

MỘT MÔ HÌNH CÀI ĐẶT CÁC GIAO THỨC TRUYỀN THÔNG TRÊN MẠNG MÁY TÍNH

NGUYỄN THỨC HẢI

Trường Đại học Bách khoa Hà Nội

I - MỞ ĐẦU

Trong thực tế, không phải bao giờ cũng có đủ điều kiện để có được một mạng chỉ gồm các máy tính thuần nhất (về phần cứng, phần mềm). Trái lại, thường các tổ chức khi có nhu cầu lập mạng máy tính thì trong tay đã có các máy tính không thuần nhất do nhiều hãng sản xuất và sử dụng các hệ điều hành khác nhau. Điều đó dẫn đến sự cần thiết phải có những phần mềm truyền thông có thể chuyển mang dễ dàng (với rất ít thay đổi) từ một máy này sang một máy khác. Tuy nhiên hệ mềm truyền thông phải được thiết kế và cài đặt sao cho tính khả chuyển (portability) đó không làm giảm độ hiệu quả (performance) của nó. Mặt khác, phần mềm truyền thông phải có tính mở cao để có thể dễ dàng làm thích nghi hoặc bổ sung thêm các giao thức truyền thông (communication protocols) mới.

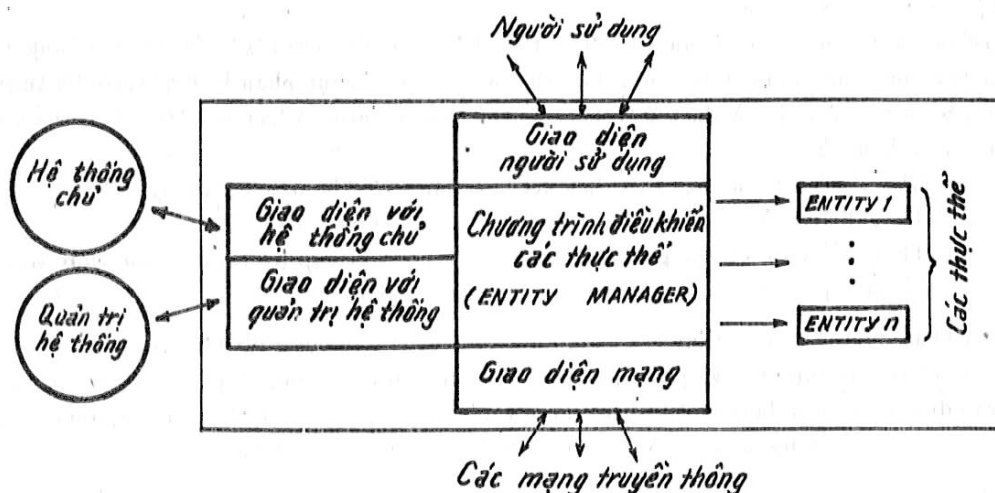
Bài này trình bày một mô hình cài đặt phần mềm truyền thông đáp ứng được các yêu cầu nói trên. Phần mềm truyền thông được xây dựng trên cơ sở các giao thức truyền thông chuẩn hóa của ISO để đảm bảo tính mở của hệ thống. Tuy nhiên, tính chuẩn của ISO không chỉ ra các phương pháp cài đặt cụ thể nên ta cần phải xây dựng các công cụ thích hợp để có thể đưa vào các cơ chế và các nguyên lý được mô tả trong kiến trúc nối kết các hệ thống mở của ISO.

II - CẤU TRÚC PHẦN MỀM TRUYỀN THÔNG

Hình 1 dưới đây cho ta sơ đồ cấu trúc để cài đặt phần mềm truyền thông, bao gồm 3 bộ phận chính:

- Các thực thể (Entity).
- Chương trình quản lý các thực thể (Entity manage).
- Các giao diện (Interface).

Mỗi thực thể là một chương trình độc lập tương ứng với các dịch vụ và giao thức truyền thông của một tầng (layer) trong mô hình tham chiếu bảy tầng của ISO. Chúng hầu như độc lập với môi trường hệ thống (chỉ phụ thuộc về ngôn ngữ lập trình sử dụng) và rất thuận tiện cho việc sửa chữa hoặc bổ sung các thực thể mới. Mọi tương tác giữa các thực thể đều thông qua chương trình quản lý các thực thể. Chương trình này được xem như nhân của hệ thống, nó cung cấp cho các thực thể tất cả các công cụ cần thiết để thực hiện sự tương tác đó. Bản thân Entity Manager cũng chỉ phụ thuộc máy chủ về phương diện ngôn ngữ lập trình sử dụng.



Hình 1: Sơ đồ cấu trúc phần mềm truyền thông

Mọi chương trình phụ thuộc vào máy chủ (phần cứng cũng như phần mềm) được nhóm lại trong các giao diện sau đây:

- Giao diện người sử dụng (user interface) cho phép người sử dụng truy nhập tới các thực thể đã được cài đặt. Các thực thể tương tác với người sử dụng theo cách như là với một thực thể khác.

- Giao diện mạng (network interface) bao gồm nhiều chương trình điều khiển các bộ ghép nối mạng của máy chủ (một máy chủ có thể đồng thời nối với nhiều mạng khác nhau). Chúng cho phép Entity Manager độc lập hoàn toàn với phương thức truy nhập tới các bộ ghép đó. Các thực thể do vậy tương tác với mạng cũng như với một thực thể bất kỳ, thông qua Entity Manager.

- Giao diện với máy chủ (host machine interface) tập hợp tất cả các chương trình phụ thuộc vào máy chủ trên đó phần mềm truyền thông được cài đặt. Các ngắt, các bộ đếm thời gian (time-out) từ hệ thống chủ cũng như các lời gọi hệ thống (system calls) sẽ được xử lý ở đây.

- Giao diện với người quản trị hệ thống (administrator interface) giúp cho người quản trị hệ thống có thể kiểm soát môi trường truyền thông bằng cách sử dụng các công cụ mà Entity Manager cung cấp. Việc kiểm soát đó có thể là:

- + Tham số hóa và khởi động môi trường truyền thông.
- + Biến đổi các tài nguyên của môi trường, thay thế các thực thể hỏng hoặc thêm các thực thể mới ...
- + Thống kê các hoạt động sử dụng tài nguyên trong môi trường truyền thông.
- + Truy nhập tới hệ quản trị mạng.

Với cấu trúc phân cấp sự phụ thuộc máy chủ như trên, phần mềm truyền thông được cài đặt rất dễ chuyển mang từ một máy chủ này sang máy chủ khác mà không cần sửa chữa nhiều các chương trình (chủ yếu chỉ sửa các giao diện). Mặt khác cấu trúc đơn thể của các thực thể làm cho hệ có tính mở cao như yêu cầu đã đặt ra. Sau đây chúng ta sẽ xem xét chi tiết hơn cấu trúc của các thực thể và đặc biệt của Entity Manager.

1. Các thực thể truyền thông

Để dễ chuyển mang, mỗi thực thể được thiết kế một cách có cấu trúc. Ngoài nội dung của giao thức tương ứng phải cài đặt, mỗi thực thể cần có các thủ tục phân biệt để trao đổi tương tác với các thực thể khác và thế giới bên ngoài thông qua Entity Manager. Mỗi thực thể phải được tham số hóa để:

- + Làm thích nghi số lượng các liên kết hiện hành với khả năng của máy chủ.
- + Thay đổi độ dài của các đơn vị dữ liệu của giao thức (PDU).
- + Thay đổi cấu hình của giao thức, ví dụ lựa chọn một lớp (class) của giao thức trong số các lớp khác nhau (giao thức Transport chẳng hạn)...

Nói cách khác, bản thân mỗi thực thể cũng được cài đặt theo một sơ đồ cấu trúc đơn thể, bao gồm phần nhân của thực thể và phần giao diện với thế giới bên ngoài. Mỗi thực thể sẽ có một điểm vào duy nhất dưới dạng một thủ tục được gọi bởi Entity Manager khi có tương tác:

Entity-name (SAP-id, inter-code, message, inter-param)

trong đó.

entity-name là tên của thủ tục, dùng để gọi khi muốn kích hoạt thực thể. Có thể dùng luôn tên của các tầng ISO như: network, transport, session,...

SAP-id là tên (identifier) của điểm truy nhập dịch vụ (SAP) hoặc của kênh (channel) mà trên đó tương tác được gửi đến.

inter-code là mã của tương tác (interaction code).

message là vùng chứa thông tin gửi cho thực thể.

inter-param là tham số của tương tác.

Từ điểm vào này sau đó điều khiển sẽ được chuyển đến cho các đơn thể chính của thực thể tùy theo mã của tương tác gửi đến.

2. Entity Manager

Là nhân của hệ thống, Entity Manager bao gồm các chương trình khởi động và quản lý các thực thể truyền thông. Nhiệm vụ đó được thực hiện thông qua một bộ của công cụ (tools) mà nó cung cấp cho các thực thể. Mỗi công cụ đó thực chất là một chương trình (dạng hàm) gọi là hàm nguyên thủy (primitive) mà các thực thể có thể gọi sử dụng khi được chuyển điều khiển. Để phản ánh kết quả thực hiện của một hàm nguyên thủy, ta dùng một mã phản hồi (return code) thích hợp cho mỗi hàm. Có hai nhóm hàm nguyên thủy chính:

- Các hàm nguyên thủy điều khiển tương tác.
- Các hàm nguyên thủy quản lý bộ nhớ.

Sau đây ta sẽ mô tả lần lượt từng hàm nguyên thủy đó.

2.1. Các hàm nguyên thủy điều khiển tương tác

1/ Attach - SAP (SAP-id)

trong đó SAP-id là tên của SAP được gắn với thực thể truyền thông.

Nhắc lại rằng theo mô hình tham chiếu OSI, ở mỗi tầng, mỗi thực thể được gắn với một điểm truy nhập dịch vụ (SAP = Service Access Point). Các thực thể ở mức cao hơn có thể truy nhập vào thực thể mức dưới qua các SAP đó. Khi cài đặt, các SAP có thể được thể hiện dưới dạng một cấu trúc dữ liệu nào đó, ví dụ các hàng đợi (queue). Các SAP có thể làm việc theo một trong ba phương thức: có liên kết, không liên kết hoặc hỗn hợp, trên mỗi SAP có thể định nghĩa nhiều kênh, mỗi kênh tương ứng với một liên kết logic. Hàm nguyên thủy trên cho phép một thực thể ở mức cao có thể liên lạc với một thực thể ở mức thấp hơn qua SAP được gắn đó. SAP này sẽ không được gắn cho một thực thể nào khác nữa.

Hệ thống sẽ quản lý một bảng chứa tên các thực thể và các SAP được gắn tương ứng.

2/ Detach - SAP (SAP-id)

Hàm nguyên thủy này dùng để giải phóng một SAP (tức là giải phóng cấu trúc dữ liệu biểu diễn SAP) để gắn cho thực thể khác.

3/ Allocate - channel (SAP-id, QOS)

trong đó QOS (Quality of Service) là tham số về chất lượng phục vụ của kênh (cả hai chiều), gồm có độ trễ (delay) và thông lượng (throughput) đòi hỏi. Riêng QOS về độ tin cậy giả thiết là hoàn hảo.

Hàm này sẽ gửi trả về tên của kênh (channel identifier) được cấp phát cho liên kết.

Sau khi đã được cấp phát, các yêu cầu hoặc chỉ thị thiết lập liên kết (connection request hoặc connection indication) của giao thức truyền thông có thể được gửi trên kênh.

4/ Free - channel (channel identifier)

Hàm này cho phép giải phóng kênh đã được cấp phát (bởi allocate - channel) để dùng cho liên kết khác.

5/ Asynchro - call (channel hoặc SAP-id, inter - code, message, inter - param)

trong đó:

- + inter - code là mã của tương tác.
- + inter - param là tham số của tương tác.
- + message là vùng thông tin trao đổi.

Hàm này cho phép một thực thể yêu cầu thiết lập một trao đổi dị bộ (asynchronous) với một thực thể khác hoặc với thế giới bên ngoài thông qua Entity Manager. Lưu ý rằng các thực thể không sử dụng các công cụ trao đổi đồng bộ (synchronous). Sự đồng bộ hóa sẽ ở mức tổng quát hơn và được điều khiển bởi Entity Manager. Tuy nhiên các thực thể có thể gửi các tương tác (tức các hàm nguyên thủy của giao thức chuẩn ISO) một cách dị bộ.

Mã tương tác biểu thị bởi mã của hàm nguyên thủy của giao thức chuẩn ISO, cho phép Entity Manager xác định độ ưu tiên của tương tác trên kênh cũng như để xử lý quan hệ loại trừ nhau giữa các tương tác.

Dữ liệu của tương tác được lưu trữ trong vùng có tên "message" còn các tham số khác của nó được để trong vùng "inter - param".

Entity Manager căn cứ vào độ ưu tiên của tương tác để quyết định trao điều khiển cho thực thể gọi hàm này và điều khiển sẽ được trả lại cho Entity Manager khi tương tác được xử lý xong.

"Channel - id" được dùng trong trường hợp dịch vụ có liên kết và "SAP-id" trong trường hợp không liên kết.

6/ Set - timer (channel hoặc SAP-id, entry, value)

Hàm này cho phép lập bộ đệm thời gian logic để điều khiển việc xử lý tương tác bằng cơ chế thời gian hạn định (time - out). "Entry" là điểm vào (tên thủ tục) chịu trách nhiệm xử lý hiện tượng time - out trong thực thể. "Value" chỉ giá trị của time - out theo đơn vị thời gian qui định trước.

7/ Disable - timer (channel hoặc SAP-id, entry)

Cho phép hủy bỏ việc sử dụng cơ chế time - out, bộ đệm thời gian logic được giải phóng.

2.2. Các hàm nguyên thủy quản lý bộ nhớ

Các thực thể tương tác với nhau bằng cách trao đổi các kênh và dữ liệu kèm theo (xem hàm asynchro - call). Do đó Entity Manager phải cho phép bộ nhớ được phân chia giữa các thực thể để các thông báo (chứa dữ liệu) được lưu trữ trong bộ nhớ. Cũng cần có một cơ chế bảo vệ bộ nhớ chịu trách nhiệm kiểm soát truy nhập tới bộ nhớ. Mặt khác, các thông báo đi qua nhiều tầng, từ người sử dụng tới tầng vật lý mạng (hoặc ngược lại). Ở mỗi tầng, thông báo có thể được biến đổi trước khi được gửi tới tầng kế tiếp. Thường sự biến đổi đó gồm có việc thêm vào hoặc cắt bỏ phần đầu (header) từ thông báo, cũng như việc phân đoạn và việc hợp nhất cho các thông báo.

Sẽ kém hiệu quả hơn nếu mỗi lần truyền từ một thực thể này tới một thực thể khác lại phải sao chép (copy) thông báo. Để tránh việc sao chép nhiều lần, cần có các công cụ xử lý thông báo và một tổ chức bộ nhớ thích hợp. Bộ nhớ cần phải được quản lý năng động. Các thông báo được lưu trữ trong các bộ đệm được liên kết sao cho việc xử lý chỉ giới hạn trong việc biến đổi các phần mô tả thông báo (message descriptor). Đồng thời một cơ chế "khóa" được xác định để quản lý việc sao chép: sao một thông báo cũng có nghĩa là lập một "khóa" cho nó. Việc hủy bỏ hoàn toàn một thông báo chỉ có thể đạt được khi không tồn tại một "khóa" nào nữa.

Các hàm nguyên thủy sau đây được định nghĩa:

1/ Allocate - message (data area, length)

Hàm này cho phép một thực thể tạo ra một thông báo từ một vùng dữ liệu (data area) được chỉ ra. Thông báo được tạo là một danh sách bộ đệm móc nối, độ dài của nó phụ thuộc vào giá trị của tham số length cũng như kích thước bộ đệm cung cấp bởi hệ thống. Từ đây nó sẽ thành vật sở hữu của thực thể cho đến khi thực thể chuyển nó cho một thực thể khác hoặc hủy nó đi.

Địa chỉ của phần mô tả thông báo sẽ được phản hồi bởi hàm này.

2/ Read - message (message, data area, position, length)

Cho phép một thực thể đọc (một phần hoặc toàn bộ) thông báo. Phần thông tin đọc được đặt vào "data area", bắt đầu từ "position" đến "position + length" (length là độ dài thông tin được đọc). Quyền sở hữu đối với thông báo cũng sẽ được kiểm tra (thông báo được tạo ra bởi thực thể này hay từ một thực thể khác chuyển giao).

3/ Concatenate - message (message 1, message 2)

Cho phép ghép nối thông báo "message 1" với thông báo "message 2", kết quả đặt tại "message 1". (Hàm này thường dùng, chẳng hạn, để ghép các phần đầu vào thân của một thông báo cần gửi đi).

4/ Length (message)

Hàm này phản hồi giá trị độ dài của thông báo, có thể dùng khi thực thể không biết độ dài của thông báo vì nó được một thực thể khác chuyển giao.

5/ Copy - message (message, position, length)

Cho phép sao chép (một phần hoặc toàn bộ) thông báo "message" và phản hồi địa chỉ phần mô tả của thông báo chứa "bản sao". Tham số "length" chỉ số lượng ký tự, tính từ "position" trở đi, của thông báo cần được sao chép. Lưu ý rằng việc sao chép ở đây chỉ hoàn toàn thực hiện một cách hình thức trên các phần mô tả của thông báo chứ không động gì đến vùng dữ liệu thực sự của thông báo. Hàm này có ích khi phân đoạn hoặc cắt xén một thông báo. Nó cũng có ích khi một thực thể muốn giữ một bản sao của thông báo mà nó đã gửi đi để chờ một xác nhận (acknowledgment).

6/ Delete - message (message, length)

Toàn bộ thông báo được hủy bỏ nếu giá trị của "length" chỉ ra là lớn hơn hoặc bằng độ dài của thông báo. Ngược lại, chỉ một phần của thông báo bị hủy bỏ. Thông báo được hủy bỏ thực sự nếu không tồn tại một bản sao nào ở một thực thể nào đó nữa.

Hàm này có thể dùng trong trường hợp một thực thể nhận một thông báo từ một thực thể ở mức thấp hơn, sau khi xử lý phần đầu (header) sẽ hủy bỏ phần này khỏi thông báo rồi gửi phần còn lại lên mức cao hơn.

III - MỘT KINH NGHIỆM CÀI ĐẶT

Mô hình trình bày ở trên của hệ phần mềm truyền thông đã được cài đặt thử nghiệm trên một máy sử dụng hệ điều hành UNIX và được nối vào một mạng cục bộ dạng Ethernet (giao thức CSMA/CD). Vì lý do chuyển mang (portability), phương án cơ bản nhất của UNIX 1A V7 và ngôn ngữ lập trình C được chọn để thử nghiệm.

Có hai phương pháp cài đặt hệ phần mềm truyền thông dưới UNIX:

- Cài đặt ngay trong nhân UNIX, như một chương trình điều khiển tập đặc biệt (driver). Tuy kiểu cài đặt này cho hiệu quả cao nhưng lại làm cồng kềnh nhân của UNIX và do vậy ảnh hưởng đến các tiến trình người sử dụng. Ngoài ra, việc kiểm tra một driver lớn như thế là khá khó khăn, đòi hỏi phải biên dịch lại toàn bộ nhân nhiều lần và do đó phải chiếm dụng toàn bộ máy (cho dù UNIX là hệ nhiều người sử dụng).
- Cài đặt như một chương trình người sử dụng bình thường. Việc kiểm tra chương trình là dễ dàng hơn. Ngoài ra, việc phát triển phần mềm trong môi trường người sử dụng cho phép tránh sử dụng các chức năng đặc biệt của nhân UNIX và do đó dễ chuyển mang hơn.

Phương pháp thứ hai đã được chọn để cài đặt thử nghiệm. Phần mềm truyền thông, bao gồm cả các thực thể, tạo thành một tiến trình duy nhất. Ở thời điểm khởi động, tiến trình này tạo ra tập truyền thông (communication file) và đặt số liệu tiến trình của nó vào tập này. Các tiến trình sử dụng sau đó sẽ đặt yêu cầu và dữ liệu của chúng trong tập (các tập là phương tiện duy nhất để truyền thông giữa các tiến trình độc lập trong UNIX V7). Cơ chế "khóa" cho phép truy nhập độc quyền vào tập truyền thông.

Tiến trình truyền thông truy nhập các driver của các mạng khác nhau thông qua các giao diện mang của nó.

Giao diện người sử dụng được cài đặt dưới dạng một thư viện "mạng". Nó cho phép các tiến trình người sử dụng tạo lập các yêu cầu của chúng theo cú pháp của các thực thể mà chúng muốn truy nhập. Chẳng hạn, để yêu cầu một liên kết ở mức "mạng" (network layer), chương trình sử dụng phải gọi hàm NCON - request của thư viện "mạng" đó.

Một đặc điểm khác của phần mềm được cài đặt là việc quản lý bộ nhớ động.

Tiến trình truyền thông có bộ nhớ riêng của nó trong đó quản lý các danh sách móc nối. Điều đó cho phép phần mềm truyền thông độc lập hơn với hệ điều hành và các chương trình tiện ích hệ thống, đồng thời có thể thay đổi kích thước bộ nhớ riêng theo khả năng của máy. Đây cũng là một cách để nâng cao tính khả chuyển của phần mềm truyền thông.

Trong quá trình cài đặt thử nghiệm, vấn đề đánh giá độ hiệu quả (performance) của hệ thống cũng là một mục tiêu quan trọng để rút kinh nghiệm. Việc đánh giá sẽ cho phép ta xác định các giá trị của các tham số cài đặt. Các yếu tố để đánh giá hiệu quả của hệ thống cần được xác định ngay ở giai đoạn thiết kế căn cứ vào nhu cầu của người sử dụng và công nghệ của mạng máy tính được sử dụng. Phần mềm truyền thông được cài đặt phải đạt được một mức độ mềm dẻo cao, cho phép thích nghi với các môi trường tác nghiệp luôn thay đổi (công nghệ mạng, các đặc trưng của máy tính...).

Qua việc xem xét tập các tham số có thể ảnh hưởng đến độ hiệu quả của một hệ thống, ta có thể quan tâm đến hai lớp tham số, đó là:

- Các tham số của các giao thức, phụ thuộc vào các chuẩn về giao thức (của ISO,...), chẳng hạn: kích thước của đơn vị dữ liệu (PDU - Protocol Data Unit), kích thước của cửa sổ trong cơ chế kiểm soát dòng dữ liệu (flow control),...
- Các tham số hệ thống, liên quan chặt chẽ với các đặc tính của các máy tính thành phần của mạng, chẳng hạn: kích thước bộ nhớ phân chia, số lượng các liên kết đồng thời,...

Dễ dàng chuyển mang là một trong những mục tiêu quan trọng nhất của hệ truyền thông được cài đặt. Để thử nghiệm khả năng đó, hệ mềm truyền thông sau khi được cài đặt đã được chuyển sang cho một máy tính cá nhân sử dụng MS-DOS. Việc chuyển các bộ phận độc lập với hệ thống chủ như là các thực thể hoặc Entity Manager là đơn giản hơn nhiều so với các bộ phận phụ thuộc (các giao diện). Khi chuyển mang như vậy, ta cần giải quyết các vấn đề sau:

- Về phần cứng: kích thước bộ nhớ trong của máy đích phải thừa sức bao kích thước của phần mềm truyền thông; tốc độ xử lý của máy tính cũng có ảnh hưởng đến số lượng các liên kết đồng thời (tham số này có thể thay đổi khi khởi động phần mềm truyền thông).
- Về ngôn ngữ lập trình: hiển nhiên là ngôn ngữ lập trình đã chọn phải khả dụng trên máy đích; tuy nhiên nên dùng ngôn ngữ chuẩn đã cài đặt.
- Về hệ điều hành: sự khác nhau giữa UNIX và MS-DOS là khả năng đa chương trình. UNIX là hệ đa nhiệm, nhiều người sử dụng, còn MS-DOS là đơn nhiệm và một người sử dụng. Như vậy, vấn đề đầu tiên cần giải quyết là phải có một môi trường đa nhiệm trong MS-DOS (ví dụ NCX là một hệ thống như vậy).

Nhận ngày 20-6-1991

TÀI LIỆU THAM KHẢO

1. ISO, Data Processing Systems - Data Communications Basic Reference Model. ISO/IS - 7498.
2. A. Benkiran, Communications dans les systems distribues. These de doctorat, Sept. 1988.
3. M. Schwartz, Telecommunication Networks. Addison - Wesley Publishing Co., 1987.
4. Nguyen Thuc Hai, Design and Implementation of a X25 packet multi interface portable software. Transactions of the IEICE Japan, Vol. E 71, No. 2, 1988, pp. 147-151.

ABSTRACT

A MODEL FOR IMPLEMENTING NETWORK PROTOCOL SOFTWARE

A model for implementing portable communication protocols is described. This study has been done to make portable network software, concerning in particular OSI standard protocols. This paper shows, first, a methodology and tools for communication protocol implementation. It also describes the practical problems met in the experimental implementation under UNIX operating system.

32. TIN TỨC HOẠT ĐỘNG

HỘI THẢO

" ỨNG DỤNG MÁY TÍNH TRONG QUẢN LÝ GIÁO DỤC ĐẠI HỌC "

Từ ngày 2 đến ngày 9 tháng 8 năm 1990, khóa bồi dưỡng và hội thảo về ứng dụng máy tính trong quản lý giáo dục đại học đã được tiến hành tại Hà Nội do Bộ Giáo dục và Đào tạo tổ chức với sự giúp đỡ của UNESCO PROAP.

Tham dự Hội thảo có 58 thành viên của 23 trường đại học từ khắp các vùng đất nước và 9 cơ quan nghiên cứu và quản lý thuộc Bộ Giáo dục và Đào tạo, trong đó có 30 thành viên là Hiệu trưởng, Hiệu phó, Vụ trưởng, Vụ phó, Viện phó.

UNESCO PROAP đã cử tiến sĩ Anthony Marshall, chuyên gia về tin học và quản lý giáo dục đại học đến giúp đỡ tiến hành hội thảo và trao đổi kinh nghiệm với các thành viên.

Hội thảo đã được nghe trình bày và thảo luận nhiều vấn đề nóng hổi và thiết thực về quản lý đại học trên quan điểm về tin học, về các phương pháp xây dựng hệ thống quản lý có máy tính hỗ trợ và những điều kiện để xây dựng các hệ thống đó. Hội thảo đã nghe phổ biến kinh nghiệm của các nước khu vực châu Á Thái Bình Dương về lĩnh vực này và trao đổi những bài học rút ra từ thực tiễn của các trường đại học ở Việt Nam.

Hội thảo đã đi đến đề xuất một số kiến nghị sau đây đối với lãnh đạo Bộ Giáo dục và Đào tạo và lãnh đạo các trường đại học:

1. Bộ Giáo dục và Đào tạo cần sớm đề xuất một chương trình hành động để điều phối công tác tin học hóa quản lý giữa các trường đại học trong Bộ, phụ trách chương trình này nên là một đồng chí trong Lãnh đạo Bộ. Ban chủ nhiệm chương trình này bao gồm ít nhất một cán bộ am hiểu tin học và có nhiệt tình, một số cán bộ phụ trách các vụ liên quan và một số chuyên gia tin học giỏi hỗ trợ. Ban chủ nhiệm chương trình cần đưa ra một kế hoạch làm việc trong thời gian 5 năm, có mục tiêu cụ thể cho từng bước, có đảm bảo kinh phí đầy đủ để triển khai công việc. Cần có qui định cụ thể về sự phối hợp giữa Ban chủ nhiệm chương trình của Bộ và các ban điều hành tiến trình tin học hóa quản lý ở các trường đại học.

2. Mỗi trường đại học cần xây dựng một đề tài triển khai tin học hóa quản lý trực thuộc chương trình quản lý tin học hóa của Bộ. Ban chủ nhiệm đề tài này do một hiệu trưởng hoặc hiệu phó phụ trách, trong số thành viên có ít nhất một cán bộ am hiểu tin học và có nhiệt tình, một số chuyên gia tin học thành thạo và đại diện của các bộ phận chức năng liên quan. Cần đề ra một kế hoạch cụ thể cho công tác này.

3. Cần sớm tổ chức cung cấp hiểu biết tin học và tạo thói quen sử dụng công cụ tin học trong cán bộ giảng dạy ở cơ quan bộ và ở các trường đại học. Ở mỗi đơn vị, ngoài các trung tâm máy tính mạnh cần tăng cường một bước mở rộng trang bị máy vi tính cho một số bộ phận khác có nhu cầu. Mở rộng trang bị thêm máy vi tính sẽ tạo nên đội ngũ đồng đảo những người có hiểu biết và có thói quen sử dụng máy vi tính phục vụ quản lý.

4. Bộ Giáo dục và Đào tạo nên đóng vai trò tiên phong trong việc đưa tin học vào mọi hoạt động xã hội, trước hết là có biện pháp tích cực để đào tạo đội ngũ giáo viên về tin học.

5. Cần tổ chức các liên kết quốc tế để thường xuyên nhận được những kinh nghiệm và thông tin mới nhất của nước ngoài trong lĩnh vực tin học hóa quản lý đại học, trao đổi những phần mềm chuyên dụng, trao đổi chuyên gia và tổ chức các cuộc tham quan các cơ sở có nhiều kinh nghiệm về lĩnh vực này ở các nước láng giềng. Đề nghị Bộ đặt vấn đề với một số nước ngoài trước hết là khu vực Châu Á Thái Bình Dương, giúp đỡ ngành đại học trong lĩnh vực này.

Chủ nhiệm Workshop
GS. TS. LÂM QUANG THIỆP
(Vụ trưởng Vụ Đào tạo đại học)