹ mãn .Bây giờ, nếu máy client này bị sụp, thì nhờ việc che phủ tẹp tin của nó, nó sẽ tự hãm cho nó và hãm giúp những client quan tâm khác .Ở đây, chức năng của các bộ kiểm tra được bổ sung thêm :Khi dò hỏi hãm, chúng kiểm tra, liệu tiến trình hãm cho tới đây có còn tồn tại không? Nếu bộ kiểm tra cục bộ viết tiến trình ngăn hãm đã "chết", do đó, việc ngăn hãm trước đó đã bị thủ tiêu và tiến trình dò hỏi nhận tẹp tin dẫn tới hãm .

Người ta lưu ý rằng, giao thức này thì không bảo vệ được các hư hỏng ở trong mạng.Tuy nhiên, một dịch vụ phòng ngừa trên bình điện các máy cục bộ thì có thể đạt được, song chi phí để làm cái đó trong mạng máy tính đơn giản cũng rất cao, do đó, phương pháp này không được dùng.

6.3.3. Vấn đề bộ nhớ Cache

Khi nghiên cứu các tệp tin ảnh xạ bộ nhớ (xem mục 4.4)và việc quản lý xuất nhập (xem mục 5.4), chúng ta nhớ rằng, ở đây, các chiến lược bộ đệm hiệu nghiệm có thể tăng tốc dòng lưu thông dữ liệu 1 cách đáng kể, nhưng vẫn còn 1 số vấn đề có thể tạo ra trở ngại. Đó là các hệ thống tệp tin mạng thì rất khác nhau. Thí dụ, chúng ta lập một bộ đệm ở trên 1 hệ thống máy tính với tiến trình A, do đó, tất cả việc dò hỏi giữa các tiến trình trên các dữ liệu này có thể được trả lời 1 cách nhanh chóng, vì chúng không chất tải lên mạng. Tuy nhiên, các thay đổi tiến trình B ở bản gốc không được quan tâm trên hệ thống máy tính khác.

Ở đây, xuất hiện các vấn đề về kiểu bộ nhớ Cachê, mà chúng ta đã tìm hiểu chúng ở mục 3.5 và 5.4. Hình 6.16 chỉ ra 1ộ cách nhìn tổng quan về hình ảnh các lớp tham gia truyền đạt thông tin. Trên mỗi lớp có thể có một bộ đệm thực thi; đương nhiên, mỗi lớp đều có thể có vấn đề riêng của nó.

hình 6.16 trang 238

Người ta lưu ý rằng, khi làm việc tại tần số cao, cáp nối mạng chất trữ thông tin: với tần số f = 300Hz thì tương ứng tỷ suất truyền đạt 3.10^8 Bit/sec hay tốc độ là hoàn hảo xng điện đạt 3.10^8 m/sec; giả thiết lưu lượng thông tin 1 Bit/sec, nếu bây giờ người ta muốn 256 Bit được liên tục nạp trên cáp mạng thì khi đó khoảng cách tối đa giữa hai máy tính có thể đạt được 256 mét.

Chiến lược cơ bản để giải quyết vấn đề (được trình bày ở trên) là việc áp dụng sự pha trộn các phác thảo đã quen thuộc:

✧ *Viết lưới (write through):*

Nếu các bản sao một tệp tin được thay đổi, do đó, nhiệm vụ thay đổi sẽ được chuyển tới bản sao ở bộ nhớ Cache và tới cả bản gốc ở máy chủ để đáp ứng ngữ cảnh tác vụ.

✧ *Viết trễ (deleyed write):*

Tuy nhiên, thực chất vấn đề là ở chỗ, ưu điểm của bộ nhớ đêmh Cache thì không phải tồn tại việc lưu thông trong mạng. Điều được làm tốt hơn là, nếu người ta thu gom tất cả các nhiệm vụ write() thành một gói tin và được chuyển tới máy chủ, thì điều đó được gọi là viết trễ, lúc đó, ngữ cảnh của việc hợp tác được thay đổi thêm cái gì đó.

✧ *Điều khiển trung tâm (central control):*

Để thay đổi các bản sao trên các máy tính khác nhau một cách hiệu nghiệm, hệ thống bộ nhớ Cache cục bộ phải cùng làm việc với máy tính chủ trung tâm. Điều đó, sẽ xảy ra ngay sau mỗi thay đổi bản gốc, hoặc ngay khi sử dụng tệp tin. Trong trường hợp này, tại mỗi lần đọc, bộ điều hành về Cache sẽ so sánh các dữ liệu tệp

tin (số ấn bản, tổng ngang của bộ đệm...) với các dữ liệu tệp tin của server. Nếu dữ liệu này giống nhau thì việc đọc sẽ được dẫn ra từ bộ đệm Cache; nếu không, thì nhiệm vụ đọc được đưa tiếp tới server.

✧ *Viết khi đóng (write on close)*:

Một quyết định để đi tới ngữ cảnh tác vụ là *ngữ cảnh họp đàm (sesion sematic).* Điều đó đạt được, khi gởi trở lại bản sao đã được thay đổi của tẹp tin sau khi xảy ra tiến trình close() ở máy chủ quản lý tệp tin và đồng thời thực thi bản gốc. Nhiều bản sao của tệp tin gốc có thể tồn tại trên các máy tính khác nhau; vấn đề này phải được quan tâm tại *ngữ cảnh họp đàm,* chứ không cần quan tam nhiều tại bộ nhớ Cache.

Nói chung, điều đó nói lên rằng, việc dẫn vào bộ nhớ Cache trên máy tính Server quản lý tệp tin không những đã tăng gia năng suất, mà còn không hê gây cho cái đó điều gì cả. Ngược lại, việc sẵn sàng thiết đặt bộ nhớ Cache trên hệ thống Client/Server có hiệu nghiệm hơn, mặc dù nó cũng còn mang lại vài vướng mắc khi xảy ra trạng thái thay đổi tệp tin. Từ lý do này, tại ngân hàng dữ liệu phân bổ tồn tại những giao thức ngăn hãm đặt biệt.

Thí dụ về chiến lược bộ nhớ Cache ở hệ thống tệp tin mạng:

Ở hệ điều hành Unix, cơ chế bộ đệm cho việc xuất-nhập các tệp tin mở ra một khả năng ứng dụng ơt hệ thống tệp tin mạng, để thiết đặt sự xuất-nhập các gọi hệ thống RPCs trình diễn xuất-nhập cơ sở biod (*basic Input-Output Demotration),* mà nó yêu cầu bổ sung các khối dữ liệu kế cạnh, gọi là quá trình đọc trước *(read ahead).* Nếu tiến trình ngươi sử dụng đòi hỏi tới khối dữ liệu kế cạnh, do đó, nó sẵn sàng ở trong bộ nhớ đệm và có thể được đọc ngay lập tức. Ngược lại, các dữ liệu của tiến trình write() được chuyển toéi bộ đệm trước tiên và chúng được gộp lại tại các gọi hệ thống flush() (3 trang đối với các dữ liệu, 30 trang đối với các thư mục) cũng như gọi hệ thống sync() (nếu không có bộ đệm nào còn trống) và được viết lên ổ đĩa, gọi là viết trễ *(delayed write).* Nếu tệp tin bị hãm, thì các khối bộ nhớ của nó sẽ hông đệm thêm: Sau khi viết, chúng được gởi ngay tới Server, gọi là viết qua *(write through).* Từ việc lý giả này, thành phần lớn nhất của mã đối với tiến trình biod tồn tại trong nhân hệ điều hành.

6.3.4. Các ý tưởng thực thi:

Có nhiều ý tưởng khác nhau về việc người ta có thể thực thi các hệ thống tệp tinmạng như thế nào đó! Đến nay có hai khả năng phổ biến hay được nói tới, đói là các dịch vụ tệp tin như là các tiến trình đứng riêng lẻ, hay các dịch vụ tệp tin được tạo lập như một bộ khích tạo đặc biệt ở trong nhân hệ điều hành. Hình 6.17 chỉ ra cấu hình khiểu thứ nhất:Ở đây, tiến hình tồn tại ở trên cả hai phía Client và

phía Server. Người ta lươi ý rằng, các lớp của hệ điều hành là đối xứng; sự khác nhau về tư tưởng biểu lộ chỉ ở loại tiến trình nhận và gởi.

hình6.17 trang 240.

Khác với cơ chế vùanêu, cơ chế này chỉ mới dộng chạm tới sự trao đổi thông tin giử các tiến trình; ở hình 6.18.chỉ ra cơ đồ một cơ chế thực thi khác.Đó là cơ chế thực thi trên bình diện độ kích tạo hệ điều hành. Ở đây, phía máy Client thì cho phép làm đúng y như vậy; ngược lại, trên phía máy Server không có tiến trình điều hành tồn tại. Viêc dò hỏi được quản lý trực tiếp ở trong nhân hệ điều hành và được chuyển hướng trên hệ thống tệp tin cục bộ.

Phương pháp này thực thi hệ thống tệp tin mạng của Server trên bộ kích tạo và tránh được các bản sao tệp tin từ gọi hệ thống ở trong khong khoảng địa chỉ của một tiến tạo Server và quay trở lại bộ đệm hệ thống để viết vào hệ thống tệp tin cục bộ của Server.

hình 6.18 trang 241

Tuy nhiên, cấu hình này thì không đối xứng. Hệ thống tệp tin mạng thì không còn mang theo một tiến trình đại diện, còn sự trao đổi thông tin giữa các tiến trình vẫn xảu ra trôi chảy; tức là, hệ thống tệp tin mạng chỉ mang theo các lớp đặc biệt (tức bộ kích tạo) của nhân hệ điều hành. Cả hai ý tưởng được thực hiện khác nhau.

Thí dụ về hệ thống tệp tin mạng của Unix:

Ở hệ thống tệp tin của Unix có hai giao thức: một giao thức dùng để điều chỉnh việc móc nối của hệ thống tệp tin server tới một thư mục trong client, và giao thức kia dùng để xúc tiến quá trình lưu thông đọc/viết. Giao thức thứ nhất được thực hiện giữa hai tiến trình: tiến trình dò hỏi client (khi khởi động hệ thống tệp tin) và tiến trình diễn mountd *(mountd demontration)* tại server. Theo đó, tiến trình mountd của client gởi một thông tin tới mountd của server; khi đó, với một hệ thống getfh(), máy chủ lấy một tệp tin quản lý *(file handle)* của lớp hệ thống tệp tin mạng của server và đưa trở lại tiến trình client. Bây giờ, tiến trình này thực hiện một hệ thống mount() và kẹp chặt hệ thống tệp tin của server ở lớp NFS-client của nhân hệ điều hành.

Giao thức thứ hai của quá trình đọc/viết trên các tệp tin được thay đổi ở trong các nhân hệ điều hành. Tuy nhiên, trên server, chương trình nfsd được khởi động như là một tiến trình, tiếp theo diễn ra gọi hệ thống nfs_svc() ở trạng thái nhân, từ đó, nó không còn quay trở lại nữa. Phương pháp này thực thi hệ thống tệp tin mạng trên bình diện bộ kích tạo. Hình 6.19 chỉ cho thấy cấu trúc cơ bản này.

Lớp bổ sung đối với hệ thống tệp tin ảo đóng vai trò rất quan trọng, hệ thống này đã thấy trước các tác vụ hệ thống tệp tin. Tất cả các tác vụ ở trong hệ thống

tệp tin là các phương pháp đối tượng *nút chỉ số ảo* và chúng được thực thi sau mỗi đối tượng.

Do đó, một sự đổi chiều giữa các tệp tin MS-DOS, các tệp tin Unix và các tệp tin NFS thì hoàn toàn không có vấn đề gì.

hình 6.19 trang 242

Việc thực thi các dịch vụ, hệ thống tệp tin mạng của Windows NT thì cũng xuất phát từ các cơ sở gống như việc thu xếp hệ thống tệp tin mạng ở trong nhân hệ điều hành và cũng tạo nên kiểu bộ kích tạo; hình 6.20 cho thấy một cái nhìn tổng quan về điều đó. Ở trong mạng, bộ kích tạo gởi lặp *(redirector driver)* có nhiệm vụ dò hỏi hệ thống tệp tin ở trong mạng: xem thử loại tệp tin mong muốn có tồn tại hay không và cái gì đang xảy ra tại các nhiệm vụ của tệp tin cục bộ? Do đó, bộ kích tạo phải tạo lại sự kết nối mạng qua lớp chuyển vận *(transport layer).* Và như vậy, các thông tin trạng thái (thí dụ: những tệp tin nào sẽ được mở...) được gởi lặp *(redirector)* quản lý và thực hiện.

hình 6.20 trang 242.

Sự trợ giúp việc xuất-nhập không đòng bộ đòi hỏi phải có những cơ chế riêng lẻ, thì sau khi thực hiện không đồng bộ nhiệm vụ, tiến trình nhận được sự điều khiển và được đánh thức; tiếp đến, hệ điều hành Windows NT sử dụng một vùng các tiến trình thread *(pool of threads)* để thực hiện các thủ tục kết thúc cho một gói tin yêu cầu xuất-nhập.

Bộ gởi lặp phát triển các thông tin mạng của nó qua giao diện bộ kích tạo chuyển vận *(transport driver interface)*, mà các kênh chuyển vận mạch áo của nó *(virtual circuit)* được thực hiện nhờ hệ thống chuyển vận được thiết đặt cho cái đó.

Lớp máy chủ quản lý tệp tin được tạo lập như là một bộ kích tạo; nó tương thích với các giao thức khối thông điệp máy chủ SMB (*server message block protocol)* của mạng Novell hay các phần mềm điều hành mạng LAN.

6.3.5. Vấn đề an toàn

Điều rõ ràng là, ở trong mạng, sự an toàn để bảo vệ các tệp tin không chỉ có hệ thống tệp tin trong mạng tự bảo vệ cho mình, mà tất cả việc kiểm tra an toàn phải được tạo lập và được thực hiện một cách riêng biệt.

Cái khó khăn nhất là việc kết nối lại với nhau các hệ thống tệp tin với các cơ chế bảo vệ tệp tin và các mức báo vệ tệp tin khác nhau. Thí dụ, điều đó gây nên khó khăn cho việc mô phỏng các tệp tin có tên dài như ở các server với các hệ điều hành Unix và Windows NT cũng như các hệ thống tệp tin mạng với hệ thống tệp tin MS-DOS có tên ngắn (nhiều nhất là 8 ký tự); ngoài ra, các quyền truy cập của danh sách điều khiển truy cập ACL *access control list)* của hệ điều hành Windows

NT cũng được mô phỏng vô cùng khó khăn trên các quyền truy cập đơn giản của hệ điều hành Unix, vì ở đó, các kiểu ACL không tồn tại.

Sự quá độ của hệ thống an toàn này trong một hệ thống khác của một hệ thống tệp tin mạng thì hầu như chỉ tạo ra nhiều bất định và mâu thuẫn.

Hệ thống an toàn các tệp tin mạng ở Unix:

Ở hệ thống tệp tin mạng của hệ điều hành Unix, tất cả người sử dụng của hệ thống được dẫn ra trong một danh sách đặc biệt; trong công nghệ máy tính, danh sách này gọi là danh bạ mậu dịch hay các trang vàng *(yellow papes)*; tức là nó cũng giống như việc mọi người dân được dẫn vào các trang của một sổ điện thoại của thành phố vậy. Danh sách này sẽ được quản lý bởi một tiến trình đặc biệt, gọi là hệ thống thông tin mạng NIS (*network information system)*; ngoài các trường thông tin như tên người sử dụng *(user name)* và mật lệnh *(password),* danh sách còn chứa đựng các sự điền vào như chìa khoá mã hiện hành cho việc trao đổi dữ liệu giữa client và server, các nhóm người sử dụng và các cơ chế khác về an toàn toàn hệ thống tệp tin mạng.

Hệ thống các tệp tin mạng được sử dụng cho các yêu cầu về dịch vụ của gọi hệ thống RPC. Nếu người ta coi các môđun về bản quyền Unix như là giấy chứng chỉ an toàn, do đó giấy chứng chỉ an toàn về gọi thủ tục cách quảng RCL chứa đựng những hiểu biết về truy cập hệ điều hành Unix của định ngiã giao diện người sử dụng và định nghĩa giao diện nhóm công tác. Nếu cả hai hệ thống Client và Server có các cơ chế an toàn giống nhau, thì ở đây vẫn có thể tồn tại các lỗ khuyết an toàn. Thí dụ, một người quản lý client thì không thể tự động thay đổi các tệp tin của người quản lý server, nếu định nghĩa giao diện trên cả hai hệ thống giống nhau. Từ lý do này, tất cả việc dò hỏi trên Server phải được quy định cho mỗi người quản lý nhóm công tác, nhưng phải tuân thủ những cơ chế an toàn thông thường.

Ngoài ra, hệ thống thông tin mạng tạo điều kiện để điều chỉnh sự tương ứng của người sử dụng trên hệ thống tệp tin của Server, để sao cho không xuất hiện sự cắt ngắn và không ổn định của các tệp tin của các người sử dụng khác nhau.

Mặc dù đã có những nổ lực khác nhau, hệ thống an toàn cho hệ thống tệp tin mạng vẫn còn chứa đựng vài lỗ hỏng. Do đó, đối với các yêu cầu tiếp theo, việc dẫn giải đầy đủ sẽ được trình bày trong chương tiếp theo.

An toàn hệ thống tệp tin ở Windows NT:

Các tệp tin ở trong Windows NT được trình bày trong mục 4.3; ở đó, chúng được thực thi bởi các danh sách điều khiển truy cập ACL; nhờ đó, các quyền truy cập đối với người sử dụng và nhóm công tác được điều chỉnh. Những cơ chế an toàn luôn luôn được thiết đặt sẵn trên Fileserver. Theo đó, mỗi người sử dụng có thể mở tệp tin; khi đó, anh ta phải đăng ký vào tệp tin người sử dụng của bộ điều hành tài khoản bảo an *(security account manager:* SAM) ở trên máy tính Client

cũng như trên máy tính Server. Nếu việc đăng lý có lỗi hay các mật lệnh khoá trên Client và Server không đồng nhất, do đó, việc truy cập lên các tệp tin bị từ chối một cách nguyên tắc. Nhưng mà, tại lần đăng ký đúng, thì các giới hạn truy câpn của danh sách điều khiển truy cập vẫn luôn luôn có giá trị như đối với hệ thống tệp tin cục bộ.

Người ta cũng lưu ý, bản phác thảo này vẫn chưa mang lại một sự điều khiển tự động của người quản lý Client đối với Server, khi cả hai vẫn còn sử dụng mật lệnh khác nhau.

6.4. Các kiểu làm việc ở trong mạng

Chúng ta đã biết, các hệ thống đa vi xử lý tạo ra một sự phân bổ công việc một cách dễ dàng nhờ *mã thực thi song song* và các bộ vi xử lý có cùng kiểu. Ngược lại với điều đó, ở các máy tính mạng, ngườ ta cần phải lưu ý thêm các vấn đề khác nhau sau đây:

- Việc chuyển đạt các dữ liệu qua mạng cần tiêu tốn một thời gian, do đó, các đoạn mã có lợi cho việc loại bỏ cả hai trường hợp không có hay có ít thông tin.
- Nếu việc chuyển dịch mật mã nhằm mục đích cần tới những gói thông tin lớn hơn, do đó, chi phí và năng suất có quan hệ mật thiết với nhau.
- Hầu hết các mạng máy tính được tạo lập từ các máy tính không đồng đều của các nhà sản xuất khác nhau và các hệ điều hành khác nhau. Do đó, một chương trình thực thi trên một máy tính thì không phải được thực thi trên tất cả các máy tính khác ở trong mạng.

Qua đó, người ta nhận thấy: hệ thống điều hành Job phải được giải quyết thoả đáng các vấn đề đã nêu ở trên; bởi vì hệ thống này không phải là các máy tính đơn mà nó là một mạng các máy tính.

6.4.1. Điều hành Job (*Job management)*

Ngày nay, trong công nghệ máy tính có những hệ thống điều hành Job khác nhau (xem mục2.2), mà chúng cho phép sử dụng thời gian nhàn rỗi của tập hợp các máy trạm. Loại hệ thống này được sử dụng trong nhiều lĩnh vực nghiên cứu và phát triển ở đó cần dùng tới những máy tính siêu hạng. Những yêu cầu đối với một hệ thống *các phương tiện dùng chung tải (load sharing facility system)* thì bao gồm những điểm chính sau đây:

- Việc phân bổ tải đối với tất các loại Job và các cấp công suất cần phải được thuyên chuyển qua lại.
- Đối với các yêu cầu khác nhau về phương tiện điều hành như thời gian tính toán và nhu cầu bộ nhớ, thì các hành đợi tập trung khác nhau cần thiết được dẫn ra.
- Thời gian thực hiện Job cần thiết phải được tối ưu.

- Các bản quyền của các chương trình cần phải được điều hành một cách thấu suốt.
- Mặc dù có tải bổ sung, việc quản lý các trạm máy (*workstations)* phải không có biến động xảy ra.
- Trong việc liên kết tải, các máy trạm phải được sắn sàng phục vụ với khả năng cao nhất, dù là ban đêm, ngày cuối tuần hay ngày lễ.
- Cấu hình chung phải rõ ràng và việc quản lý phải được thuận lợi.

Thực hiện những yêu cầu này không phải đơn giản, vì hệ thống điều hành Job không gây ảnh hưởng lên tất cả các nhân tố. Tuy nhiên, nó có thể đơn giản hoá việc quản lý hệ thống bởi một giao diện người sử dụng tốt và các cơ chế cấu hình đơn giản; nó có thể sử dụng nhiều giờ liền vào ngày lễ nhờ việc *xử lý theo đợt (batch processing)* và nó có thể di chuyển Job trên các máy tính thông qua việc trao đổi thống tin trong mạng; tuy nhiên, việc thu xếp các Job trên các máy tính thì không phải tuỳ tiện. Do đó, người ta không chỉ lưu ý tới các vấn đề như nhu cầu bộ nhớ, kiểu bộ vi xử lý, kiểu hệ điều hành và nhu cầu ổ đĩa cục bộ đối với các tệp tin; người ta còn lưu ý tới bản quyền của một số chương trình. Mà chúng được kết nối với các máy tính các định (gọi là kiểu khoá nut: *node locking)* hoặc chỉ một số lượng xác định các máy tính trong nhóm có thể được thực hiện đồng thời (gọi là kiểu bản quyền nổi: *floating license)*

Sự ưu tiên của các Job phải được lựa chọn sao cho người sử dụng một máy tính thực thụ luôn luôn có một bước đi trước và vì thế, khi sử dụng máy tính này, anh ta không cần phải lưu ý gì cả.

6.4.2. Máy tính mạng hai chức năng *(netcomputer)*

Một bản phát thảo đầy đủ khác đối với các hệ thống mạng phân bổ, đó là bản phát thảo về máy tính mạng do các công ty máy tính nôi tiếng phát triển (Oracle, Sun, IBM, Apple...). Nếu một máy tính bình thường tự chứa đựng tất cả các thành phần của hệ điều hành để có thể làm việc với các ổ đĩa, máy in... một cách tự chủ; do đó, ở các hệ thống với hệ điều hành phân bổ, các chức năng hệ điều hành khác giống như các dịch vụ chuyên dụng (như dịch vụ file, dịch vụ tiến trình...) được nạp lên một máy tính, được chuyên môn hoá.

Bản phác thảo về một máy tính vừa làm hai nhiệm vụ Client/Server này của hệ điều hành phân bổ được vận hành mạnh mẽ ở các máy tính mạng. Ngoài một bộ nhớ chính, một bộ vi xử lý, một card kết nối mạng và một màn hình, máy tính này không còn chứa đựng gì thêm. Cũng như các phần mềm của hệ điều hành (bao gồm các chương trình dịch vụ, nhân của hệ điều hành...), ở máy tính mạng hai chức năng này, chúng được thu gọn thành một vi nhân (*microkernel)*; vi nhân là nhân của hệ điều hành được thu tóm từ các lớp kết nối mạng cũng như các chức năng nạp chương trình. Tất cả các chương trình đều nằm trên ổ đĩa cứng cục bộ và đều được dẫn tới từ mạng. Và cũng giống như chức năng hệ điều hành, ở máy tính này, vi nhân có thể được nạp cho bộ tải tự khởi động ROM khi đóng nguồn điện

cho máy ở trong mạng. Tất cả dung lượng lưu trữ các tệp tin, các dịch vụ in ấn... được đảm nhiệm bởi một dịch vụ chuyên dụng.

Đối lập với mạng các máy tính đang dùng, việc điều hành một vi nhân như vậy mang lại cho chúng ta những ưu điểm sau đây:

- *Các tệp tin thực thi:*

Nhờ việc quản lý tập trung các dữ liệu được sử dụng, các tệp tin này thì luôn luôn thực thi ổn định. Đồng thời, chúng cũng quan hệ với các thành phần của hệ điều hành (mà các thành phần này đã được nạp vào mạng như bộ kích tạo, các chương trình dịch vụ...), với các chương trình và các tệp tin của người sử dụng

- *Các phần cứng lẻ tiền hơn:*

Máy tính mạng hiai chức năng thường sử dụng ít phần cứng hơn, vì các thành phần hệ điều hành xác định và không gian ổ đĩa cục bộ loại bỏ được các chương trình dịch vụ. Điều đó làm cho máy mạng này lẻ hơn khi mua sắm.

- *Việc quản lý tốn kém hơn:*

Việc chăm sóc các phần mềm hệ thống, việc chuyển giao bản quyền và cấu hình các máy mạng hai chức năng được thực hiện và chăm sóc một lần. Điều đó cho thấy, việc quản lý máy mạng hai chức năng đem lại một sự hao tốn ít hơn, tức là điều đó đã đem lại một cái gì đó không phải là không đáng kể: Người ta tính toàn rằng chi phí hằng năm cho việc chăm sóc bảo dưỡng cho phần mềm và cấu hành hệ thống cho một máy tính đúng bằng trị giá mua sắm thiết bị đó. Ngoài ra, việc sửa chữa các phần cứng cũng đơn giản hơn, vì các thành phần của máy tính có thể được trao đổi một cách tuỳ ý. Tất cả các tệp tin và các giao diện của người sử dụng đều nằm trên máy tính trung tâm và không bao giờ bị đánh mất.

- *An toàn dữ liệu cao hơn:*

Nhờ việc đảm bảo dữ liệu tập trung, khi một máy bị sụp, công việc của mọi nhân viên khác trong mạng vẫn tiến hành tiếp tục. Kể cả việc đánh cắp thông tin cũng được đảm bảo nhờ một cơ cấu ngoại vi.

- *Sử dụng tốt hơn các nguồn tài nguyên*

Bên cạnh việc sử dụng tốt hơn các máy in, các thiết bị FAX, người ta có thể đạt được nhờ các kết nối mạng; thật vậy, ở máy tính mạng hai chức năng, người ta có thể sử dụng một cách tốt hơn các bộ nhớ quảng đại (ổ đĩa cứng, ổ đĩa mềm, băng từ...), vì chúng được tồn tại trong vùng quản lý tập trung. Ngoài ra, những việc mở rộng cần thiết máy tính này có thể định hướng tốt hơn tới những nhu cầu mong muốn.

Tuy nhiên, bản phát thảo này cũng còn những tồn tại nhất định. Sau đây, chúng ta phân tích thêm một vài điểm nữa để thấy rõ phạm vi sử dụng của nó trong tương lai:

- *Chi phí mạng có gia tăng:*

Nếu tất cả các trình tiện dụng đều chạy trên mạng, do đó, mạng nội hạt *(intranet)* và các dịch vụ được sử dụng sẽ làm gia tăng đáng kể công suất của nó (lưu lượng thông tin, dung lượng bộ nhớ...). Tức là phải tăng chi phí thêm cho cái đó.

- *Chi phí bộ đệm có gia tăng:*

Ở loại máy tính này, ngươi ta còn thay thế các phần cứng (như bộ nhớ chính, bộ nhớ quảng đại) để lưu trữ các dữ liệu hay các chương trình lớn để có thời gian nạp và tải mạng nhanh hơn, điều này rất cần thiết đối với các thiết bị đầu cuối của hệ điều hành Unix; và do đó, lợi thế về giá cả của loại này không thể giảm nhhiều được.

- *Chi phí của người sử dụng có gia tăng:*

Do việc quản lý tập trung các phần mềm và việc điều hành các phần cứng của Trung tâm EDV, người sử dụng cảm thâyd phải trả thêm chi phí thời gian sử dụng trên mạng: Vì chỉ có trung tâm mới quyết định, chương trình này có thể được sử dụng và khoảng bao nhiêu không gian bộ nhớ được phân bổ cho riêng từng chương trình? Khoảng thời gian này, trung tân cũng tính trong bảng thanh toán hàng tháng.

Cho đến nay (1999) sự đón nhận loại máy tính mạng hai chức này chưa được dùng phổ biến; nhưng trong tương lai, nó sẽ là một hệ thống được dùng năng động trong mọi lĩnh vực của đời sống con người.

Một điều quan trọng nữa về máy tính mạng hai chức năng là nó cho phép mỗi chỗ làm việc có cấu hành riêng lẻ; nhờ việc quản lý tập trung đối với cácn chương trình chuẩn (*standarprogramms)* mà hầu hết những vấn đề về cấu hình được loại trừ tại chỗ làm việc và nó quan tâm đầy đủ đối với các tệp tin cần thiết. Do đó, có thể phân biệt chúng thành 2 loại vấn đề:

- *Thực thi tập trung:*

Một cách tổng quát, người điều hành mangnj có quyền hạn thay đổi một cách tích cực cấu hình của một máy tính nối mạng. Điều này đòi hỏi những hiểu biết tối thiểu về một sự thay đói này, thí dụ, phải nắm rõ các dữ liệu cần thiết cho từng máy tính riêng biệt. Tuy nhiên, một hệ thống mạng không đồng nhất với các phần cứng và phần mềm khác nhau là một trở ngại lớn đối với người sử dụng.

- *Thực thi phân tán:*

Vì lý do vừa nêu, bên cạnh các chương trình đặc biệt và các khả năng dịch vụ tiêu chuẩn của mình, các máy tính làm việc riêng lẻ (trong mangnj) có thể chứa đựng thêm một phần hệ thống tệp tin mà không làm ảnh hưởng tới người sử dụng; phần này đươc hiệu chuân một cách đều đặn về cấu hình cũng như các trữ lượng dữ liệu của một hay nhiều dịch vụ chuyên dụng trong mạng. Nếu trữ lượng các tệp tin cần thiết và các dich vụ (dịch vụ chương trình, dịch vụ đối tượng...) được gắn một cách sáng tạo vào các công cụ của máy tình. Như vậy, máy tính này làm việc như là một máy tính mạng hai chức năng; bởi vì, nó có các thành phần hệ điều hành và các chương trình tiện dụng cần thiết ở bộ đệm Cache nhờ thế giảm được thời gian truy cập. Do đó, tiến trình đại diện của việc thực thi phải có chức năng cảnh giới và chức năng này được tồn tại ở trên Cache (xem mục 3.5).

Phạm vi ứng dụng của ngôn ngư JAVA:

Vấn đề sáng tỏ nhất của bản phác thảo về máy mạng hai chức năng-đó là sự không tương thích của các phần cứng và phần mềm ở các máy tính khác nhau của các nhà sản xuất khác nhau. Để giải quyết vấn đề này, một ngôn ngữ lập trình thống nhất được sử dụng, gọi là ngôn ngữ lập trình JAVA. Đó là một ngôn ngữ lập trình hướng đối tượng, là sản phẩm phần mềm của hãng máy tính SUN (1997). Về mặt cú pháp, ngôn ngữ này cũng tương tự như ngôn ngữ lập trình C^{++}; nhưng, nó có sự đơn giản hơn vì không cần bộ chỉ thị *(pointer)*, không viết chồng nhiều lần và bản thiết kế giao diện tốt hơn.

Khi thực hiện chương trình, các máy mạng được trợ giúp bởi kết quả các lệnh máy tính ảo; máy ảo này gọi là máy ảo JAVA. Nhiệm vụ của bộ vi xử lý máy mạng hai chức năng (cũng như các thiết bị khác cần thực hiện bằng mã JAVA) là để quy mô phóng máy ảo này. Khi đó, nhiệm vụ của hệ điều hành của một máy mạng này bao gồm việc thiết lập vùng diễn biến các chương trình JAVA được nạp trên mạng. Sau đây là vài đặc điểm đặc biệt cần lưu ý:

- ♦ Việc quản lý bộ nhớ chính được thực hiện một cách tự động để dành không gian nhớ cho các đối tượng như thực hiện việc dẫn đường hoặc loại bỏ các đối tượng không sử dụng gọi là sưu tập rác *(garbarge collection)*.
- ♦ Thực hiện việc cách ly các chương trình chạy đồng thời với nhau hay với hệ thống còn lại. Để tạo điều kiện cho việc thực thi nhanh chóng và đơn giản, trong JAVA, không có bộ chỉ thị (pointer) địa chỉ, và do đó, có khả năng hoạt động của các chương trình mạng và các ứng dụng trên máy tình cũng bị hạn chế.
- ♦ Sự thông dịch các mãByte của máy ảo JAVA không được mô tả trực tiếp. Đặc biệt, các kiểu dữ liệu JAVA phải được phù hợp với chiều dài của máy tính đích.
- ♦ Việc thiết lập các chức năng tiêu chuẩn đối với đồ hoạ hay việc xuất-nhập thì có thể cho phép đối với chương trình mạng hay các chương trình tiện

dụng. Một cách phổ dụng, các chương trình mạng không được truy cập lên các ổ đĩa cục bộ hay máy in.

Sự cần thiết của một ngôn ngữ lập trình đơn giản và an toàn đối với mạng máy tính đã dẫn tới việc hoàn thiện các ứng dụng của các hệ thống siêu text trên trang WEB của mạng internet. Chức năng của một trang text đơn giản không đem lại đươc nhiều sự mời chào. Để luôn luôn gia tăng các chức năng mới trong các chương trình mô tả siêu text (WEB browser), điều tốt hơn là, nạp trực tiếp mã đối với một chức năng đặc biệt của người yêu cầu, hay thực hiện một cách cục bộ nhờ một trình thông dịch JAVA được tích hợp ở trong chương trình WEB browser. Mã JAVA thực hiện những yêu cầu này nhờ đặc điểm lệnh máy đã được thiết đặt theo tiêu chuẩn và nhờ thư viện thời gian chạy đã được thực hiện một cách linh hoạt hơn các chức năng của chúng, tức là, các giao thức và các chức năng mới có thể được tạo lập nhờ việc nạp các chương trình JAVA hoàn hảo (*content handlers)*, mà không cần viết mới trình đọc lướt browser.

Nhược điểm của phương pháp này là ở chỗ, không có một nhiệm vụ nào to tát có thể được làm hoàn hảo vì lý do những điều kiện giới hạn của một ứng dụng. Xuất phát từ lý do an toàn, việc truy cập các tệp tin như các tệp tin tạm thời và các nguồn tài nguyên cục bộ khác thì không được phép, và do đó, các ứng dụng được giới hạn trên phạm vi của nó khi không có các chức năng trợ giúp đối với trình browser.

6.5.Các cơ chế an toàn và các thao tác trên mạng

Những câu hỏi về an toàn ở các hệ thống máy tính không phải là những vấn đề kỷ thuật mà chỉ là những vấn đề về con người. Một máy tính hoạt động không có những thiết bị an toàn thì về mặt kỹ thuật không có vấn đề gì. Nhưng nếu, một máy tính chưa đươc làm cô lập và lại được điều khiển bởi người sử dụng ít chuyên môn, đặt biệt, ở một mạng máy tính phân bổ có nhiều người sử dụng đăng ký là khách (tức không biết rõ địa chỉ của nó), khi đó, vấn đề an toàn cần được đàm luận và cần phải tính tới. Một trong các nhiêmh vụ quan trọng của việc quản lý hệ thống là bao gồm việc nắm vững các hệ thống này, việc chi dùng về thời gian và tiền bạc, nhằm nâng cao chúng hay tối thiểu để lựa chọn được những biện pháp thích hợp. Những xoay chuyển cần thiết đã đụng chạm tới các cơ chế an toàn của hệ điều hành. Từ lý do đã nêu, trong mục này, chúng ta sẽ đề cập tới những sự cố có thể xảy ra và cả biện pháp khắc phục.

6.5.1. Lịch sử cận đại về an toàn trên mạng máy tính

Từ sau tháng 11 năm 1988, một chương trình (làm nhiệm vụ tái tạo) được nạp phần lớn các máy tính kết nối mạng internet bằng hệ điều hành Unix ở trường đại học Carnergie-Mellon (Mỹ); tại đó, nhóm đáp ứng khẩn cấp CERT (*conputer emergency response team)* được thành lập; nhóm này thực nghiệm một cách hệ

thống để phát hiện, cung cấp các tài liệu khảo cứu và thanh trừng sai sót về an toàn ở hệ thống máy tính. chỉ tính năm 1997, nhóm CERT đã tổng kết có khoảng 4.000 vụ tử nghiêmh bẻ khoá đột nhập, với khoảng 30.000 đến 40.000 người tấn công (*hacker);* những người của nhóm này luôn luôn phát hiện ra những sai sót về an toàn trong các hệ thống, họ làm điều đó nhanh hơn những người quản lý hệ thống muốn hoặc có thể thanh trừng chúng. Tuy chiên, Hội đồng nghiên cứu quốc gia Mỹ vẫn đưa ra nhwnj định: "Bọn kẻ cắp hiệnđại bằng máy tính, chúng có thể ăn cắp được nhiều hơn bằng các vũ khí lợi hại khác". Hội đồng này còn cảnh cáo: "Kẻ khúng bố ngày mai không đặt nhiều bom, mà chúng chỉ ấn nút bàn phím và chỉ thế nó tạo vô vàn thiệt hại". Cho đến nay, công việc làm đươc rất ít để có thể kiến trúc một hệ thống máy tính an toàn.

Trong quan hệ với cải đó, điều cần đươc quan tâm là, Bộ quốc phòng Mỹ đã điều hành các nhóm công tác để chuẩn bị tiến hành cuộc chiến trnah hạn chế những tổn thất lớn nhất do hệ thống máy tính kẻ thù gây nên. Điều đó, không có nghĩa chỉ bao vây các vụ ăn trộm bẻ khoá mật lệnh ở trong máy tính kết nối mạng với việc lôi kéo hay tẩy bỏ các tệp tin quan trọng, đặc biệt, còn bao vây các vụ khống chế phần cứng của các máy tính ở trong mạng. Thí dụ, nhờ việc thực hiện lặp các định hướng xác định, bọn chúng có thể hun nóng các thành phần xác định của một bộ vi xử lý.

Dự đoán cho cuộc tấn công này trên các hệ thống máy tính là sự xâm nhập vào các dữ liệu ở một máy tính nối mạng. Điều này có thể xảy ra trên nhiều phương cách khác nhau, sau đây chúng ta sẽ lần lượt đề cập tới những vấn đề này.

6.5.2. Đột nhập qua mạng

Một trong các khả năng dễ dàng để đột nhập vào các mạng máy tính là việc sử dụng máy tính qua mạng, thí dụ, bằng một chương trình chạy trên mạng điện thoại thông thường Telnet. Để dẫn tới việc điều khiển lối vào mạng qua mật lệnh *(password)*, vấn đề này đươc các tên trộm giải quyết bằng các phương pháp khác nhau như sau:

Giải mật lệnh (password guessing):

Một trong các kiểu lỗi đươc dẫn tới là do việc không đươc thay đổi các mật lệnh chuẩn (*standardpassword)* của nhà sản xuất hệ điều hành đối với công việc quản lý hệ thống; vì nếu, mật lệnh bị quên thì phải đọc lại bản hướng dẫn dài dòng. Nếu các mật lệnh đã trở nên phổ biến, thì điều chẳng có gì lạ, từ bên ngoài kẻ trộm sẽ dễ dàng xâm nhập vào hệ thống.

Một lỗi nữa của người sử dụng hợp pháp cũng thường hay xảy ra, đó là việc sử dụng những từ thông thường để làm mật lệnh (như tên người, tên thành phố, tên quốc gia hay loại cây cối, loại động vật...). Nếu mật lệnh khoá tệp tin có thể đọc dễ dàng (thí dụ trong các hệ thống Unix với mật lệnh /etd/passwd) do đó, một du

khách quỷ quái (tìm cách đăng ký vào mục guest dành cho khách mới nhập) có thể sao chép một tệp tin, và với cơ chế khoá mật lệnh, hắn ta có thể bẻ khóa bằng các sự điền vào của một sổ từ (nếu cần 250.000 lần điền vào và giả sử 1 miligiây cho 1 lần điền vào, thì công việc bẻ khoá chỉ cần khoảng hoăn 4 phút!). Ngoài ra, số phòng ngủ hay số điện thoại dùng làm mật lệnh cũng giải được một cách dễ dàng.

Những phương tiện đối phó như dùng các mật lệnh không có quy luật, phương pháp cưỡng bức hay ngẫu nhiên cũng không trợ giúp đươc; vì người sử dụng quên hay viết chúng vào trang cuối sổ ghi chép thì có thể thấy rõ ràng sẽ dẫn tới sự thất lạc. Ngay cả việc thay đổi thường xuyên đều đặn mật lệnh cũng vẫn không trợ giúp được mấy; chỉ có người sử dụng phải phòng chống bằng cách luyện tập thường xuyên mật lệnh mới và đánh lừa hệ thống là đang dùng mật lệnh cũ.

Một khả năng để loại trừ những mật lệnh quá đơn giản, đó là ngay khi nhập vào, ngươ sử dụng phải kiểm tra mật lệnh mới, liệu nó có bị giải một cách dễ dàng không?

Dò thám mật lệnh (passwordspying):

Nếu có một kẻ đột nhập giành được sự điều khiển qua một máy tính chuyển giao của mạng, do đó, hắn có thể theo dõi và ghi chép việc vận chuyển thông tin của một người sử dụng hợp pháp khi truy cập vào mạng. Do đó, sau đó hắn ta có thể truy cập vào, nếu mật lệnh của người sử dụng đã bị hắn ta giải được.

Ở đây, chỉ có thể có một sự trợ giúp tốt, đó là tạo nên một mật lệnh của người sử dụng đã được khoá lệ thuộc theo thời gian, hoặc giả, tạo nên một giao thức an toàn giữa các máy tính của người sử dụng và các máy tính mangnj và các tham số này phải luôn luôn được thay đổi.

Quấy phá các dịch vụ mạng (netservice interference):

Qua câu chuyện thần thoại Hy Lạp về con ngựa thành Tơ-roa, chúng ta nhận biết rằng, thành Tơ-roa không bị kẻ giặc xâm chiếm là nhờ sự chế ngự các thành luỹ vững chắc của nó và đặc biệt nhờ người dân thành Tơ-roa mưu trí; vì vậy, trong đêm tối, chỉ bằng một con ngựa gỗ nghi binh, họ đã đánh cho bọn xâm lược một đòn tơi bời ngay trong thành.

Nói chính xác, chiến lược mà người dân thành Tơ-roa dùng để đánh bại kẻ xâm lược khi chúng đột nhập vào thành cũng giống như chiến lược chống bọn quấy phá hệ thống mạng máy tính. thật vậy, các khiếm khuyết tồn tại ở các dịch vụ trong mạng đã được dùng để điều hành một chương trình đơn lẻ trong máy tính. Ở trong mạng máy tính có rất nhiều kiểu khiếm khuyết như vậy tồn tại trong các dịch vụ sau đây:

- *Các chương trinh thư điện tử (e-mail programm):*

Một thông điệp của một gởi người lạ có thể chứa đựng tất cả, kể cả các ký tự trống; những điều này thường không biểu lộ nội dung text; mà đặc biệt, tạo thời cơ cho thiết bị đầu cuối để nạp các lệnh bằng một chưc năng của bàn phím.

Nếu điều đó xảy đến đối với người quản lý hệ thống (*super user)* của hệ điều hành Unix, do đó, tại lần khởi động kế tiếp của nút ấn chức năng, các lệnh được gởi đi với hiệu lực lệnh của super user. Nếu lệnh cuối cùng xoá dòng hồi âm và xoá sự che phủ của nút chức năng; do vậy, người quản lý hệ thống không nhận biết được về sự hướng dẫn do anh ta đã phân phát; thí dụ quyền truy cập xác định đã bị thau đổi; tức là sau đó, có kẻ đột nhập đóng vai trò một người khách có thể thao tác các tệp tin này mà không có gì ngăn cản được.

Một biện pháp đối phó với cái đó là, phải lọc một cách triệt để tất cả các ký tự trống ở trong các dữ liệu của e-mail

- *Các dịch vụ trang WEB:*

Các trình đọc lượt browser của hệ thống siêu text của trang WEB hầu hết sử dụng các chương trình trợ giúp khác nhau, để thực hiện các dịch vụ đặc biệt. Thí dụ cho cái đó là các chương trình đêmô âm thanh và hình ảnh cho các khuôn khổ tệp tin khác nhau. Nếu có một tệp tin tái bút *(postscript-file)* tồn tại ở trong mạng và người sử dụng muốn nhìn thấy nó, do đó, tệp tin được nạp lên mặng và được trao quyền cho một chương trình đặc biệt ghostview, chương trình sẽ mô tả bài text chứa đựng về cái đó và cả hình ảnh lên màn hình. Đáng tiếc, sự tái bút này không chỉ là một ngôn ngữ mô tả trang, mà còn là một ngôn ngữ lập trình; do đó, nó có thể chứa đựng những chương trình khả thi, và cho nên chúng được thực thi ngay lập tức bởi trình ghostview ở trong thư mục của người sử dụng mà không cần biết gì thêm về chúng. Tính chất này được người ta đưa vào cấu hình hệ thống nếu nó còn là mối lo ngại và nếu người quản lý hệ thống nắm vững các tài liệu khảo cứu...

Một khả năng xâm nhập khác cũng được nhóm Action-X-Technologie của hàng Microsoft đề cập. Theo nhóm này, các đối tượng được mạp vào mặng có chứa đựng mã và nhờ thế chúng có thể truy cập lên máy tính.

- *Các dịch vụ chuyển vận tệp tin FTP:*

Trên mỗi máy tính được kết nối vào mạng có tồn tại một dich vụ chuyển vận tệp tin fftp; dịch vụ này có thể được gọi từ bên ngoài. Cái đó có thể được chiếm giữ như là một điểm lèo vào. Khi đó tiến trình ftp trên một máy tính thực ra có ít quyền hạn. Tuy nhiên, nếu các quyền hạn đươc điều chỉnh sai hay nếu các quyền truy cập đối với các tệp tin của các người sử dụng khác bị thiết đặt sai; do đó, ke tấn công từ bên ngoài có thể sao chép các tệp tin chủ lực như tệp tin về mật lệnh đăng ký vào một máy tính, và vì thế, hắn ta có thể thực hiện những cuộc tấn công rộng hơn.

Từ lý do này, thực tế tiến trình ftp không có quyền hạn truy cập nào cả và những điều kiện hoạt động của nó phải bị giới hạn trên cây thư mục (với các tệp tin bị chúng chặt nhỏ một cách chính xác).

6.5.3. Việc đảm nhận điều khiển trên một máy tính:

Nếu có một kẻ đột nhập đã tìm được lần đầu tiên lối dẫn vào hệ thống máy tính; do đó, hắn ta có thể thử nghiêmh nhiều cách khác nhau để đạt được quyền

quản lý hệ thống mạng máy tính. Ngay cả ở các máy tính có hệ điều hành Unix cũng chưa thông thạo các quyền quản lý mạng; việc đảm nhận một chương trình hệ thống quan trọng thì cũng có ý nghĩa như việc đảm nhận điều khiển chung qua máy tính. Bấy giờ, nó có điều kiện như thế nào để đạt được trạng thái của một người sử dụng bình thường về các quyền quản lý mạng? Thuộc cái đó, có nhiều phương pháp khác nhau; các phương pháp này sử dụng những điểm chủ yếu sau đây:

Quyền truy cập hệ thống tệp tin:

Nếu các quyền hạn đọc/viết đối với các tệp tin hệ thống được đặt sai vị trí; do đó, một kẻ đột nhập có thể dễ dàng sử dụng các tệp tin nay.

Thí dụ về đường dẫn lệnh:

Nói chung, các lệnh ở trong Unix đều được che đậy; do đó, người ta chỉ có thể tìm kiếm ở các vị trí khác nhau ở trong thư mục các tệp tin khả thi với tên lệnh của nó sẽ được tìm thấy; đó chính là chương trình được thực thi. Dãy tuần tự các vị trí thì nạp vào trong chuỗi ký tự với tên đường dẫn của nó. Nếu người ta muốn cho lệnh cmd được thực hiện, do đó, người ta phải tìm kiếm theo các đường dẫn /bin/cmd, /usr/bin/cmd, usr/local/bin/cmd và ./cmd. Nếu mỗi người có các quyền truy cập trên một trong các thư mục ở cây thư mục, do đó, anh ta có thể điều chỉnh ấn bản lênh của anh ta chính thức ở đó, thí dụ ấn bản của các lệnh rlogin, si hay ls để hiển thị các tệp tin.

Bấy giờ, nếu người ta thực hiện một chương trình như vậy, do đó, nó có thể cung cấp các kết quả mong muốn và có thể nạp bổ sung thêm (với các lệnh rlogin và su) các mật lệnh ở tại tệp tin của kẻ đột nhập (thí dụ chiến lược giữ thành Tơ-roa nói trên). Một chương trình đăng ký vào mạng login có thể nghĩ tới, đó là việc nhận biết mật lệnh chuẩn của kẻ đột nhập mà không cần biết hắn ta có được mời hay không.

Quyền truy cập các chương trình hệ thống:

Có rất nhiều chương trình dịch vụ về các công việc của hệ điều hành có cấc quyền hạn rộng rãi. Thí dụ, trình soạn thảo phải được đọc/viết trên tất cả các tệp tin của người sử dụng; một chương trình thư điện tử e-mail phải được phép viết các tệp tin thư điện tử cho tất cả các người sử dụng ở trong hộp thư của họ và một bộ kiểm tra trạng thái tiến trình có thể truy cập trên các bảng của nhân hệ điều hành. Nếu nó đòi hỏi sử dụng các hoạt động của chương trình hệ thống, thì điều đó tác động như là những công việc đã được dự định, do đó, kẻ tấn công co cum ở trong hệ thống.

Thí dụ về lỗi Emacs:

Emacs là một trình soạn thảo e-mail thông thường. Trong trình soạn thảo này, người ta có thể viết một e-mail vào trong một thư mục (với cửa sổ *movemail)*. Tuy nhiên, trình Editor này đã không kiểm tra xem, liệu thư mục đích đã thuộc người sử dụng hiện hành chưa (?), và do đó, đã tạo điều kiện viết đè các tệp tin hệ thống khác. Điều này đã là một trong các kẻ hở để kẻ tấn công người Đức sử dụng và xâm nhập vào các nhiệm vụ bảo vệ an ninh của các máy tính quân sự ở Mỹ và nghiên cứu các bí mật quân sự (1988).

Về vấn đề này, những ấn bản mới của các chương trình hệ thống được tạo lập vẫn chưa giải quyết được hoàn chính. Các gói thông tin dạng COPS của CERT đã chỉ cho hệ điều hành Unix một nhiệm vụ an toàn về quyền truy cập và những điểm yếu khác như các mật lênh quá đơn giản, các mật lệnh không có hiệu lực hay còn chứa đựng các lỗi. Những thông tin tiếp theo được tìm thấy ở SAT 1997 về các công cụ đối phó Santan, cũng như ở DFN 1997 về các miếng vá an toàn (*safe patches).*

Tạo Virus

Một kiểu tấn công quấy phá hệ thống khác cũng được nói tới, nó không thực hiện bằng tay mà bằng chương trình để tạo ra virus. Nếu có một khả năng nào đó mà nó chứa đựng các bản sao của một chương trình gọi là "*bẻ khoá ăn trộm"*, do đó, chương trình này có thể lan truyền "*một căn bệnh"* từ máy tính này tới máy tính khác. Ngoài ra, nó còn sử dụng một số lượng lớn các chương trình khác nhau (thí dụ các chương trình hệ thống, các chương trình mạng...) đóng vai trò kẻ chuyển giao bệnh, vì vậy, kiểu các chương trình này cũng được gọi là virus.

Có nhiều loại virus khác nhay. Các Virus lây lan trong các hệ thống MS-DOS (khoảng 80%) là loại virus di cư trong các chương trình khởi động (*bootstrap),* chúng hoạt động được khi hệ thống khởi động, gọi là virus khởi động (*bootstrap-virus)*. Về nguyên tắc, loại này được truyền qua đĩa mềm, thường nhiễm vào các sector khởi động (*sector* số 0 của đĩa mềm) và tiếp tục truyền lan sang ổ đĩa cứng. Từ lý do này, các cấu hình *bootstrap* mới (BIOS-EPROM) tạo cho người chủ sơe hữu phương tiện phương pháp ngăn hãm việc mô tả sector số 0 của ổ đĩa cứng.

Một kiểu virus nữa tồn tại trong máy tính, nó không truyền bằng con đường khởi động đĩa mềm (thí dụ các máy tính với hệ điều hành Unix), mà nó lây lan qua các tệp tin thực thi của người sử dung. Khi đó, con virus sẽ hợp gộp một bộ phận mã khi kết thúc chương trình, bộ phận này chứa đựng bản sao virus. Hình 6.21 chỉ ra nguyên tắc tạo virus theo kiểu này.

Hình 6.21 trang 255.

Virus chốt lại trong dãy tuần tự khởi động, để khi khởi động, nó được thực hiện trước nhất, do đó, nó dẫn tới quá trình làm việc mập mờ, đồng thời nó cúng còn gây ảnh hưởng tới việc điều khiển chương trình góc. Trong quá trình ấy,

người sử dụng không nhận biết cái gì cả về các công việc khác nhau, chính virus đã gây nên cái đó cho hoạt động của chương trình. Nếu virus lây lan lên hệ thống tệp tin chậm chạp và đều đặn, do đó, người ta có thể nói rằng, không một cái gì có thể nhận biết điều đó.

Loại virus thứ ba thường tồn tại trong các thủ tục; các thủ tục này được pha trộn thêm một đối tượng, mà người sử dụng không hề nhận biết điều đó. Một thí dụ tượng trưng cho cái đó là các tệp tin text; các tệp tin này được nạp thêm các chức năng phù hợp với định nghĩa người sử dụng, thí dụ các chức năng được viết bằng ngôn ngữ bậc cao (không phụ thuộc vào phần cứng), được gọi là ngôn ngữ Macro đối với trình soạn thảo Word của Microsoft. Nếu người ta nhận được một tệp tin viết trên Word từ một máy tính bên ngoài, do đó, nó có thể chứa đựng một virus được viết bằng ngôn ngữ Macro, gọi là *Macrovirus*; trong nội dung bài text, người ta không nhìn thấy virus này. Khác với loại vius nói trên, loại virus này không được khởi động trực tiếp với chương trình chủ; đặc biệt, nó chỉ xuất hiện khi nạp các dữ liệu. Nhờ sự độc lập của nó với phạm vi hệ điều hành và phạm vi máy tính, mà Macrvirus là loại lây lan và di truyền mạnh mẽ. Những tệp tin text chưa quen biết, đầu tiên, phải được thu gom lại bằng trình quét virus *(virusscan)* lên các chức năng quen biết và tiếp đến chỉ việc dời chúng ra khỏi tệp tin.

Phát hiện virus:

Điều kiện để phát hiện virus là phải tạo ra một tổng ngang, gọi là tổng kiểm tra của tất cả các tệp tin có thể thực thi và phải nạp chúng vào các tệp tin đặc biệt. Nếu khuôn dạng của tệp tin tổng kiểm tra đều đặn các tổng ngang có thể dẫn tới việc phát hiện virus.

Để phát hiện virus còn có một phương pháp khác, đó là phương pháp tạo lập một tệp tin trắc nghiệm được định nghĩa chính xác và phải kiểm tra thường xuyên. Nếu chúng bị virus làm thay đổi, do đó, người ta không thể khẳng định có tồn tại virus không (!), khi đó, người ta có thể tách chia mã và có thể tìm kiếm virus trong hệ thống máy tính và tẩy xoá chúng.

Diệt virus:

Phương cách có khả năng diệt được virus là phải đảm bảo sự yên tĩnh (*don't panic)*. Nếu chúng ta có một chiếc máy tính PC, do đó, chiến lược diệt virus được nêu ra như sau: Trước hết chúng ta phải có một đĩa mềm hệ thống kèm theo một chương trình chông virus hữu hiệu, mà chương trình này nhận biết sự nghi vấn có virus; thì người ta đạt được việc thanh trừng virus trong máy, và diệt sạch virus khỏi bộ nhớ chính. Sau đó bật điện và khởi động hệ thống, cho chạy lại chương trình diệt virus, khi đó, bộ nhớ ổ đĩa cứng mới được sạch sẽ hoàn toàn.

Nếu chương trình tìm kiếm virus vẫn không tìm thấy virus hoặc nếu chúng ta hoàn toàn không có một chương trình diệt virus thích ứng, khi đó, chúng ta phải tiến hành phương cách khác. Đầu tiên, tất cả các chương trình khả thi của người sử

dụng không cần thiết được xoá sạch. Sau đó, trong chuỗi logic các biện pháp hữu hiệu, chúng ta phải thay thế tất cả các chương trình hệ thống. Thuộc cái đó, chúng ta khởi động hệ thống với bản sao hệ điều hành sạch sẽ và tạo lập mới tất cả các chương trình hệ thống quan trọng. Tiếp đến chúng ta sử dụng các chương trình gốc của đĩa mềm hệ thống (cũng như các phương tiện tiện dụng gốc cũng được copy lại), cho tới khi tất cả được trở lại như trước đây. Cuối cùng, tất cả các chương trình người sử dụng được biên dịch phục hồi lại.

Với một hệ thống máy tính lớn, thì thật tiếc, người ta không thể tiến hành theo cách vừa nêu, vì việc điều hành hệ thống vẫn còn tiếp diễn. Ở đây, với một chương trình scaner trợ giúp, chương trình này sẽ kiểm tra virus một cách hệ thống tất cả các chương trình có nghi vấn và đồng thời diệt sạch virus. Tuy nhiên, kết quả của phương pháp vừa nêu cũng chưa được hoàn thiện: Nếu chương trình diệt virus chạy chậm hơn quá trình diệt virus lây lan, thì không kết quả nào được dẫn tới cả. Khi đó, người ta có thể viết một cách có phân tích một chương trình chống virus đặc biệt để phòng thủ một cách sinh động.

Phòng ngừa virus:

Trước hết, chúng ta khảo cứu các tệp tin mã nguồn (quellcode files) với các hoạt động virus; và tiếp đến, chúng ta biên dịch (compile) các chương trình hệ thống là các chương trình sử dụng một cách mới mẽ. Nếu chúng ta nói rằng, không còn virus ở trong hệ thống nữa (!), thì điều đó chẳng bao giờ có được (!).

Theo K.Thompson, ông là một trong những người cha đẻ của hệ điều hành Unix, cho rằng (1984), khi một con virus có thể tụ lại lâu dài, thì mã thực thi của trình biên dịch sẽ bị nhiễm bệnh. Mã này được phân biệt thành hai trường hợp: Trường hợp thứ nhất xuất hiện khi chương trình logic được biên dịch. Khi đó mã bị bệnh sẽ hợp lại và được người sử dụng thả vào một chương trình hướng đối tượng đến một cái tên xác định. Trường hợp thứ hai xuất hiện khi trình biên dịch từ mã nguồn một cách mới mẽ; do vậy, mã bị bệnh sẽ tụ họp lại trong mã của trình biên dịch.

Với chu trình này, virus tồn tại mà ngươi sử dụng không thể đọc thấy; nó có thể len lỏi ở trong hệ thống; khi biên dịch với text nguồn của các chương trình. Cho nên, mục đích là phải kiến tạo lại một bản sao cho các tệp tin nhị phân.

Hệ thống một chương trình tái phục hồi nói ở trên thực hiện hoàn toàn không đơn giản, nhưng có thể làm được. Người ta thử nghiệm chỉ một lần bằng cách viết một chương trình nhỏ khoảng vài dòng, mà chương trình này biểu lộ text nguồn khi chạy (không phải mở tệp tin nguồn)

Dưới đây, người ta sẽ nói tới những cơ chế an toàn ở các hệ điều hành Unix và Windows.

Sự nhận dạng ở trong Unix:

Việc điều khiển đăng ký vào mạng ở hệ điều hành Unix được dẫn ra với tên người sử dụng và một mật lệnh của cá nhân ngươi sử dụng. Hình 6.22 chỉ ra một quá trình logic của người sử dụng.

hình 6.22 trang 258.

Cũng như ở mục 4.3.1 đã được mô tả, ở đây, mỗi người sử dụng được thu xếp một nhận dạng người sử dụng uid và một nhận dạng nhóm công tác gid; tại mỗi tệp tin, các quyền truy cập được định nhhĩa cho các sự nhận dạng này. Ngoài ra, các sự nhận dạng này cũng tồn tại một mật lệnh để khoá vào một tên của người sử dung (chúng được tách biệt bởi ":") trong tệp tin/etc/passwd; sự điền vào của chúng được dẫn ra bằng một thí dụ như sau:

Brause:ntkgblic☛kh3j:105:12:&Brause:/user/user2/NIPS:/bin/csh

Dòng lệnh trên được giải thích như sau:

. brause là tên người sử dụng
. ntkgblic☛kh3j là mật lệnh;
. 105 là nhận dạng người sử dụng uid
. 12 là nhận dạng nhóm công tác gid;
. &Brause là tên người sử dụng;
. :/user/user2/NIPS là thư mục hiện hành của người sử dụng;
. /bin/csh là tên tiến trình khởi động của shell.

Tên nhóm được định nghĩa bởi tệp tin /etc/group, thí dụ:

Staff:🗐:12:boris,peter,brause

Dòng lệnh trên được giải thích:

. staff là tên nhóm;
. 🗐 là mật lệnh (để khoá) của nhóm;
. 12 là nhận dạng của nhóm;

. boris, peter, brause là tên người sử dụng được tạo lập cùng với nhóm.

Ở đây chỉ cần lưu ý, một người sử dụng có thể tồn tại đồng thời trong các nhóm công tác khác nhau.

Phân quyền ở Unix:

Mỗi tiến trình ở Unix có hai quyền hạn truy cập khác nhau: quyền truy cập của người tạo lập, được gọi là quyền có thực ruid và rgid; và do đó, các quyền này của người sử dụng thì cũng giống như các quyền hiệu nghiệm euid và egid. Nếu một chương trình được thực hiện, mà nó được tạo ra bởi một người sử dụng hay bởi một nhóm công tác; do đo, các chỉ dẫn về tệp tin không cần phải lưu y,nó được thay thế bằng các phép gán euid:=ruid cũng như egid:=rgid. Nếu một chương trình được tạo ra bởi người quản lý(super user), do đó, thuộc chương trình, còn có các quyền truy cập của người sử dụng.

Những chương trình như các chương trình e-mails phải được viết lên các tệp tin của người sử dụng khác nhau. Để trao cho chương trình các quyền hạn mạnh mẽ hơn về thư mục root; bây giờ, người ta có thể nhận thấy tại tệp tin rằng, các quyền hạn truy cập của chúng(uid và gid) được chuyển tới trên tiến trình (tức trên euid và egid) và không cần phải lưu ý tới các lệnh ruid và rgid (tức thay thế các thuộc tính set user id và set group id).

Nhận dạng trong mạng ở Unix:

Ở các mạng cục bộ với bộ điều hành Unix, người ta có thể điền vào một máy tính với lệnh rlogin (romote login) hay rsh (remote shell). Để phòng ngừa, người ta phải luôn luôn thay đổi mới mật lệnh; do đó, trên mỗi máy tính chỉ có một tệp tin etc/hosts.equiv tồn tại; trong đó, tất cả các máy tính đều nhận biết điều đó; vì vậy, tạo nên được sự tin cậy. Nếu một người sử dụng của một hệ thống láng giềng đăng kí vào; điều đó chỉ rõ, mật lệnh sẵn sàng được đọc chọn, và người sử dụng được đặt vào với các quyền truy cập của nó mà không cần xem xét gì cả. Đó là một chỗ yếu được kẻ đột nhập ưa chuộng; ngờ điểm yếu này kẻ đột nhập đảm nhiệm quản lý các máy tính của mạng LAN với hiệu ứng domino.

Nhận dạng người sử dụng ở Windows NT:

Đối với chức năng của máy chủ, ở Windows NT có các bản phác thảo khác nhau được thực hiện. Về danh sách điều khiển truy cập ACL các hệ thống tệp tin cục bộ, sự điều khiển các nhóm người sử dụng cục bộ hay toàn cục đan chéo nhau được tạo lập. Do đó, nói chung có ba điều kiện khác nhau đối với một người sử dụng phải thông báo vào hệ thống máy tính:

- Trình báo cục bộ, tức là buộc người sử dụng phải trình báo tên người sử dụng cục bộ của mình ở trên máy tính của anh ta.
- Trình báo mạng là công việc nhằm đảm bảo cho người sử dụng có thể truy cập lên các tệp tin của dich vụ tệp tin trên mạng.
- Trình bóa trong vùng tức là trình báo với máy vùng, nơi người sử dụng tồn tại nhằm đảm bảo việc điều chỉnh truy cập lên máy tính và các dịch vụ của máy chủ địa phương (domain server) cho người sử dụng.

Tuy nhiên, mỗi máy trạm (*workstation*)của hệ điều hành Windows NT có thể tác dụng như một server nhỏ, con máy vùng có chức năng như một máy chủ đặt biệt. Nó là một máy chủ kiểu *windows NTserver*(NTS) và tác dụng như một bộ điều khiển vùng DC(*domain controler*). Để nâng cao hiệu suất và khả năng dịch vụ, ở một vùng lớn hơn; khi đó không chỉ có một mà có nhiêu bộ DCs. Nhưng trong số đó, chỉ có một DC chính, gọi là bộ điều khiển vùng quan trọng PDC(*primary domain controller*), còn có các bộ DCs khác đóng vai trò bộ DC dự phòng (*backup domain controller*: BDC), chúng hoạt động được nhờ bộ DC chính. Do đó, sự nhận dạng người sử dụng trên một bộ BDC là có thể, nhưng không được

điều chỉnh trực tiếp các dữ liệu đã đăng ký. Điều này chỉ có thể đạt được một cách gián tiếp từ các người quản lý hệ thống, họ điều hành các dữ liệu của máy vùng(DS). Do đó, mô hình về các bộ BDC và PDC là đồng nhất.

Sự trao đổi giữa người sử dụng và các dữ liệu của các máy vùng DSs khác nhau có thể được làm giảm nhẹ nhừ thiết bị quan hệ tin cẩn (*trust relationship device)* giữa các máy vùng; do đó, người sử dụng nhận được một cách tự động các quyền truy cập xác định từ một máy vùng này tới một máy vùng khác. Đó là những quyền được khẳng định một cách rõ ràng bởi sự thu xếp các quan hệ tin cẩn. Cấu trúc này đã giảm nhẹ sự hợp tác giữa các công ty; do đó, không cần phải dẫn tới một quyền ưu tiên lớn hơn.

6.5.4. Cấu hình bức tường lửa(*fire wall configuration)*:

Một bản phác thảo quan trọng để ngăn chặn sự lan tràn của ngọn lửa trên nhiều nóc nhà là phải làm cách ly các ngôi nhà hay phân đoạn ngôi nhà nhờ những bức tường chống cháy và những cửa chống cháy. Bản phác thảo này là bảo bối để đạt được mọi mong muốn, nó giới hạn nhữnh lo âu về các vụ bẻ khóa trộm hệ thống ở một lĩnh vực nào đó trong mạng máy tính. *Bức tường chống cháy (fire wall)* ở trong trường hợp này là bao gồm *một máy tính trung chuan đặc biệt,* gọi là thiết bị dẫn lộ trình (*router*); nó hoạt động như là một khâu nối giữa một mạng cục bộ với một thế giới bên ngoài (đầy thù địch!), thí dụ với mạng internet. Với thiết bị fire-wall-router này, người ta đã giải quyết được nhiều nhiệm vụ; đó là nghiên cứu tất cả các gói dữ liệu (screening), đẽo gọt và tiểu trừ các gói thông tin với các địa chỉ internet hay địa chỉ bưu cục không mong muốn. Điều đó xảy ra rất nhanh, vì phải tránh được sự giao lưu thông tin cần thiết; do đó, việc nghiên cứu và gói thông tin được thực hiện như là một sự kết hợp giữa những biện pháp phần cứng và những biên pháp phần mềm.

Tuy nhiên, chúng ta vẫn còn tồn tại một vấn đề: Thiết bị fire -wall-router phải nhận biết các vấn đề được xuất phát từ dâu tới và những địa chỉ bưu cục của dịch vụ nào là có nghi vấn? Hầu hết, đó là những vấn đề phụ thuộc ngữ cảnh và thuộc người sử dụng.

Từ lí do vưa nêu, thuộc thiết bị bức tường lửa này, người ta còn đặt thêm *một máy tính thứ hai làm nhiệm vụ tiếp đón và truyền đi (relay host).*Máy này tồn tại ngoài mạng LAN, nó dược kết nối với mạng internet qua một thiết bị dẫn lộ trình khác ở bên ngoài *(external router).* Hình 26.3 chỉ ra cấu hình bức tường lửa kiểu như thế. Bây giờ, tất cả các chương trình quan trọng (*application relay software)* và các dịch vụ mạng *(netservice)* đều tồn tại trên thiết bị relay - host. Do đó, việc thực hiện của chúng được gói gọn theo những định hướng xác định; những định hướng này luôn luôn được kiểm tra theo các danh sách điều khiển truy cập lối vào (ACL) và tùy thuộc từng chương trình. Ở các danh sách nay, người ta không chỉ miêu tả người sử dụng mà còn miêu tả các quyền truy cập của họ; đặc biệt, người ta chỉ cho phép:ai và những chức năng nào của chương trình được quyền sử

dụng!Các chương trình cảnh giới *(wrapper)* có thể kiểm tra các kết nối trao đổi thông tin và khống chế các gói dữ liệu trên thiết bị relay-host.

(hình 6.23./261)

Các lãnh thổ an toàn được các thiết bị relay-host hoàn thiện với các chương trình và các kết nối internet được tuyển chọn đảm bảo của nó. Bấy giờ, người ta có thể truyền thông tin một cách dễ dàng qua thiết bị fire-wall-router.Do đó, người ta cho phép chỉ một mình thiết bị routernayf cho đi qua các gói dữ liệu giữa các máy tính trong mạng LAN cả thiết bị relay-host; tất cả các kết nối trao đổi thông tin khác được tiến hành một cách cưỡng bức qua thiết bị relay-host. Trên mạng internet, theo kiểu cấu hình này có *hai rào chắn an toàn* đối với kẻ đột nhập: đầu tiên là rào chắn external-router và sau đó là rào chắn relay-host.

Nếu chúng ta giảm bớt thiết bị external-router và kết hợp chức năng của nó với thiết bị relay-host; do đó, chúng ta sẽ tiết kiệm được khá tiền bạc, tuy rào chắn an toàn có giảm đôi chút. Ngoài ra, chúng ta còn có một khả năng nữa, tiết kiệm hơn nhiều. Đó là sự kết hợp các chức năng của ba thiết bị thành chức năng bao quát của một thiết bị duy nhất, đó là cấu hình thiết bị kiểm tra và khống chế quay về đối ngẫu (dual-homed-host) như ở hình 6.24 ở dưới đây.

(hình 6.24 /262)

Ở đây, với việc chấp nhận một cơ chế an toàn như vừa nêu, sẽ xảy ra hiện tượng: Nếu kẻ tấn công đạt được việc bẻ gãy khóa hệ thống ứng phó đối ngẫu này, do đó, nó cũng bẻ gãy được bức tường chữa cháy. Từ lý do này có các hệ thống dùng các cơ chế thanh lọc để điều khiển người sử dụng. Một trong các cơ chế quen thuộc nhất là hệ thống canh giữ cerberus.

6.5.5. Nhận dạng cerberus

Hệ thống an toàn kiểu cerberus là một phần của dự án nhằm phát triển việc quản lý hệ thống cho một mạng với khoảng 1000 máy tính trạm ở viện nghiên cứu MIT của Mỹ. Ở một mạng như thế thì xuất hiện một câu hỏi, liệu tiến trình dò tìm muốn có một khả năng dịch vụ nào đó nó có đạt được sự ưu tiên cho nó không? Khi đó, vấn đề định dạng không chỉ giơi hạn tại thời điểm trình báo của người sử dụng, mà còn dẫn tới các khả năng dịch vụ ở trong mạng. Vì nó cũng là vấn đề nan giải, để gởi một mật lệnh bỏ ngỏ chưa mã hóa trên nhiều máy tính, do đó, ở trong dự án này, có ba giao thức khác nhau được chuyên môn hóa để giải quyết vấn đề an toàn.

Ý tưởng trung tâm của dự án này là ở chỗ, phải chốt lại việc dò tìm và pahỉ kết nối nội dung của người sử dụng vơi nhận dạng mà chìa khóa (kí hiệu s_1) đã được chứa đựng. Khi khóa, một hàm khóa (kí hiệu f) còn gọi là hàm bẩy (trap function) được sử dụng, hàm này không dễ dàng bị biến đổi. Nội dung đích xác của thông

tin (kí hiệu y) và hàm của khóa f thì không phải được xác định một cách dễ dàng qua sự nhận dạng về f và y. Nói chung, khi đó có một hàm giải mã g(.) được sử dụng; hàm này có thể tái phục hồi các thông tin cũ vơi sự trợ giúp của một chìa khóa thứ hai (kí hiệu s_2).Những quan hệ vừa nói được biểu diễn trong biểu thức sau:

$$\text{Thông tin} = g(y, s_2) = g(f(\text{Thông tin}, s_1)s_2)$$

Hay:

$$\text{Thông tin} = g_{s2}(f_{s1}(\text{Thông tin}))$$

Có những hệ thống, mà tại đó, có thể dẫn tới f = y và $s_1 = s_2$; nhưng điều đó thì cũng không nhất thiết. Sau đây giới thiệu 3 giao thức để giải quyết vấn đề này.

① Giao thức trình báo (single sign on protocol)

Khi người sử dụng trình báo với kiểu nhận dạng cerberus, mọi việc sẽ dẫn ra một cách quen thuộc: người sử dụng điền mật lệnh của anh ta, khi đó, việc trinh báo cáo này được một tiến trình kiểm tra và thu nhận sự nhận dạng của server. Thêm vào đó, anh ta nhận được hai thứ: thứ nhất là các thông tin xác thực C, thứ hai là một vé (ticket) đặc biệt T_G, gọi là vé vào cổng.

Các thông tin xác thực C được khóa với mật lệnh P của người sử dụng và có thể được anh ta mở khóa một cách dễ dàng. Điều đó chứa đựng chìa khóa $S_p(S_B,$ TGS, $t_L,$.....) (6.2)

Chìa khóa này chỉ dẫn tới một sự giới hạn cần thiết:Nó chỉ hoạt động với một thiết bị phân bổ chìa khóa trung tâm, thiết bị này gọi là một server phân bổ vé TGS (ticket granting server), và tồn tại trong khoảng thời gian t_L.

Khi tuổi thọ của một chìa khóa đàm luận tồn tại một cách tiện dụng khoảng 8 giờ đồng hồ, người sử dụng có thể đón nhận một chìa khóa ở TGS với sự trợ giúp của vé vào cổng T_G cho các truy cập biến động (sao tệp tin,dich vụ in ấn, ...). Với vé vào cổng T_G, đối với serverTGS, không chỉ người sử dụng chứa đựng chìa khóaS_B,mà chứa đựng cả tuổi thọ của vé T_G, tức là:

$$T_G = f_{ST}(S_B, \text{người sử dụng}, t_L, ...) \qquad (6.3)$$

Vé vào cổng được khóa lại bởi mật lệnh S_T của server TGS, do vậy, không có một trường hợp nào có thể dẫn tới sai sót được. Do đó, giao thức trình báo được kết thúc.

② Giao thức phân bổ chìa khóa (key distribution protocol):

Bây giờ, tiến trình người sử dụng muốn thực hiện một biến cố, do đó, anh ta không thể gởi ngay sự dò tìm bị khóa của anh ta tới server: Khi đó mọi việc hoàn toàn chưa biết, với chìa khóa nào làm cho thông tin đi tới bị khóa.Vì vậy, đầu tiên nó hỏi tới một thẩm cấp trung tâm chẳng hạn server TGS về một ticket (vào cổng làm việc)T_T(transaction ticket). Tương tự, trước đó tại server nhận dạng, người sử dụng gởi đi một sự dò tìm (về client, server, tuổi thọ, số biến ngẫu nhiên, ...)trong

một bài text rõ ràng tới server phân bố vé vào cổng TGS và anh ta nhận trở lại một giấy chứng nhận đã được mã hóa S_B cũng như một vé vào cổng T_T. Trong giấy chứng nhận, tất cả những điều cần thiết được chứa đựng cho một biến cố mong muốn: chìa khóa S_{CS}đối với biến cố giữa client và server, một số biến cố cũng như các sự chỉ dẫn thích hợp đều chứa đựng trong vé vào cổng T_T, mà vé này đã được mã hóa với chìa khóa server S_S.

③ Giao thức nhận dạng (authentication protocol):

Trong khoảng thời gian đã được chuyên môn hóa nhờ sự tồn tại ngắn ngủi của biến cố (khoảng 5 phút), bấy giờ, client có thể dẫn ra một biến cố với vé vào cổng T_T và mã hóa các dữ liệu của chúng với chìa khóa client và server: Máy chủ chấp nhận với vé vào cổng này, một biến cố (chẳng hạn sự nhận dạng)và ngoài ra với chìa khóa chứa đựng về cái đó, nó còn có thể mã hóa các dữ liệu tiếp theo.

Ba giao thức tóm lược ở trên đã bổ sung cho việc kết nối nhiều mạng LAN với nhiều domainserver (máy chủ khu vực) làm việc với nhau nhờ một trong bốn yếu tố sau đây:Nếu một client muốn sử dụng tài nguyên của server ở một mạng LAN khác, do đó, đầu tiên, nó gởi đi một sự dò tìm tới một server phân phối ticket vào mạng LAN của nó, và qua đó, để có một biến cố được bảo đảm bởi một server phân phối ticket của mạng LAN láng giềng. Nếu mọi chuyện xảy ra tốt đẹp, người sử dụng dò hỏi tiếp tới server về một biến cố tiếp theo cho nó, và ngay khi đó, máy client này sẽ có một ticket cho việc xâm nhập một server mong muốn. Bây giờ, nó có thể trực tiếp sử dụng các tài nguyên mong muốn tại một server với một ticket và một mật lệnh khóa đã nhận.

Hệ thống nhận dạng kiểu cerberus thì thực ra rất khó; có một vài khó khăn cần phải vượt qua:

- Dịch vụ chỉ hoạt động khi đồng hồ của tất cả các người sử dụng chạy đồng bộ với nhau, và do đó, các ngăn xếp thời gian được kiểm tra chặt chẽ. Điều này yêu cầu một hệ thống nhận dạng thời gian ở các khu vực mà không cân đồng hồ của đài phát thanh trung ương Mỹ để tránh được việc thao tác đồng hồ, vì điều này phải được tạo lập trên một dịch vụ tương tự.
- Một ngăn xếp thời gian được tạo lập cố định đối với chìa khóa biến cố và chìa khóa đàm luận nảy sinh các vấn đề, khi biến cố xảy ra lâu hơn hay người sử dụng làm việc lâu hơn.
- Chìa khóa biến cố và ticket biến cố có được đặt trong các tệp tin; các tệp tin này có thể được đọc trong phạm vi nhiều người sử dụng.

Do đó, nó là con đường hơi chậm để đi tới một hệ thống an toàn, đặc biệt khi nhiều hệ thống được kết nối với nhau với nhiều cơ chế khác nhau. Từ nguyên do này, các hoạt động của nhóm X/Open đã đạt được mục đích an toàn mạng với những phương hướng an toàn thông nhất, gọi là bản phương hướng an toàn cơ bản của nhóm X/Open(baseline security specification).

6.6. Các bài tập của chương 6

6.6.1. Các bài tập về trao đổi thông tin trong mạng

Bài tập 6.1. Các giao thức xếp lớp

Trong nhiêu giao thức xếp lớp, mỗi lớp định roc khuôn dạng thích hợp phần đề mục của nó khi bắt đầu một thông tin. Tại mỗi lớp, mỗi lần trao đổi, thông tin được khảo sát như một gói dữ liệu, mà các gói này với một khuôn dạng và một đề mục thông tin thích hợp được dẫn tới lớp kế cạnh lần lượt từ trên xuống dưới.Nó sẽ chắc chắn hiệu nghiệm hơn, nếu chỉ có một đề mục duy nhất ở đầu thông tin, mà đề mục này chứa đựng toàn bộ các thông tin điều khiển.Tại sao điều này không được người ta thực hiện?

Bài tập 6.2.

Bộ vi xử lý SPARC là loại 32 Bit có địa chỉ Byte cuối lớn; còn bộ vi xử lý 386 là loại 16 Bit có địa chỉ Byte cuối nhỏ. Gỉa sử bộ vi xử lý SPARC gởi một chữ số hai tới bộ vi xử lý 386. Hỏi bộ vi xử lý 386 chỉ ra giá trị nào?(Bỏ qua lớp chuyên vận, xem hình 6.14).

Bài tập 6.3. Về định vị trong Internet

Gỉa sử Internet có một địa chỉ logic:134.106.21.30

a). Nó là chữ số 32 Bit nào,nếu tất cả các chữ số ở trong địa chỉ logic Internet sử dụng một số lượng các số lượng các Bit bằng nhau? Nó phù hợp với số thập phân nào?

b). Bạn hãy tìm ra máy tính nào thì đạt được địa chỉ này? Ghi chú:Bạn hãy sử dụng một dịch vụ mạng!

c). Cho một số 32 Bit có kiểu địa chỉ Byte cuối lớn. Một số thập phân nào được dẫn tới,nếu kiểu địa chỉ Byte cuối nhỏ được sử dụng?

d). Những số cổng nào sử dụng các dịch vụ sau: ftp, telnet và talk trên các máy tính của cơ quan bạn?Ghi chú:Bạn hãy xem lại ở trong Unix và tệp tin có đường dẫn/etc/services.

Bài tập 6.4. Gọi thủ tục trở lại

Việc thực thi các gọi thủ tục đã được trình bày(ở mục 6.2.2.). Cái gì phải được dốc làm để thay thế các RPCs(ở Unix)bằng các thủ tục con, hay để thay thế các thủ tục con bằng các RPCs?

Bài tập 6.5. Về an toàn

Một phương tiện bất kỳ để đọc chọn các mật lệnh thì ở trong một chương trình, mà chương trình này thông báo lên màng hình "login:".Khi sử dụng một máy tính,người sử dụng phải trình báo *tên* và *mật lệnh* của anh ta.Nếu chương trình này ghi chép các dữ liệu và biểu thị một thông báo, chẳng hạn Password failure, please repeat you!hoặc tương tự và kết thúc. Nhưng điều đó cũng không làm người ta sực nhớ lại:ở đây,một mật lệnh đã bị lãng quên!Nếu người là bạn sử dụng hay nhà quản lý mạng, bạn có thể ngăn ngừa như thế nào cho hậu quả của một chương trình như vậy?

Bài tập 6.6.

Gỉa sử bạn nhận biết một virus xuất hiện ở trong máy tính của bạn. Dưới hoàn cảnh nào, người ta không thể đạt được việc làm sạch bộ nhớ ổ đĩa (cứng hoặc mềm)với một chương trình tìm kiếm virus?

Bài tập 6.7. Về bức tường lửa

Những nhược điểm của bức tường lửa là gì khi nghiên cứu các gói dữ liệu?

Bài tập 6.8. Về phương pháp cảnh giới cerberus

a). Sự khác nhau giữa hai khái niệm Authorize (ủy quyền) và Authentification (nhận dạng) là gì?

b). Trong phương pháp cerberus, hai chìa khóa được dùng thay vì một chìa để làm gì? Sự khác nhau hai chìa khóa đó là gì?