

VIỆN KHOA HỌC VIỆT NAM

**KHOA HỌC
TÍNH TOÁN và
ĐIỀU KHIỂN**

HÀ NỘI

Tập III—Số 3

1987

CV 14

Tập chí

KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN

XUẤT BẢN VÀO CÁC THÁNG 3, 6, 9, 12

Tổng biên tập : PHAN ĐÌNH DIỆU

Hội đồng biên tập :

PHAN ĐÌNH DIỆU, NGUYỄN VĂN BA
NGUYỄN GIA HIỆU, NGUYỄN XUÂN HUY
BẠCH HƯNG KHANG, NGUYỄN THỨC LOAN
NGUYỄN XUÂN LỘC, LÊ THIỆN PHỐ
NGUYỄN VĂN QUÝ, NGUYỄN XUÂN QUỲNH
PHẠM HỮU SÁCH, HỒ THUẬN, HOÀNG TỤY
HOÀNG HỮU TIẾN, BÙI ĐOÀN TRỌNG

Trụ sở : 70 Trần Hưng Đạo, Hà Nội. Dãy số: 52825

Rus.

MỘT SỐ PHƯƠNG PHÁP TỰ ĐỘNG HÓA TỔNG HỢP VĂN BẢN

ĐÓ VIỆT-NGA

Tự động hóa trong lĩnh vực hành chính sự nghiệp ngày càng trở nên vấn đề cấp thiết của một nền kinh tế hiện đại. Tốc độ cơ khí hóa nhanh chóng trong mọi lĩnh vực kinh tế quốc dân gây ra khoảng cách lớn giữa sản xuất và bộ máy điều hành. Sự chênh lệch giữa một bên là nền sản xuất được tự động hóa cao và một bên là bộ máy hành chính công kênh, lâu đời, ít thừa theo kiểu thủ công làm giảm đáng kể năng suất lao động. Theo thống kê trong khi phần lớn nền kinh tế ở một số nước phương Tây cho thấy ở nhiều lĩnh vực, trên 50% giá thành sản phẩm là do chi phí cho các giấy tờ, các khâu hành chính để sản phẩm có thể đi từ nguyên liệu cho đến tay người tiêu dùng [1].

Do đó việc tăng năng suất lao động trong các khâu hành chính ngày càng được quan tâm ở các nước tiên tiến. Trong công việc hành chính, các phần việc liên quan tới văn bản, giấy tờ lại là phần việc phổ biến và rắc rối nhất. Vì vậy, trong công cuộc «Tự động hóa văn phòng» việc đầu tiên người ta quan tâm tới là tự động hóa các khâu văn bản giấy tờ.

Hàng loạt các kiểu hệ thống soạn thảo văn bản được ra đời từ thập kỉ 70. Ban đầu chúng được dùng để phục vụ cho các thảo chương viên, giảm đáng kể lao động của họ. Dần dần các hệ thống này được cải tiến và được dùng để phục vụ cả việc soạn thảo văn bản. Ngày nay trên thế giới đã có tới hàng trăm hệ thống soạn thảo khác nhau, ngày càng thích hợp hơn với nhu cầu của người sử dụng.

Các hệ thống soạn thảo văn bản hiện hành có thể chia làm hai kiểu:

1. Kiểu vận năng: Có thể sử dụng chúng để soạn thảo một văn bản bất kỳ (ví dụ như hệ thống kiểu WORDSTAR). Các hệ thống này có ưu thế ở điểm vận năng, nhưng thường bị hạn chế ở tốc độ (tốc độ vào dữ liệu ngang với tốc độ đánh máy chữ).

2. Kiểu điền vào mẫu có sẵn: Kiểu này thường được dùng trong các công đoạn sản xuất chuyên dụng. Các văn bản được thảo sẵn để chừa cho một vài thông số, và hàng ngày, khi cần thiết các chỗ trống được điền thích hợp. Cách soạn thảo này cho tốc độ lớn, nhưng lại bị hạn chế bởi khuôn khổ của văn bản thảo sẵn. Mặt khác không phải loại văn bản nào cũng có thể mẫu hóa theo cách đó.

Trong bài báo này tác giả muốn trình bày một cách tiếp cận dung hòa giữa hai cách trên nhằm xây dựng các phương pháp tổng hợp văn bản cho phép tăng tốc độ soạn thảo văn bản, mặt khác không hạn chế tính vận năng của hệ thống.

Tác giả đề xuất hai phương pháp tổng hợp văn bản và cách thiết kế trên máy tính điện tử.

Trong bài có giới thiệu hệ thống tự động hóa soạn thảo văn bản đã được xây dựng dựa trên cơ sở phương pháp đã đề xuất.

I - CÁCH TIẾP CẬN NHẪM XÂY DỰNG PHƯƠNG PHÁP TỔNG HỢP VĂN BẢN

Theo cách tiếp cận này một hệ thống tổng hợp văn bản được xây dựng từ hai thành phần: phần hệ thống cơ sở và phần đặc thù.

Phần hệ thống cơ sở làm công việc điều hành, tổng hợp văn bản dựa trên thông tin lấy từ phần đặc thù.

Phần đặc thù chứa đựng các thông tin số dạng của một loại văn bản.

Phần hệ thống được xây dựng bởi người lập trình, còn phần đặc thù do chuyên gia, chuyên trách việc soạn thảo tài liệu tự tạo lập trên cơ sở phân tích và kinh nghiệm sẵn có về đặc thù của loại tài liệu trong phạm vi trách nhiệm của mình.

Mọi hệ thống gồm phần hệ thống cơ sở và phần đặc thù tạo nên một hệ chuyên để soạn thảo một vài loại thư tín, công văn, tài liệu.

Bằng cách đó, hệ thống có thể tăng tốc độ soạn thảo một cách đáng kể dựa trên các thông tin đặc thù đã được phân tích và lập sẵn. Mặt khác, tính vạn năng của hệ thống có thể được tăng thêm bằng cách mở rộng phần đặc thù.

Như vậy, ta có thể thu được một hệ thống có tốc độ soạn thảo nhanh và theo một nghĩa nào đó, vẫn có tính vạn năng. Hơn thế nữa, hệ thống luôn có thể được bổ sung, mở rộng bởi công lao động của bản thân người sử dụng, thường là các nhân viên thư ký.

Dựa trên cách tiếp cận đã trình bày, tác giả đề xuất hai phương pháp tổng hợp văn bản.

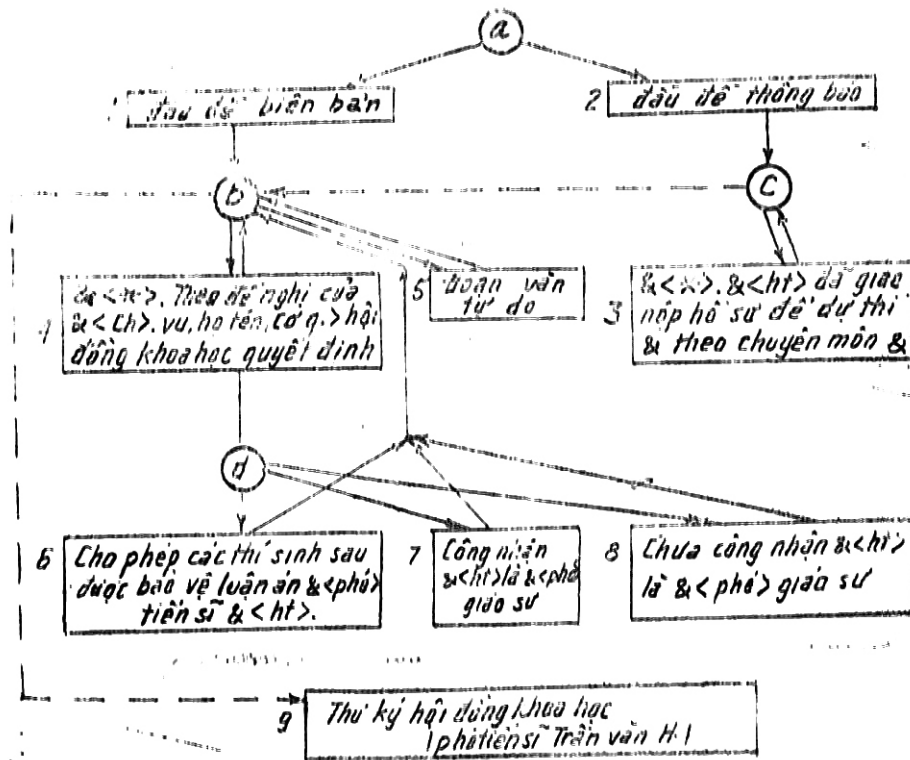
II - PHƯƠNG PHÁP TỔNG HỢP VĂN BẢN DỰA TRÊN VĂN BẢN MẪU

Phương pháp này giả thiết rằng, các văn bản được tạo ra từ các đoạn văn. Các đoạn văn lại có thể chuẩn hóa, thảo sẵn, lưu trữ trong máy tính và được sử dụng trong quá trình soạn thảo. Các đoạn văn soạn sẵn cũng có thể chứa các phần trống được điền giá trị cụ thể trong quá trình soạn thảo.

Phần đặc thù ở phương pháp này chứa một loạt các maket (văn bản mẫu). Mọi văn bản mẫu chứa một số các đoạn văn mẫu, các phần trống để điền và các thứ tự xuất hiện cho phép của các đoạn văn.

Phần hệ thống cơ sở ở phương pháp này là một chương trình cho phép sử dụng maket để soạn thảo một hoặc một loạt văn bản, trong đó có thể lựa chọn đoạn văn, lập một đoạn văn ở nhiều chỗ, điền giá trị phần trống từ bàn phím hoặc từ một danh sách soạn sẵn.

Ta có thể xét phương pháp này trên một ví dụ. Với mẫu văn bản trên hình 1 ta có thể xây dựng các biên bản và các thông báo khác nhau. Một trong các biên bản có ở hình 2.



Hình 1

BIÊN BẢN SỐ 16

Phiên họp Hội đồng Khoa học viện...

ngày 12-04-1986

Cuộc họp bắt đầu từ 14h đến 16h tại hội trường của Hội đồng Khoa học viện...

Trong cuộc họp có sự tham gia của các vị: giáo sư tiến sĩ Lê Thanh M., giáo sư p/s Trần Việt A....

- 4
+
6
1. Theo đề nghị của viện KHTTĐK Hội đồng Khoa học quyết định cho các thí sinh sau đây bảo vệ luận án phó tiến sĩ: Nguyễn Văn A và Lê thị B.
- 4
+
7
2. Theo đề nghị của chủ tịch Hội đồng giám khảo chọn giáo sư. Hội đồng khoa học quyết định công nhận p/s Trần H là phó giáo sư.
- 5
3. Hội đồng khoa học quyết định trao tặng bằng khen kèm phần thưởng hiện vật trị giá 1000đ cho...

9

*Thư ký Hội đồng khoa học
p/s Trần Văn H*

Hình 2

Trong hình 1 các mũi tên liền nét chỉ thứ tự xuất hiện của các đoạn văn, các mũi tên không liền nét chỉ điểm ra của các vòng lặp. Các biên bản, thông báo lập từ mẫu văn bản này là một đường đi (có thể có vòng lặp) từ điểm a cho tới điểm 9. Các nút tròn a, b, c... chỉ các nút cho phép lựa chọn. Biên bản trên hình 2 được tạo lập khi hệ thống đi từ a tới 9 theo đường a1b4d6b4d7b5b9. Các chỗ được điền thêm từ bản phim được đánh dấu trong ví dụ bằng cách gạch dưới chân.

Cần lưu ý là ở đây thứ tự xuất hiện của các đoạn văn dường như gõ văn bản theo một khuôn mẫu, nhưng thực ra thứ tự này hoàn toàn do người sử dụng quyết định, thậm chí có thể vứt bỏ thứ tự (ví dụ, các đoạn văn 4,5 hoặc 6, 7, 8 là không có thứ tự trước sau. Giữa chúng có thể lựa chọn tùy ý). Thứ tự giữa các đoạn văn được dựa vào văn bản mẫu với một dụng ý hoàn toàn khác. Ta xét, giả sử trong văn bản mẫu có phần về việc bầu cử. Đoạn văn đầu tiên về phần bầu cử phải là đoạn văn giới thiệu mục đích, sau đó đến thành phần bầu cử và cuối cùng là kết quả. Ba đoạn văn này nhất thiết phải theo thứ tự đã mô tả và nếu đoạn văn đầu tiên không xuất hiện thì cả hai đoạn văn sau cũng bị bỏ qua, có nghĩa là trong văn bản không có phần bầu cử. Qua ví dụ ta có thể nhận thấy là thứ tự được đưa vào để giúp người sử dụng có thể lựa chọn theo nội dung các đoạn văn.

III - HỆ THỐNG TỰ ĐỘNG HÓA SOẠN THEO VĂN BẢN (АСИДО)

Hệ thống được thiết kế và cài đặt dựa trên phương pháp đã được mô tả ở phần II.

Hệ thống cơ sở bao gồm ba modul. Modul chính thực hiện việc hiện các đoạn văn từ văn bản mẫu ra màn hình, cho phép người sử dụng được lựa chọn đoạn văn cần thiết điền các chỗ trống hoặc đưa đoạn văn mới vào văn bản từ bản phim. Quá trình soạn thảo chỉ kết thúc khi văn bản được hoàn thành.

Hệ thống có cả các chức năng kể từ bên phải, cho phép phân chia từ đối với các ngôn ngữ Anh, Nga, Bun; chức năng phân trung, soạn thảo một loạt thư tín cùng kiểu cho nhiều người nhận khác nhau, v.v...

Hệ thống đã được thử nghiệm cho các công văn, biên bản, tài liệu hành chính của Hội đồng Khoa học, viện Hàm lâm Bungari. Hệ cũng đã được thử nghiệm trong việc soạn thảo văn bản tiếng Việt.

Hệ thống АСИДО được viết trên ngôn ngữ lập trình PASCAL cho máy vi tính 16 bit kiểu PC/IBM hoặc tương thích, với cấu hình tối thiểu.

IV — PHƯƠNG PHÁP TỰ ĐỘNG HÓA TỔNG HỢP VĂN BẢN BẰNG CÁCH VIẾT TẮT

Phương pháp này giả thiết là trong mọi lĩnh vực đời sống hàng ngày các từ ngữ được dùng thường có đặc thù riêng. Có nghĩa là một số từ hoặc nhóm từ được dùng lặp đi lặp lại nhiều lần. Sử dụng tính chất này ta có thể tạo lập một hệ thống cho phép ghi tóm tắt một số từ, nhóm từ.

Phần đặc thù trong phương pháp này là một từ điển do bản thân người sử dụng lập ra.

Phần hệ thống cơ sở là một bộ soạn thảo văn bản có khả năng hiểu được cách viết tắt của người sử dụng.

Để hiểu rõ hơn, ta lấy một ví dụ viết tắt thường gặp trong văn viết « v.v... » có nghĩa là « vân vân » hay « đ/c » có nghĩa là « đồng chí ». Trong hệ thống cho phép cả hai kiểu viết tắt: viết tắt bằng nhóm chữ đầu tiên và dấu viết tắt (ví dụ như « v. ») hoặc viết tắt bằng kí hiệu riêng. Để viết tắt theo kí hiệu riêng người sử dụng cần phải khai báo trước mình kí hiệu của mình khi đưa từ tương ứng vào từ điển. Khi viết tắt theo nhóm chữ đầu người sử dụng có thể viết dài tùy ý (ví dụ có thể viết « v. », « vâ. », « vân. » thay cho từ « vân » có trong từ điển).

Phương pháp đặc biệt hiệu quả đối với các ngôn ngữ đa âm với các từ dài.

Hai phương pháp trên ngoài việc tăng đáng kể tốc độ soạn thảo dữ liệu còn cho phép giảm phần lớn các lỗi chính tả, cú pháp, lỗi hành văn. Các phương pháp này đòi hỏi phải được thử nghiệm, nghiên cứu sâu sắc cách làm việc của các nhân viên thư ký. Hiện nay các phương pháp này vẫn đang tiếp tục được nghiên cứu và phát triển.

Nhận ngày 1-3-87

TÀI LIỆU THAM KHẢO

1. К. Кочетков, Автоматизация конторского труда в США издательство наука, 1985.
2. П. Бърнев, До Виет Ига, Система автоматизированного составления документов. Сб. Математика и математическо образование, 1986.
3. До Виет Ига, Система автоматизированного составления свободных текстов. Труды конференции КИВЗТ, Будапешт, 1986.
4. До Виет Ига, Система автоматизированного синтеза документов. Труды школы «Программирование 86», 1986.

РЕЗЮМЕ

НЕКОТОРЫЕ МЕТОДЫ АВТОМАТИЗИРОВАННОГО СИНТЕЗА ДОКУМЕНТОВ

В работе описываются два метода автоматизированного синтеза документов: метод синтеза по текстовым образцам и метод синтеза по сокращенным словам. Первый метод служит основой для построения автоматизированной системы составления документов АСИДО. Данная система экспериментировалась в некоторых учреждениях. Система может обрабатывать тексты написанные на русском, болгарском и Вьетнамском языках.

TẮM CPU-Z80 TRÊN BUS VT-64

NGUYỄN VĂN TAM

Viện KHTT và ĐK

MP-Z80 có những ưu việt về cả 7 cấu trúc và hệ lệnh so với tất cả các bộ vi xử lý 8 bit khác. Khả năng này cho phép thiết kế các hệ vi xử lý với giá thành thấp và tính năng của hệ lại mạnh. MP-Z80 được nhiều hãng sản xuất, nhiều hãng sử dụng để xây dựng bán thành phẩm, lại có hệ lệnh hoàn toàn tương thích với MP-8080 nên khả năng tiếp thu và chuyển đổi sản phẩm, đặc biệt là phần mềm rất thuận lợi. Tăm CPU-Z80 được thiết kế cho BUS VT-64 và là 1 trong những tấm chuẩn trong hệ tám chuẩn VT. (Tạp chí khoa học tính toán và điều khiển số 2-1986).

1. Một số đặc điểm của bộ vi xử lý MP-Z80

Về cấu trúc MP-Z80 có nhiều đặc điểm, đặc biệt là hệ thống các thanh ghi như hình 1 và hình 2. Thanh ghi PC chứa địa chỉ của lệnh đang thực hiện, thanh SP chứa địa chỉ của ngăn xếp. Ngăn xếp có thể đặt ở bất cứ địa chỉ nào của RAM. Các thanh ghi IX và IY được sử dụng trong chế độ địa chỉ hóa dùng chỉ số. Ở chế độ này thanh ghi IX và IY chứa địa chỉ cơ sở của vùng số liệu. MP-Z80 có chế độ ngắt mà 8 bit cao của địa chỉ làm tươi cho RAM động. Nội dung của R sẽ tăng 1 sau mỗi chu kỳ nạp lệnh. Địa chỉ làm tươi được đưa ra 7 bit thấp của địa chỉ khi bên trong MP đang giải mã và chuẩn bị lệnh. Thanh ACC chứa kết quả của các phép tính số học và logic. Các cờ F cho biết kết quả của các phép tính. MP-Z80 có 2 bộ thanh ghi đa năng. Mỗi bộ có 3 đôi thanh ghi BC, DE, HL. Bằng lệnh người ta có thể chọn bộ thanh ghi này hay thanh ghi kia, đặc biệt khi sử dụng ngắt. MP-Z80 có thể thực hiện 158 lệnh và là MP có hệ lệnh mạnh nhất trong tất cả các bộ vi xử lý 8 bit.

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

Các thanh ghi chính

Các thanh ghi phụ

Hình 1

Các thanh ghi đa năng

Vector ngắt I	làm tươi R
Thanh ghi chỉ số IX	
Thanh ghi chỉ số IY	
Con trỏ ngăn xếp SP	
Thanh đếm chương trình PC	

Hình 2

Các thanh ghi chuyên dụng

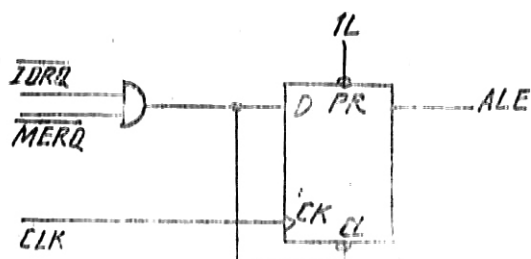
2. Đệm BUS

Các đường địa chỉ được khuếch đại qua các mạch 74LS244.

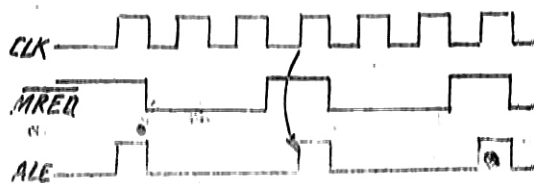
Các đường địa chỉ A12, A13, A14, A15, sau khi RESET được chọn từ ngoài MP (A12, A13, A14, A15). Bằng cách này cho phép ta bắt đầu chương trình với địa chỉ khác 0000H, ví dụ F000H.

Số liệu được khuếch đại và điều khiển bởi thanh đệm 2 chiều. Mạch này được chọn khi MP không bị treo và chỉ trong khoảng thời gian tín hiệu ALE có mức logic thấp. Chiều mở của 74LS245 do tín hiệu SI quy định.

BUS VT-64 được sử dụng cho nhiều loại MP khác nhau nên một số tín hiệu của BUS điều khiển của MP-280 phải được sửa hoặc tạo dạng để thích ứng. Tín hiệu SI được tạo từ tín hiệu R, tín hiệu INTA = $\overline{M1+IORQ}$, tín hiệu IO/M = \overline{MREQ} . Tín hiệu ALE được tạo từ đồng hồ nhịp CLK và tín hiệu MREQ như hình 3, và hình 4. Tín hiệu \overline{RD} của MP-280 trước khi đưa ra BUS hệ thống cũng được sửa dạng theo hình 5 và hình 6.



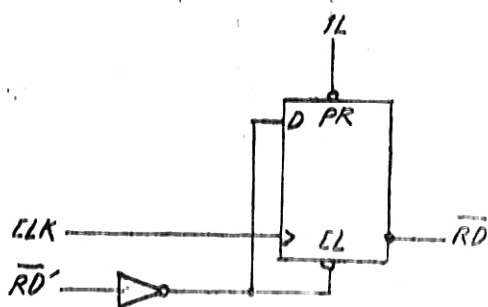
Hình 3
Tạo tín hiệu ALE



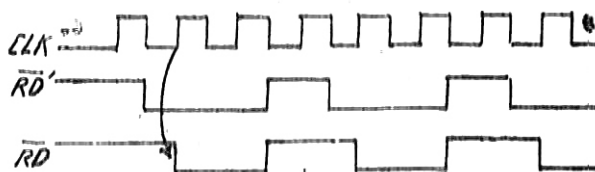
Hình 4
Biểu đồ thời gian tạo tín hiệu ALE

Các tín hiệu điều khiển như \overline{RD} , \overline{WR} , INTA, IO/M, HLDA, SI, ALE đều được khuếch đại trước đưa ra BUS. BUS địa chỉ, BUS số liệu và BUS điều khiển khi làm việc ở chế độ DMA đều bị treo.

Để cách ly các mạch được sản xuất bằng công nghệ MOS như các mạch nhớ và các mạch ngoại vi với BUS hệ thống, một tầng đệm nữa được sử dụng với các mạch 74LS245 cho BUS số liệu và 74LS244 cho BUS địa chỉ và điều khiển. Tầng khuếch đại đệm này nhằm tăng độ tin cậy cho toàn hệ thống.



Hình 5
Tạo tín hiệu \overline{RD}



Hình 6
Biểu đồ thời gian tạo tín hiệu \overline{RD}

3. Giải mã

Để tạo điều kiện cho việc sử dụng vùng nhớ một cách mềm dẻo và linh hoạt, hai mạch giải mã 74LS138 cùng với các giắc cắm chuyên đổi đã được dùng. Cách giải mã này cho phép dung lượng mạch nhớ có thể thay đổi (2kB, 4kB, 8kB) và địa chỉ của từng mạch có thể chọn thích hợp.

Địa chỉ các mạch ngoại vi trên tấm CPU-Z80 được định nghĩa theo bảng sau.

Địa chỉ	Mạch	Chú giải
00 01	8251-I	Cửa số liệu của 8251.I Cửa điều khiển, trạng thái của 8251-I
02 03	8251-II	Cửa số liệu của 8251.II Cửa điều khiển, trạng thái của 8251-II
04 05 06 07	8253	Bộ đếm CTO của 8253 Bộ đếm CT1 của 8253 Bộ đếm CT2 của 8253 Thanh ghi điều khiển của 8253
08 - 09	5016	Chọn tần số cho 5016 và chuyển địa chỉ A'12, A'13, A'14, A'15 sang A12, A13, A'14, A'15
0A - 0B	74LS74	Số liệu 00 cho phép giải mã các vở nhớ 01 cấm giải mã các vở nhớ.
0C - 0D	74LS74	Tạo địa chỉ A16

Hình 7 Địa chỉ, các cửa vào/ra của tấm CPU-Z80

4. Các mạch nhớ và ngoại vi.

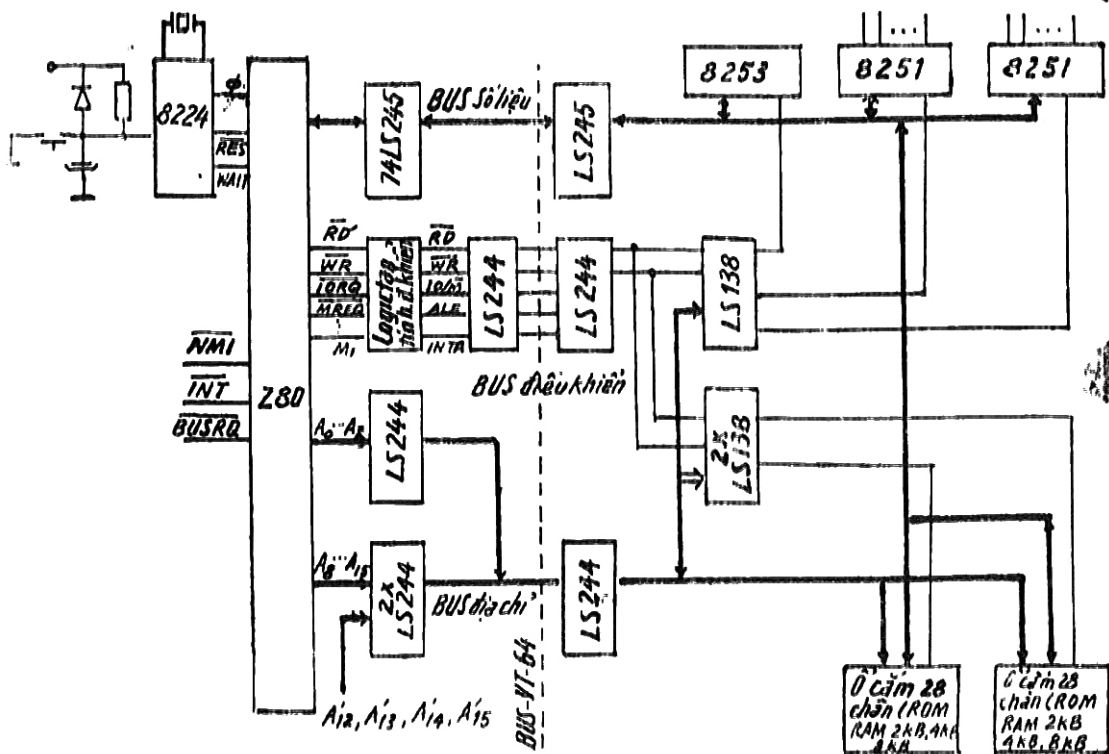
Trên tấm CPU-Z80 có 3 ð cắm 28 chân. Bằng cách chọn địa chỉ và các tín hiệu điều khiển, trên các ð cắm này có thể đặt các mạch nhớ ROM hoặc RAM với dung lượng loại 2kB, 4kB hay 8kB. Địa chỉ các mạch nhớ có thể thay đổi trong 32kB đầu hay cuối của bộ nhớ. Hai mạch nhớ này tạo thành một băng khi cấu thiết có thể cắt bằng tín hiệu cấm giải mã nhớ (Lệnh OUT ra cửa OA hay OB với số liệu 01). Ví dụ khi ta đưa vào hệ vi tính tấm DRAM 64kB.

Hai mạch USART - 8251 có thể làm ở chế độ đồng bộ hoặc dị bộ được sử dụng để ghép nối với CRT, máy in hay truyền tin theo chuẩn RS-232 qua các mạch MC1488, MC1489.

Mạch 8253 với 3 bộ đếm 16 bit cho phép tạo ra những tín hiệu thời gian dưới sự điều khiển bằng chương trình. Mạch 8253 có thể làm việc ở 5 chế độ khác nhau.

Sơ đồ khối của tấm CPU-Z80 ở hình 8

Cấu trúc của tấm CPU-Z80 cho phép làm việc độc lập trong các hệ vi xử lý nhỏ, cùng với các tấm nhớ hay điều khiển ngoại vi khác trong hệ thống tấm chuẩn VT có thể xây dựng các hệ vi xử lý thích hợp dùng trong công nghiệp hay máy vi tính dùng trong khoa học và quản lý.



Hình 8 Sơ đồ khối CPU-Z80

TÀI LIỆU THAM KHẢO

1. Solutions Microsystemes. Cartes Micro-ordinateurs 8 bits Eurotechnique, Mars 1982
2. Cartes europe BUS G. 64.
3. Nguyễn Văn Tam và nhóm... Các tiêu chuẩn của hệ vi xử lý VT. Tạp chí KHIT và ĐK số 2-1986.

ABSTRACT

CARD CPU-Z80 ON THE BUS VT-64

Card CPU-Z80 was designed to connect to BUS VT.64. This card is capable to operate like a single-board computer for a small Microsystem. When connected to other card via BUS VT.64, the CPU-Z80 works like CPU-part of Microsystem.

CÀI ĐẶT NGÔN NGỮ PADRE BẰNG KỸ THUẬT TIỀN XỬ LÝ

NGUYỄN VĂN LƯU, ĐHTH Ple Quyri và
Mari Quyri.

HỒ THUẦN, Viện Khoa học Hình toán
và điều khiển.

TÓM TẮT

Trong bài báo này những bước khác nhau và những ưu điểm của kỹ thuật tiền xử lý đã được chỉ ra thông qua cài đặt cụ thể ngôn ngữ PADRE, được thiết kế nhằm mô tả, biến đổi và thể hiện những lớp mạng Petri khác nhau. Ở đây chúng tôi cũng chứng tỏ rằng kỹ thuật tiền xử lý này hoàn toàn có thể áp dụng được trên máy vi tính.

1. MỞ ĐẦU

Trong [1] đã chứng tỏ rằng PADRE là ngôn ngữ lập trình thử nghiệm có thể dùng để mô tả, biến đổi và thể hiện những lớp mạng Petri khác nhau, chẳng hạn mạng Petri thông thường, mạng Petri có trọng số ([2]), mạng Petri màu ([3]), mạng Petri tân từ/chuyển ([4]),... Trong bài báo này sẽ trình bày một kỹ thuật cài đặt cụ thể ngôn ngữ Padre.

Với cách cài đặt này chúng tôi nhằm hai mục đích chính là tốc độ thực hiện và khả năng trao đổi. Mục đích thứ nhất có thể đạt được nhờ kỹ thuật tiền xử lý nhằm dịch từ ngôn ngữ nguồn cấp cao sang ngôn ngữ đích cấp cao khác. Vì ngữ nghĩa của các ngôn ngữ nguồn và đích thường dễ hiểu, nên quá trình dịch không có nhiều khó khăn lắm do có sự tương đương trực tiếp về ngữ nghĩa giữa chúng với nhau. Nhằm nâng cao khả năng trao đổi, chúng tôi đã chọn một ngôn ngữ đã được cài đặt trên phần lớn tất cả các hệ máy tính, đó là ngôn ngữ Pascal. Luôn hướng tới hai mục đích này, kỹ thuật tiền xử lý đã được đưa ra bao gồm 4 bước: (i) Tạo sinh các biểu diễn trong có các kiểu khác nhau cùng với những thông tin thích hợp để dùng về sau, (ii) dịch toàn bộ chương trình viết bằng ngôn ngữ PADRE sang chương trình viết bằng ngôn ngữ PASCAL tương đương về ngữ nghĩa, (iii) tạo sinh các thủ tục tiền xử lý đáp ứng nhu cầu của người sử dụng (iv) chuyển những thông tin ra ở các bước trước cho chương trình dịch PASCAL để hoàn thành quá trình dịch. Tất nhiên, thiết kế bộ tiền xử lý phải bao gồm tất cả những thuật toán chính quen biết trong quá trình dịch ([5]) như phân tích từ vựng và cú pháp, phân tích ngữ nghĩa, phát hiện lỗi,... chỉ có điều là mã đối tượng được tạo sinh dưới dạng ngôn ngữ cấp cao

Bài báo chia làm 3 phần. Trong phần 1 sẽ nhắc lại một vài cấu trúc cơ bản của ngôn ngữ PADRE, phần 2 mô tả kỹ thuật cài đặt, còn trong phần 3 chúng tôi sẽ chứng tỏ rằng kỹ thuật này hoàn toàn có thể dùng cho máy vi tính.

2. SƠ LƯỢC ĐÔI NÉT VỀ NGÔN NGỮ PADRE

Trong phần tóm tắt về ngôn ngữ PADRE được đưa ra sau đây chúng tôi chỉ nhắc lại một vài cấu trúc điển hình của ngôn ngữ này. Ngoài ra có thể tìm thấy mô tả chặt chẽ cú pháp ngôn ngữ PADRE trong [1] hay mô tả mở rộng của nó trong [6].

Đề khác phục những phiền phức khi mô tả, biến đổi và thể hiện các lớp mạng Petri khác nhau bằng các ngôn ngữ lập trình cấp cao hiện có, chúng tôi đã thiết kế ngôn ngữ PADRE được trang bị hệ thống kiểu dữ liệu và tập các chỉ thị và các câu lệnh sau đây:

2.1. Hệ thống các kiểu dữ liệu

Ngoài các kiểu dữ liệu sơ cấp chuẩn như CHAR, BOOLEAN, INTEGER, và miền số các số nguyên, ngôn ngữ PADRE còn có kiểu *xây mạng* với cú pháp được thể hiện bằng công thức chuẩn EBNF ([7]) sau đây:

net-id = « NET OF »
 [const - def « ; »]
 [type - def « ; »]
 element - decl « ; »
 connection - clause
 « END »

ở đây element - decl = « ELEMENT » elem - item { « ; » elem - item }
 elem - item = elem - id { « , » elem - id } « : » elem - struct
 elem - struc = « TRANSITION » | « SIMPLE PLACE » | « COLORED PLACE OF »
 int - subrange - id
 connection - clause = « CONNECTION » trans - connect [interface]
 { « | » trans - connect [interface] }

Trans - connect có cú pháp đơn giản như sau:

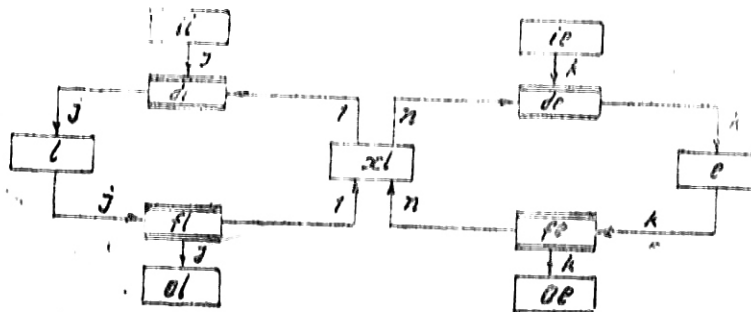
Trans - connect = trans - id « [»
 [in - clause « ; »]
 [guard - clause « ; »]
 [out - clause]
 «] »

ở đây với mỗi tên cái chuyển trans-id:

- in-clause đặc tả tất cả những in-places và những tiền điều kiện kèm theo của nó.
- guard-clause chỉ ra những ràng buộc bổ sung áp lên các biến có trong tiền điều kiện hoặc mặt tiếp giáp đồng bộ của nó với bên ngoài.
- out-clause đặc tả tất cả những out-places cùng với những hậu điều kiện kèm theo.
- interface đặc tả mặt tiếp giáp với bên ngoài.

Thay cho việc đi sâu vào những chi tiết cú pháp cụ thể đã được mô tả đầy đủ trong [6], chúng ta sẽ xem xét một ví dụ mô tả mạng Petri (đã chỉ ra trong [1]).

Ví dụ 1: Trên hình vẽ 1 mạng biểu diễn lời giải, trong đó không có cung ức chế của bài toán đợ ghi cổ điển ([8]). Sự liên kết trung nhập kéo theo 3 ràng buộc quen biết là (i) các thao tác ghi toán trừ lẫn nhau, (ii) cho phép các thao tác đợ tranh, (iii) các thao tác đợ và ghi toán trừ lẫn nhau.



Hình 1

ở đây il (tương ứng io) là vị trí chứa tất cả các quá trình đợi để đợ (tương ứng ghi),
 l (tương ứng o) là vị trí chứa tất cả các quá trình đợ (tương ứng ghi),
 ol (tương ứng oe) là vị trí chứa tất cả các quá trình đã đợ (tương ứng ghi) xong,
 xl là simple place,

di (tương ứng Fi) là cái chuyển biểu diễn điểm đầu (tương ứng điểm cuối) của quá trình đọc,

de (tương ứng fe) là cái chuyển biểu diễn điểm đầu (tương ứng điểm cuối) của quá trình ghi.

Rõ ràng là chúng ta đã có một mạng màu, nó có thể mô tả bằng ngôn ngữ PADRE như sau

```
const n = 13 ;
type np = 1.. n ;
rwnet = NET OF
    ELEMENT il, l ol, ie, e, oe: COLORED PLACE
        OF np ;
    xl: SIMPLE PLACE ;
    dl, fl, do, fe: TRANSITION ;
```

CONNECTION

```
di [ IN xl : (xl >= 1), il(j: np) : (il [j] >= 1) ;
    OUT l : (1 TO l [j]) ]
fl [ IN l(j: np) : (l [j] >= 1) ;
    OUT xl : (1 TO xl), ol : (1 TO ol [j]) ]
de [ IN xl : (xl = n), ie(k: np) : (ie [k] >= 1) ;
    OUT e : (1 TO e [k]) ]
fe [ IN e (k: np) : (e [k] = 1) ;
    OUT xl : (n TO xl), oe : (1 TO oe [k]) ]
END ;
```

Trên cơ sở toán tử xây mạng, ta có các thao tác sau đây trên mạng :

- Các thao tác bên trong mạng: các thao tác này cho phép thêm vào hay loại bỏ các phần tử của mạng.

- Các thao tác trên các mạng: như hợp 2 mạng bằng cách hợp nhất các phần tử có tên giống nhau, hợp các mạng bằng cách hợp nhất các phần tử có tên tương minh hoặc hợp các mạng bằng cách hợp 2 tập vị trí.

2.2. Tập các câu lệnh và chỉ thị

Tập này bao gồm các câu lệnh và các chỉ thị sau đây :

2.2.1. Các câu lệnh và chỉ thị truyền thống

- Câu lệnh gán, được định nghĩa như sau :

Biểu « := » biểu thức ;

- Câu lệnh chọn có điều kiện được mô tả nhờ câu lệnh IF

« IF »

guarded - command { « | » guarded - command }

« FI »

guarded - command = guard « : » statement - sequence ;

guard = boolean - expression ;

statement - sequence = statement { « ; » statement }

Trong câu lệnh IF, mỗi guarded - command sẽ được kiểm tra. Nếu guard tương ứng của nó đúng thì dãy câu lệnh statement - sequence sẽ được thực hiện.

- Câu lệnh LOOP không tiên định

« LOOP » guarded command { « | » guarded command }

« POOL »

Trong câu lệnh LOOP, các câu lệnh guarded - command được kiểm tra ngẫu nhiên. Nếu guard tương ứng đúng thì dãy câu lệnh statement - sequence đã được thực hiện. Câu lệnh LOOP chấm dứt thực hiện khi tất cả các guards đều sai. Câu lệnh IF và LOOP lấy ra từ [9] với đôi chút sửa đổi về ngữ nghĩa.

- Chỉ thị vào/ra: Tất cả các chỉ thị vào/ra của ngôn ngữ Pascal đều được giữ nguyên trong ngôn ngữ PADRE.

2.2.2. Những chỉ thị riêng với mạng

- Phát sinh một trạng thái tức thời của mạng. Chỉ thị « CREATE » (« net-var-id «;» net-type-id ») phát sinh một trạng thái tức thời của mạng có tên net-var-id với kiểu được đặc tả trong net-type-id.

- Hủy bỏ một trạng thái tức thời của mạng. Chỉ thị « DELETE » (« net-var-id ») sẽ hủy bỏ trạng thái tức thời có tên net-var-id và trả lại phần bộ nhớ bị chiếm.

- Thử nghiệm ngẫu nhiên một mạng. Chỉ thị « RANDOMEXEC » (« net-var-id ») sẽ kích hoạt quá trình thử nghiệm ngẫu nhiên một trạng thái mạng tức thời net-var-id. Quá trình đó kết thúc khi tất cả các cái chuyển đã được chuẩn bị để cháy. Do chọn ngẫu nhiên, nên một cái chuyển có thể được chọn nhiều lần.

- Thử nghiệm một mạng có ưu tiên. Chỉ thị « PRIORITYEXEC » (« net-var-id ») sẽ kích hoạt quá trình thử nghiệm mạng trong đó các cái chuyển được chọn để cháy tùy thuộc vào độ ưu tiên tương đối của chúng.

2.2.3. Các chỉ thị cho phép xử lý và thử nghiệm các mạng theo chế độ hội thoại

- Chỉ thị

« STEPBYPSTEP » (« net-var-id »)

cho phép truy nhập đến trạng thái tức thời net-var-id và tạo ra những khả năng hội thoại sau đây:

- + Hiện các đánh dấu mạng và ma trận liên kết trên màn hình.
- + Liệt kê các vị trí và/hay hiện danh sách các cái chuyển của mạng lên màn hình.
- + Hiện những cái chuyển có thể cháy tại thời điểm đang xét lên màn hình.
- + truy nhập đến một vị trí hay một cái chuyển nào đó để thay đổi và / hoặc kiểm tra các thuộc tính của nó.

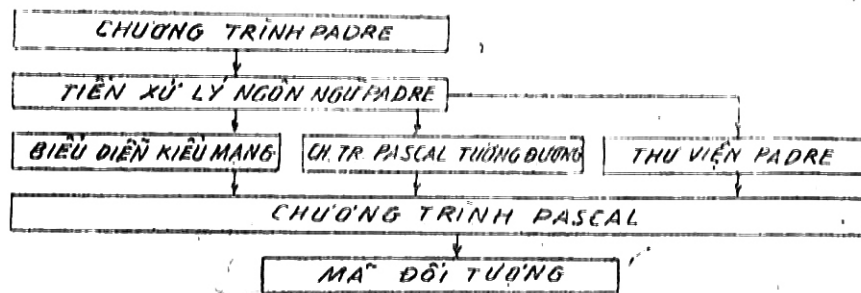
- Chỉ thị

« NETWORK » (« net-var-id1 «;» « net-var-id2 ») tạo ra những thao tác trên hai trạng thái tức thời được đặc tả nhờ tham số như sau:

- + hợp nhất hai vị trí hay hai cái chuyển,
- + hợp nhất những phần tử có tên giống nhau,
- + hợp nhất hai tập vị trí ([10]).

3. CÀI DẶT NGÔN NGỮ PADRE

Quá trình cài đặt có thể mô tả như sơ đồ sau



Ta có thể phân nhỏ quá trình cài đặt thành 2 bước, ở bước đầu mô tả quá trình dịch phần khai báo bằng ngôn ngữ PADRE và ở bước hai mô tả quá trình dịch phần lệnh.

3.1 Dịch phần khai báo

Dịch những kiểu dữ liệu chuẩn nguyên thủy như char, boolean, integer, miền con số nguyên không có khó khăn gì lớn lắm là do sự tương tự về cú pháp giữa Pascal và Padre.

Nhưng dịch kiểu dữ liệu xây mạng lại đòi hỏi phải phân tích sâu sắc những mô tả kiểu mạng để phát sinh động những biểu diễn trong của mạng. Để có thể sử dụng được những mạng lớn, tất cả những biểu diễn trong được tạo ra sẽ lưu trữ ở bộ nhớ ngoài.

Với mỗi mô tả kiểu mạng, bộ dịch sẽ tạo sinh biểu diễn trong tương ứng, một đồ thị ít nhiều phức tạp trong đó tất cả những thông tin thích hợp để sử dụng về sau chẳng hạn như các tiền điều kiện, guard, hậu điều kiện, các mặt tiếp giáp với bên ngoài, được thay đổi một cách phù hợp và lưu giữ lại nguyên vẹn.

Với mỗi thao tác bên trong mạng hay giữa các mạng với nhau, trong thư viện của ngôn ngữ PADRE có một thuật toán tương ứng đã được thử nghiệm tốt. Do vậy thư viện này phải gồm ít nhất một họ các thuật toán thao tác trên đồ thị như:

- Thuật toán duyệt đồ thị chẳng hạn để phát sinh những tiền điều kiện hay hậu điều kiện cho mỗi cái chuyển.
- Thuật toán sao chép đồ thị để phát sinh một thể hiện tức thời của mạng.
- Thuật toán hợp nhất đồ thị để thể hiện các phép kết hợp các mạng lại với nhau.
- Thuật toán thu dọn bộ nhớ khi có một trạng thái tức thời của một mạng nào đó bị hủy bỏ.

3.2 Dịch phần lệnh

Nói chung quá trình dịch phần lệnh không có gì khó khăn lắm, chính là do có sự tương đương trực tiếp về ngữ nghĩa giữa các cấu trúc điều khiển trong các ngôn ngữ Pascal và Padre. Có lẽ chỉ trừ các câu lệnh guarded-command và các chỉ thị thể hiện và xử lý mạng.

Dịch câu lệnh guarded-command

- Câu lệnh IF. Câu lệnh IF viết bằng ngôn ngữ PADRE

```
IF guard - 1 : statement - sequence 1 |
   guard - 2 : statement - sequence 2 |
   .....
   guard - k : statement - sequence - k |
FI
```

được dịch thành dãy các câu lệnh Pascal sau :

```
if T(guard-1) then begin T(statement-sequence-1)end ;
if T(guard-2) then begin T(statement-sequence-2)end .
.....
if T(guard-k) then begin T(statement - sequence -k)end ;
```

ở đây T(A) là kết quả dịch của A

- Câu lệnh LOOP

Câu lệnh LOOP trong ngôn ngữ PADRE

```
LOOP guard-1 : statement - sequence - 1 |
   .....
   guard - k : statement - sequence - k |
POOL
```

được dịch thành dãy các câu lệnh PASCAL sau :

```
j := 0 ;
repeat case j of
  0 : begin end ;
  1 : if T(guard-1) then begin T(statement-sequence-1) end ;
  2 : if T(guard-2) then begin T(statement-sequence-2) end ;
  k : if T(guard-k) then begin T(statement-sequence-k) end ;
end {case} ;
j := alea(generator, k) ;
```

until not (T(guard-1) or.. or T(guard-k)).

Trong sơ đồ dịch này, cơ chế không tiền định trong câu lệnh LOOP được thể hiện thông qua chọn ngẫu nhiên để thực hiện các câu lệnh guarded-command. Khi sử dụng kỹ thuật phân tích cú pháp xem trước 1 ký hiệu, do số các câu lệnh không biết được trước khi vào câu lệnh, nên đồ có thể áp dụng được nó trong quá trình dịch, chúng tôi đã cho vào một câu lệnh cảm:

O: *begin end* :

Dịch các *chỉ thị xử lý* và *thể hiện mạng*.

Quá trình dịch các chỉ thị về trạng thái tức thời và hủy bỏ các mạng rất dễ dàng vì mỗi chỉ thị tương ứng với một thuật toán trong thư viện, chẳng hạn :

- chỉ thị phát sinh trạng thái tức thời mạng sẽ kích hoạt thuật toán sao chép đồ thị,

- chỉ thị loại bỏ mạng sẽ kích hoạt thuật toán thu dọn bộ nhớ.

Mặt khác, quá trình dịch các cơ cấu thể hiện mạng khác nhau đòi hỏi 2 thủ tục cơ bản. Thủ tục đầu tiên thể hiện sự cháy của các chuyển t được mô tả bởi :

firing-a-transition-t ;

i) : { nếu t không có một vị trí in-place nào thì dãy câu lệnh Pascal sau đây sẽ được phát sinh :

« IF » [guard-of-t « AND »] precondition-of-t « THEN »

« BEGIN » [interface-activation « 3 »]

mark-consuming-sequence ; »

mark-adding-sequence » ; »

« END »

ii) : { nếu t có ít nhất một vị trí in-place được tô màu thì dãy câu lệnh sau đây được phát sinh trong đó các tham số x,..., w là màu của vị trí in-place của t.

Khi đó thuật toán thể hiện sẽ thực hiện tìm kiếm tất cả các khả năng để kiểm tra giá trị chân lý của điều kiện và guard của t }

xt := *alea* (generator, max-of-type-of-x) ;

x := *xt* ;

.

wt := *alea* (generator, max-of-type-of-w) ;

w := *wt* ;

repeat *x* := *x mod* max-of-type-of-x+1 ;

.

repeat *w* := *w mod* max-of-type-of-w+1 ;

if [guard-of-t « AND »] precondition-of-t

then *begin done* := true ;

[interface-activation ;]

mark-consuming-sequence ;

mark-adding-sequence ;

end

until (*w* = *w*) or *done* ;

until (*x* = *x*) or *done* ;

Thủ tục thứ hai thể hiện một danh sách các các chuyển mà p trở tới :

{ Thể hiện danh sách p }

While p < > nil do

begin t := transition-pointed-by-p ;

firing-a-transition-t ;

p := transition-following-p ;

end ;

Với hai thủ tục này, các chỉ thị khác dễ dàng dịch sang ngôn ngữ PASCAL, chẳng hạn

- Chỉ thị RANDOMEXEC được dịch thành

R := ;

repeat t := choice (T) ;

firing-of-t ;

until R = T ;

Ở đây T là tập các cái chuyển trong mạng, R là tập hiện thời những cái chuyển được chọn ngẫu nhiên, và chọn một Thuật toán chọn ngẫu nhiên nào đó

—Chỉ thị PRIORITYEXEC được dịch thành:

Phát sinh danh sách p các cái chuyển tùy thuộc vào độ ưu tiên của chúng;

Thờ hiện danh sách p;

Việc dịch các chỉ thị STEPBYSTEP và NETSWOK chỉ cần đến các cơ chế đã chỉ ra ở trên.

4. KẾT LUẬN

Hai mục đích được đưa ra ở trên khi cài đặt ngôn ngữ PADRE nhờ kỹ thuật tiền xử lý đã đạt được:

—Tốc độ thực hiện: Bộ tiền xử lý đã được viết xong và hiện nay đang hoạt động trên hệ thống MULTICS. Việc cài đặt nó đòi hỏi năng suất trong 12 tháng người với khoảng 10.000 dòng lệnh PASCAL.

—Khả năng trao chuyển: Việc chuyển hệ thống sang hệ máy VAX/UNIX đang được tiến hành, quá trình đó dường như không có một chút khó khăn gì và hệ thống sắp tới sẽ hoạt động.

Vấn bản chương trình hiện nay của bộ tiền xử lý có thể chia như sau:

—127 K bytes cho mã nguồn viết bằng PASCAL.

—138 K bytes cho thư viện của ngôn ngữ PADRE viết bằng PASCAL.

—232 K bytes cho mã đích sau khi liên kết.

Vì vậy kỹ thuật cài đặt này hoàn toàn phù hợp cho phần lớn máy vi tính hiện đang sử dụng.

Nhận ngày 15-2-1987

Tài liệu tham khảo

1. Van-Lu Nguyen, « PADRE: A Petri nets based parallel programming language » in Proceedings of the 4th Hungarian comp. Science conference, Győr, July 1985 Budapest. Eds: M. Arato, I. Katai, L. Varga.
2. Charles ANDRE, « Systèmes à évolutions parallèles: Modélisation par réseaux à capacités et Analyse par abstraction ». Thèse d'Etat, Université de Nice, Février 1981.
3. K. Jensen, « Coloured Petri nets and the invariant - method » in Theoretical Computer Science, N° 14, 1981.
4. H.J. Genrich, K. Lautenbach, « The analysis of distributed systems by means of predicate/Transition nets » in Semantics of concurrent computation, Lecture notes in Computer Sciences, N° 70.
5. N. Wirth, « Data structures + Algorithms = Programs », Prentice-Hall, 1976.
6. Van-Lu Nguyen, « PADRE: un langage d' experimentation de programmation parallèle et distribuée basé sur les Réseaux de Petri » Thèse d'Etat, Université Paris VI, à paraître 1986.
7. N. Wirth, « What can we do about the unnecessary diversity of notation for syntactic definitions », in CACM, November, 1977.
8. P.J. Coutois et. al. « Concurrent control with readers and writers », CACM, October, 1971.
9. E. W. Dijkstra, « Guarded commands, nondeterminacy and formal derivation of programs », CACM, August, 1975.
10. V.E. Kotov, « An algebra for parallelism based on Petri nets », in Lecture notes in Computer Sciences, N° 64, 1978.

ABSTRACT

ON THE PADRE'S IMPLEMENTATION BY PREPROCESSOR TECHNIQUE

Different steps and advantages of preprocessor technique are shown in this paper by way of a concrete implementation of Padre, a programming language, especially designed for describing, transforming and interpreting various classes of Petri nets. We also show that the technique is perfectly applicable on microcomputers.

Rus

THỰC HIỆN LIÊN LẠC CẤP THẤP TRONG XÂY DỰNG HỆ ĐA XỬ LÝ TỪ HAI MÁY TÍNH MINI16

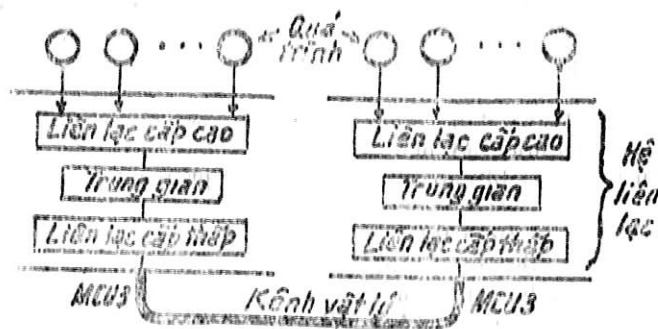
NGUYỄN NGỌC KIẾN
ĐOÀN THANH VỊNH
VŨ DUY MÃN

TÓM TẮT

Hoạt động liên lạc trong hệ đa xử lý từ hai máy tính MINI16 được thể hiện theo kiến trúc phân mức. Phần liên lạc cấp cao đảm bảo cho các quá trình phân tán trên hai máy tính có thể trao đổi, tương tác lẫn nhau: Phần liên lạc cấp thấp ứng với hai mức 1 và 2 trong mô hình liên lạc ISO--OSI, thực hiện liên hệ giữa hai máy tính trên kênh vật lý duy nhất và là cơ sở của các kênh logic. Bài này giới thiệu về vấn đề thiết kế và cài đặt các chương trình thể hiện phần liên lạc cấp thấp trên cơ sở liên kết hai máy tính qua giao diện nội bộ MCU3.

1. Yêu cầu và liên lạc trong xây dựng hệ đa xử lý.

MINI16 với hệ điều hành đa dạng MAS được thiết kế cho nhiều loại ứng dụng khác nhau [9]. Liên kết hai máy tính MINI16 sẵn có để tạo thành một hệ đa xử lý (multip-rocessing system - theo cách gọi của [5]) là bước đầu tiên trong việc thực hiện một liên kết nhiều máy tính. Yêu cầu đặt ra đối với hệ đa xử lý là tận dụng những khả năng sẵn có về phần cứng và phần mềm để tăng cường tiềm năng về xử lý tin phục vụ cho nhiều mục đích. Trước hết đó là tiềm năng về tốc độ (khả năng tính toán song song) và về tài nguyên thiết bị. Trên cơ sở thực hiện một liên kết vật lý giữa hai máy tính, công việc chủ yếu về phần mềm là xây dựng một hệ liên lạc (communication system) kết nối hai hệ điều hành MAS. Đối tượng phục vụ của liên hệ lạc là các quá trình tồn tại đồng thời và hoạt động tương tranh ở hai máy. Hệ liên lạc phải đảm bảo cho nhiều cặp quá trình phân tán có thể cùng thực hiện các cuộc đàm thoại trên những kênh logic độc lập (hình 1). Hệ cũng được thiết kế để trong đàm thoại các quá trình có thể trao đổi dữ liệu cũng như thực hiện tương tác lẫn nhau thông qua một số mệnh lệnh về đồng bộ hóa và hợp tác theo kiểu những id hợp nhiều máy tính phục vụ mô phỏng và điều khiển (ví dụ [8] hay DPS [3] v.v ..). Hệ liên lạc này được tổ chức theo kiến trúc phân mức (layered architecture) dựa trên những mô hình liên lạc phổ biến và được đơn giản hóa cho trường hợp một hệ nhỏ, gồm ba phần chính mô tả ở hình 1. Phần trên là liên lạc cấp cao đảm bảo các kênh logic và tương tác giữa các quá trình (ứng với ba mức trên trong mô hình quy chiếu liên kết các hệ mở của ISO). Phần dưới là liên lạc cấp thấp, ứng với mức 1 và 2 điều khiển truyền các tin báo qua kênh vật lý (là tài chung cho nhiều kênh logic). Hai phần này có những nhiệm vụ khác biệt nhau và được xây dựng một cách độc lập lẫn nhau. Phần trung gian có nhiệm vụ kết nối hai phần trên và dưới và là một cửa ngõ phục vụ những thay đổi và bổ sung nếu cần. Trong phạm vi hai máy tính thì những nhiệm vụ đặc trưng của mức 3 và 4 (ISO) cần thể hiện ở Phần trung gian là rất ít [1, 2, 3, 7]. Giao diện MCU3 [10] là một bộ điều khiển nhiều thiết bị, được dùng để tiếp nối giữa CPU MINI16 với các thiết bị vào-ra theo kiểu hai chiều luân phiên, di bộ, tuân tự theo byte với khoảng cách 500m (không cần mô-lem). Tuy không phải thiết bị truyền tin ra, nhưng MCU3 có cơ sở là chuẩn V24 của CCITT và sẵn có ở hai máy tính. Ghép nối hai MINI16 qua MCU3 là một giải pháp tận dụng khả năng sẵn có để tăng cường tiềm năng. Vì vậy hai máy tính đã được nối với nhau qua thiết bị này bằng cáp bọc kim dài 100m với tốc độ cao nhất của nó là 9600 baud theo kiểu MCU3 này coi MCU3 kia là ngoại vi. Đây là tốc độ chậm nhưng cũng có thể sử dụng cho những ứng dụng có mật độ trao đổi tin không cao.

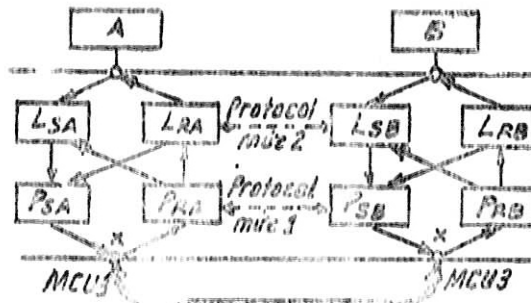


Hình 1

Bài này chưa đề cập đến liên lạc quá trình và tương tác ở những mức trên mà chỉ giới hạn trình bày về thực hiện liên lạc cấp thấp. Trong thực tế có nhiều mức độ thể hiện hai mức liên lạc 1 và 2 kể cả cứng hóa hoàn toàn. Song ở đây, về mặt kỹ thuật chỉ dựa trên một liên kết vào-ra bình thường qua hai giao diện nội bộ MCU3. Mọi yêu cầu cơ bản cho một kênh liên lạc điểm-điểm cũng như một số tính năng còn thiếu đối với một thiết bị liên lạc đều được giải quyết bằng phần mềm.

2. Mô hình về liên lạc cấp thấp.

Thiết kế một mô hình liên lạc là xác định các thực thể trao đổi cấu trúc liên hệ giữa chúng và các protocol tương ứng. Đối với những mức thấp đã tồn tại nhiều chuẩn và được phổ biến rộng. Vì vậy yêu cầu đặt ra là vận dụng một cách có chọn lựa các mô hình quen thuộc để giải quyết cho trường hợp cụ thể có những đặc thù nhất định. Để thuận tiện trong mô tả, cài đặt cũng như khi cần sửa đổi, ở đây đã thực hiện theo như SNA [4] là lập ra ở mỗi mức và mỗi bên những thực thể đơn thuần gửi hay nhận và xác định một cấu trúc như mô tả ở hình 2.



Hình 2

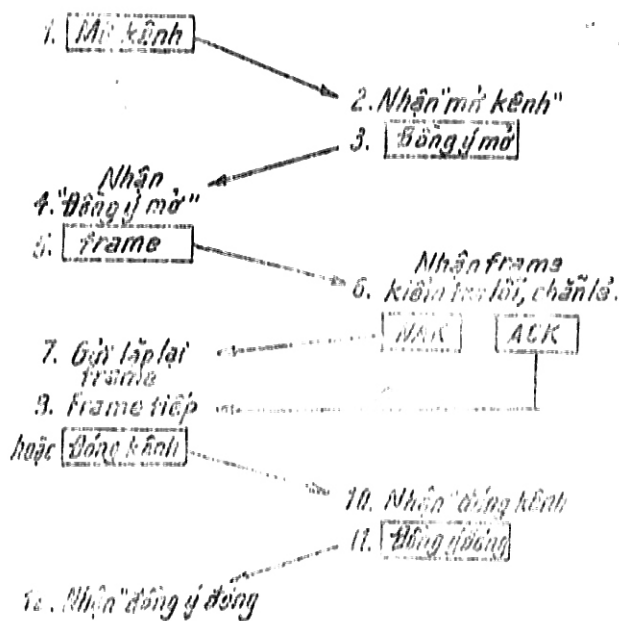
Nhu cầu trao đổi ở mỗi bên được qui tụ lại trong một đối tượng tổng quát là A và B và liên lạc cấp thấp phải đảm bảo chuyển các tin báo từ A sang B và ngược lại mỗi khi có yêu cầu. Công việc để đáp ứng một yêu cầu như vậy gọi là một hoạt động gửi - nhận. Các thực thể của mức 2 có nhiệm vụ phối hợp với nhau theo protocol mức 2 để thực hiện từng hoạt động gửi - nhận. Khi thực hiện chúng cần đối thoại, trao đổi cho nhau dữ liệu và các tín hiệu điều khiển thông qua các thao tác gửi - nhận qua kênh. Các thực thể của mức 1 có nhiệm vụ liên hệ trực tiếp với thiết bị và phối hợp với nhau theo protocol mức 1 để thực hiện các thao tác gửi - nhận đó mỗi khi mức 2 yêu cầu.

Các thực thể mức 2 được ký hiệu là L (link) và mức 1 là P (physical) kèm theo S hay R ứng với gửi (send) hay nhận (receiv). Cặp LSA-LRB phối hợp với nhau theo protocol mức 2 để thực hiện hoạt động gửi - nhận từ A sang B (tương tự cặp LSP-LRA thì từ B sang A). Ở bên A, PSA tiếp nhận các yêu cầu gửi (tín hiệu và dữ liệu) của cả LSA và LRA, rồi phối hợp với PRB để thực hiện dưới dạng thao tác gửi - nhận từ A sang B. PRB sau khi nhận thì chuyển lên cho LSB hay LRB tùy theo đối tượng cần nhận. Chiều ngược lại cũng tương tự như vậy.

Protocol mức 1 được coi là theo V24 trên cơ sở các qui định riêng của MCU3, trong cái đặt được thể hiện bằng các thủ tục tạo thành từ các lệnh vào—ra như: khởi động, dừng bộ điều khiển, kiểm tra và chuyển trạng thái thiết bị, gửi và nhận byte.

Đối với một liên kết điểm—điểm như ở đây thì quan trọng và phức tạp hơn là xây dựng Protocol mức 2. Protocol mức này phụ thuộc những đặc trưng chung cơ bản của thiết bị chứ không phụ thuộc cụ thể thiết bị nào. Trong số nhiều protocol phổ biến ở đây đã chọn BSC. Đó là một protocol dùng cho truyền tin theo byte, tuần tự, hai chiều luân phiên và đơn giản hơn các protocol khác. Tuy được thiết kế cho truyền tin đồng bộ nhưng nó hoàn toàn có thể dùng cho dị bộ và cả truyền song song [1, 3], vì vậy nó phù hợp với thiết bị MCU3. Khi áp dụng cho trường hợp dị bộ thì không cần đến vai trò trực tiếp của các tín hiệu đồng bộ. Ở đây chỉ dùng tín hiệu này khi mở kênh. Vì vậy có thể đơn giản hóa chúng trong một số trường hợp. Hơn nữa cũng cần giải quyết không chỉ một hoạt động gửi—nhận đơn lẻ mà là các chuỗi những hoạt động như vậy. Do đó cách làm ở đây là dựa theo luật trao đổi cơ bản của BSC với một tổ chức tin có biến đổi.

Chế độ chung là mỗi hoạt động gửi—nhận được thực hiện một cách trọn vẹn và loại trừ lẫn nhau. Các yêu cầu ở phía trên có thể tạo thành hàng đợi, song đối với Liên lạc cấp thấp thì nhiều nhất chỉ một yêu cầu phải đợi nếu LR ở đó đang nhận tin báo và sẽ được xử lý ngay sau khi nhận xong tin báo đó. Còn xét riêng thì mỗi hoạt động gửi—nhận được thực hiện làm ba pha như mô tả ở hình 3. Đây là sơ đồ trao đổi của BSC được mô tả trong [3].



Hình 3

Giả sử A cần gửi cho B một tin báo. Hoạt động của cặp LSA—LRB là:

— Pha mở kênh: LSA kiểm tra trạng thái kênh, nếu kênh tự do thì gửi cho LRB tín hiệu mở kênh. Kênh được coi là mở khi LSA nhận được từ LRB tín hiệu khẳng định về mở kênh. Ở đây cần phải giải quyết trường hợp có đùng độ khi trong một khoảng cách thời gian quá nhỏ cả LSA và LRB cùng thấy kênh tự do và cùng gửi tín hiệu mở kênh.

— Pha trao đổi: Khi kênh đã mở, LSA gửi tin báo sang cho LRB. Tin báo có độ dài không cố định được chia thành các đoạn có độ dài cố định (trừ đoạn cuối) và được đối thoại để gửi và nhận từng đoạn một có kiểm tra lỗi và gửi lặp một số lần khi có lỗi. Mỗi đoạn tin được tổ chức theo một khuôn dạng nhất định gọi là frame có các tín hiệu đầu, cuối, độ dài, mã phát hiện lỗi. LRB nhận các frame, kiểm tra, loại bỏ các tín hiệu và tạo lại tin báo từ những đoạn rời rạc, gửi cho LSA các tín hiệu khẳng định cần thiết.

- Pha đóng kênh: Được thực hiện giống như khi mở kênh. Nếu hình thành dây đợi thì khi đóng kênh sẽ có tín hiệu gửi kèm để đảo lại vai trò gửi, nhận nhằm đáp ứng luân phiên các yêu cầu đang đợi.

3. Vài nét về cài đặt.

Điềm qua những nét đặc trưng về cài đặt như sau:

Theo sơ đồ ở hình 2, mỗi thực thể Ls, Lr, Ps, Pr được thể hiện bằng một modul với các liên hệ như đã xác định trên hình. Ngoài ra tại điềm X trên hình 2 còn có một modul nữa để xử lý ngắt gắn với chế độ làm việc của MCU3 và CPU MINI16. Phần Trung gian ở phía trên đặt yêu cầu, nhận kết quả cũng như xếp yêu cầu vào hàng đợi đối với Liên lạc cấp thấp tại điềm chuyển tiếp giữa hai phần. Các modul sử dụng một tập các biến trạng thái chung truy nhập theo kiểu loại trừ lẫn nhau. Hoạt động của những modul này cũng như diễn biến về chuyển trạng thái là theo như các mô tả đã nêu. Độ dài một tin báo do các mức trên quy định cực đại là 512 byte, còn độ dài từng đoạn để lập frame là 80 byte. Các kích thước này được chọn vì 512 byte và 80 byte là các chỉ số nằm trong phạm vi thể hiện của nhiều hệ, đối với MAS thì 512 byte là đủ bao một sector đĩa từ còn 80 byte thì đủ chứa một mệnh lệnh tương tác giữa các quá trình ở mức trên (mỗi mệnh lệnh đều được chuyển hết ngay trong một frame).

Cơ chế ngắt cho phép tận dụng một khoảng thời gian xen kẽ đáng kể để dùng vào các công việc khác song song với liên lạc. Khi hai MCU3 làm việc với nhau để gửi, nhận từng byte thì CPU được rảnh rỗi và có thể dùng vào hoạt động khác. Khi MCU3 chuyển sang làm việc với CPU thì tín hiệu ngắt sẽ kéo CPU về chế độ liên lạc. Số liệu thực hiện cho thấy khi tiến hành một hoạt động liên lạc cấp thấp thì chỉ có 25% thời gian là thực sự cho liên lạc, 75% còn lại cho phép phục vụ khá nhiều đối với những hoạt động đồng thời khác. Tốc độ chậm được đền bù lại một phần nhờ khả năng này.

Cơ chế ngắt còn được dùng để xử lý các trường hợp bất thường như mất nguồn, tắt bật bất ngờ, v.v... trong đó quan trọng nhất là giải quyết đụng độ. Đụng độ có thể xảy ra ở byte đầu tiên của tín hiệu mở kênh ở cả hai bên. Tại đây đã giải quyết bằng cách, về mặt thiết bị, đặt một MCU3 ở chế độ «nhảy cảm» và MCU3 kia ở chế độ «không nhảy cảm» với đụng độ. Chỉ bên «nhảy cảm» khi đụng độ thì bị dừng và Ls của nó sẽ xếp yêu cầu vào hàng đợi rồi chuyển sang khởi động Lr để nhận. Sự chuyển đổi này được thực hiện trong khoảng thời gian có kéo dài giữa hai byte «đồng bộ» đầu tiên của tín hiệu mở kênh. Như vậy nếu đụng độ thì một bên luôn được ưu tiên, còn một bên thì nhờ vào những tín hiệu chuyển tiếp khi đóng kênh cũng chỉ phải đợi sau một tin báo. Ngoài ra có thể thấy rằng sự nhảy cảm đụng độ ở đây là nhảy cảm trên trạng thái ghi nhận lại hiệu tượng «đang ở chế độ gửi mà vẫn có tín hiệu từ bên kia tới». Sự đụng độ bao hàm cả trạng thái như vậy giúp cho việc mô phỏng hiện tượng đụng độ khi cài đặt chương trình được thuận tiện thông qua thử nghiệm ở chế độ thực hiện từng lệnh một theo yêu cầu ở cả hai bên tại thời điềm mở kênh.

Một số tính năng cần thiết cho liên lạc không có trong MCU3 (vì là giao liên nội bộ) được khắc phục bằng những chương trình-thủ tục nhỏ. Hai byte CRC (cyclic redundancy code) được lập bằng một thủ tục tìm dư khi chia đa thức dư hiệu (frame) cho đa thức sinh là $x^{16} + x^{12} + x^5 + 1$ của CCITT [6, 7]. Ta biết rằng CRC theo đa thức sinh như vậy cho phép giảm lỗi từ 10^3 đến 10^5 lần cho 2000 đến 10.000 bit [7]. Tuy vậy việc lập và kiểm CRC bằng phần mềm cũng chiếm tới 15% so với thời gian liên lạc cấp thấp. Để phân biệt các ký hiệu đặc biệt, một thủ tục khác cũng được lập để tạo ra và loại bỏ các ký hiệu chèn (byte-stuffing). Để nhận biết time-out đã thực hiện mở rộng thủ tục xử lý ngắt thời gian bằng cách đưa vào một thủ tục nhỏ và một con đếm thời gian dành riêng cho liên lạc. Khi xử lý tín hiệu ngắt của đồng hồ thời gian thực (real-time clock) con đếm này được tích lũy và khảo sát để xác định time-out. Ngoài ra, để thực hiện «loại trừ lẫn nhau» tại những công đoạn cần thiết, đã sử dụng cặp lệnh «cấm» và «cho phép ngắt» để cấm tất cả các hoạt động xen kẽ.

Toàn bộ hệ liên lạc kể cả ba phần đã được cài vào hạt nhân hệ điều hành MAS như một bộ phận của chức năng vào-ra, cũng như những can thiệp hệ thống khác có liên quan, bằng giải pháp chung đối với phát triển MAS thông qua khảo sát, sửa đổi các chương trình nguồn khởi phục được [11] và bổ sung xen kẽ vào qui trình sinh hệ sẵn có [12]. Riêng phần liên lạc cấp thấp được thể hiện như một chương trình điều khiển thiết bị (driver) đối với MCU3, sử dụng được tất cả các khả năng phụ sửa cơ chế vào-ra. Phần liên lạc cấp thấp (cũng như toàn bộ hệ liên lạc) được viết trên hợp ngữ (assembler) của MINI16, chiếm một

vùng trong bộ nhớ là 2,5 Kbyte. Thời gian để truyền một frame mất 160ms đạt tỉ lệ 60% so với tốc độ vật lý cao nhất của MCU3-9600 baud. Thời gian để lập và kiểm tra CRC bằng phần mềm là tốn kém, song so với tốc độ chậm của thiết bị thì có thể chấp nhận được vì nếu bớt đi 15% thời gian dành cho CRC thì coi như MCU3 làm việc với tốc độ khoảng 7200 baud - là khả năng thứ hai trong nhiều tốc độ mà MCU3 cho phép. Ngoài ra cũng đã xây dựng một thủ tục kiểm tra đơn giản hơn và nhanh hơn, đó là LRC (longitudinal RC). Trường hợp có nhu cầu phải giảm thời gian thì có thể rút bớt khoảng cách và áp dụng LRC. (Tất nhiên khả năng phát hiện lỗi bằng LRC thấp hơn nhiều so với CRC).

Cuối cùng tác giả xin chân thành cảm ơn PTS Đỗ Minh Thái cũng như Xemina « Cơ sở dữ liệu và lập trình » của Viện KHTT và ĐK về những ý kiến đóng góp cho nội dung trên.

Nhận ngày 21-3-1987

TÀI LIỆU THAM KHẢO

1. ЯКУБАЙТИС Э. А., Архитектура выч. сетей. Москва, статистика, 1980.
2. ЯКУБАЙТИС Э. А., Локальные ИНФ. Выч. сети. Рига, Зинатне, 1985.
3. Woltzman C., Distributed Micro/Mini Computer Systems, Prentice - Hall 1980.
4. Cypser R., Communication Architecture for Distributed System, Addison Wesley 1978.
5. Yourdon E., Design of on-line computer System, Pr. Hall 1972.
6. Knuth D., The art of computer programming, Add. Wesley, 1969.
7. Guy Fajolle, La télématique: Réseaux et Appl. Eyrolles, Paris 1982.
8. Garotti P., Laface P., Rivoira S., On the development of a Distr. OS kernel for real-time appl. Annu. Rev. Autom. Progr. 1983, 11 (abstract).
9. MIN16 - Multi Application System, Release 4, 1976.
10. MIN16. - MCU3 Service Manual, 1976.
11. N. N. Kiên., Về tương đương giữa các chương trình nguồn và vấn đề khôi phục dạng nguồn. Tạp chí khoa học tính toán và điều khiển, Tập 2, số 1, 1986.
12. N. N. Kiên, L. N. Đứ, Đ. M. Thái., Phát triển các chức năng chuyên dụng cho một hệ điều hành thời gian thực nhiều mục đích. Thông báo khoa học Hội nghị Toán học 1985.

РЕЗИОМЕ

РЕАЛИЗАЦИЯ НИЗКО - УРОВНЕВОЙ КОММУНИКАЦИИ В КОНСТРУИРОВАНИИ
МУЛЬТИ-ОБРАБАТЫВАЮЩЕЙ СИСТЕМЫ ИЗ ДВУХ ЭВМ МИНИ 16

Коммуникация в мульти-обрабатывающей системе из двух ЭВМ МИНИ 16 реализуется по многоуровневой архитектуре. Высшая часть коммуникации обеспечивает распределённым процессам возможность взаимодействовать друг с другом. Низкая часть, соответствующая двум уровням 1 и 2 эталонной модели МОС выполняет связь между двумя ЭВМ. На единственном канале и является основой логических каналов. Эта работа посвящена проблеме конструирования программ реализующих низкую часть коммуникации на основе соединения этих машин через внутренние интерфейсы MSU 3.

Rus

THÔNG BÁO KẾT QUẢ NGHIÊN CỨU ỨNG DỤNG

HỆ TV - 86

NGUYỄN NHƯ CẬN

Máy tính điện tử A6102 (Cộng hòa Dân chủ Đức) là một máy mini mới được trang bị ở Việt Nam, chưa có hệ cơ sở dữ liệu. Vì vậy khi tổ chức khai thác thông tin, đặc biệt trong các trường hợp phải đáp ứng một số yêu cầu đột xuất, đã gặp phải những hạn chế nhất định. Hệ chương trình dùng chung TV-86 được xây dựng nhằm khắc phục một phần những hạn chế đó.

Hệ TV-86 được viết bằng ngôn ngữ Cobol bao gồm các chương trình: nạp dữ liệu, cập nhật dữ liệu, in dữ liệu, tìm kiếm dữ liệu và nhập các tập dữ liệu.

Sau đây xin trình bày tóm tắt tính năng tác dụng của từng chương trình:

I - CHƯƠNG TRÌNH NẠP DỮ LIỆU

1. Dùng để đưa dữ liệu vào máy qua máy đọc băng giấy hoặc bàn phím, đồng thời tiến hành kiểm tra, thông báo từng lỗi và tổng số lỗi trong quá trình đưa vào.

2. Một số tính năng phục vụ cho việc nạp dữ liệu:

- Với mỗi bộ, số liệu được nạp theo độ dài thay đổi.

- Có thể sao lại giá trị một số thuộc tính hoặc cả bộ tự động gán Φ hoặc blank cho giá trị của một số thuộc tính

- Tự động xóa 1 phần của bộ hoặc cả bộ.

- Nạp số liệu sửa được thực hiện qua những phím chức năng đơn giản.

II - CHƯƠNG TRÌNH CẬP NHẬT DỮ LIỆU

1. Dùng để hủy, sửa, thêm dữ liệu, đồng thời tiến hành kiểm tra và thông báo lỗi trong quá trình cập nhật.

2. Có thể tiến hành cập nhật theo số thứ tự của bản ghi trong tệp hoặc theo khóa với:

- Vị trí của khóa trong bản ghi và độ dài của mỗi khóa là tùy ý.

- Số lượng khóa tối đa là 15

III - CHƯƠNG TRÌNH IN DỮ LIỆU

1. Dùng để in dữ liệu qua máy in hoặc màn hình.

2. Có thể in toàn bộ bản ghi hoặc một số trường (lấy theo trật tự tùy ý) của một bản ghi.

- Với một trường, có thể in theo 10 kiểu.

- Trên 1 dòng có thể in 1 hoặc 2 bản ghi (có hoặc không có số thứ tự).

- Có thể thực hiện việc cộng trong cả tệp theo 1 đến 20 trường và in tổng số ra trên 1 hoặc 2 dòng.

IV - CHƯƠNG TRÌNH TÌM KIẾM DỮ LIỆU

1. Dùng để tìm kiếm và tách 1 tệp dữ liệu thành 2 tệp gồm các bản ghi thỏa mãn và không thỏa mãn một số các điều kiện được biểu diễn dưới dạng một biểu thức logic.

2. Biểu thức logic bao gồm các chữ cái (biểu diễn điều kiện đơn), dấu các phép toán logic \vee , \wedge , \neg và các dấu ngoặc đơn mở và đóng.

- Về trái của điều kiện đơn có thể là tên trường hoặc vị trí đầu cuối của trường trong bản ghi, về phải của điều kiện đơn có thể là 1 hằng, 1 trường hoặc 1 biểu thức số học đơn giản (với các phép toán +, -, ×, /) trong đó toán hạng thứ nhất là 1 trường và toán hạng thứ hai là 1 hằng số.

- Dấu của các phép toán so sánh gồm có: =, ≠, <, >, ≤, ≥, [],], (,), {},

- Có thể tìm min, max trong cả tệp hoặc trong 1 tệp con (tệp gồm các bản ghi thỏa mãn một số điều kiện).

- Có thể tách theo số thứ tự của bản ghi trong tệp.

- Có thể tách thành các tệp dùng để in theo nhiều kiểu khác nhau (2 trang trước sau, 4 trang trước sau, in trang trước sau theo khóa. ...).

- Có thể kiểm tra tính duy nhất của giá trị một trường đối với từng bản ghi trong cả tệp và thông báo sai.

3. Dạng tổ chức của các bản ghi thỏa mãn biểu thức và các bản ghi không thỏa mãn biểu thức hoàn toàn độc lập đối với nhau.

Sơ với dạng tổ chức của bản ghi trong tệp gốc, có thể tổ chức dạng của các bản ghi được tách ra như sau :

- Giữ nguyên.

- Rút ngắn hoặc kéo dài độ dài bản ghi.

- Lấy ra một số trường trong bản ghi.

- Chuyển đổi vị trí của các trường trong bản ghi.

- Thay thế giá trị của các trường.

- Chèn thêm các giá trị mới vào bản ghi.

4. Ghi chú: Sau khi nhận biểu thức logic và điều kiện đơn, có tiến hành kiểm tra và thông báo lỗi.

V - CHƯƠNG TRÌNH TRỘN CÁC TỆP DỮ LIỆU

1. Dùng để trộn 2 tệp dữ liệu đã được sắp xếp theo chiều tăng hoặc giảm của các khóa thành 1 tệp được sắp xếp theo chiều tương ứng của các khóa.

2. Vị trí của khóa trong bản ghi và độ dài của mỗi khóa là tùy ý.

Số lượng khóa tối đa là 15.

Trước khi thực hiện các chương trình dùng chung trên đây, người sử dụng phải khai báo các tham số (tên, kiểu, độ dài của các trường, kiểu in ra, khóa, các trường cần cộng,...) trong một tệp gọi là tệp tham số.

Nhận ngày 10-7-1987.

РЕЗЮМЕ

СИСТЕМА ТУ-86

В работе описываются основные характеристики системы ТУ-86. Система означена для обеспечения данных в бухгалтерских расчетах. Она выполняет следующие функции над файлами данных :

- вывод данных в файлы,

- вывод данных по различным формам,

- поиск данных в файлах по достаточно сложным логическим выражениям,

- слияние двух файлов.

Система написана на языке программирования коболе и работает на мини ЭВМ А6102.

PS/2 VÀ NHỮNG TIÊU CHUẨN TƯƠNG LAI

NGUYỄN CHÍ CÔNG

Năm 1987 có thể sẽ ghi nhận một sự kiện đáng nhớ trong lịch sử vi tính học giống như những năm 1976 và 1981 đã đánh dấu sự ra đời của máy tính cá nhân Apple II rồi IBM PC. Không ai ngờ nhiên khi biết nguồn gốc của những sản phẩm gây chấn động lần này vẫn chính là những tên tuổi quan trọng nhất đã từng đóng vai trò quyết định trong quá khứ nói trên.

Trước hết, ngay từ tháng ba năm 1987 hãng Apple Computer Inc. đã đưa ra MAC II, chiếc máy vi tính cá nhân 32 bit đầu tiên kế thừa LISA và MAC I, MAC PLUS, MAC SE về phần mềm nhưng lại có phần cứng mở cửa qua NU BUS. Tuy nhiên bước ngoặt thực sự có lẽ là vào tháng tư, khi hãng IBM Corp thách thức những kẻ ăn theo mình (clones) bằng cả một họ gồm tới 8 máy tính cá nhân gọi là PS/2, kèm theo cùng một hệ điều hành đa nhiệm mang tên OS/2 vẫn từ Microsoft mà ra.

Trong khi Apple tiếp tục chọn họ vi xử lý MC 68000 thì IBM vẫn tin nhiệm Intel 8086/80286/80386. IBM cũng nghiêng hẳn về chuẩn đĩa từ 3,5 inch như Apple đã làm từ 1982 và có thể sẽ dẫn đến việc chuẩn này trở nên phổ biến hơn đĩa từ 5,25 inch trong vài năm tới. PS/2 cũng đi theo quan điểm "thân thiện với người" của họ MAC bằng cách gài con chuột và màn hình đồ họa vừa nét hơn vừa có nhiều cửa sổ linh hoạt dưới sự quản lý của Windows Presentation Manager (Microsoft). MAC II tỏ ra ưu việt ở mặt âm thanh do sử dụng các bộ đồng xử lý của hãng Sony, nhưng nổi bật nhất có lẽ là việc từ bỏ kiến trúc khép kín của MAC I để trở về truyền thống mở cửa qua ổ cắm theo chuẩn NUBUS của Texas Instruments (đã phổ biến trong nhiều máy "trí tuệ nhân tạo") và qua ghép nối các thiết bị ngoại vi theo chuẩn quốc tế SCSI.

Ngược lại PS/2 hầu như đoạn tuyệt với quá khứ PC/XT/AT bằng một BUS mới (MCA = Micro-Channel Architecture) và một hệ điều hành mới, không tương hợp MS-DOS. Hệ OS/2 này khai thác được các ưu điểm của bộ vi xử lý 80286 mà AT chưa đạt được, nhưng những chương trình ứng dụng với OS/2 còn phải đợi thời gian kiểm nghiệm, trong khi MAC II lại thừa hưởng phần mềm có sẵn từ 5 năm nay.

Trên bìa mẹ PS/2 có hầu hết mọi linh kiện điện tử, kể cả phần điều khiển các thiết bị ngoại vi thông thường; tuy vẫn có 7 ổ cắm nhưng BUS MCA và kích thước hộp máy không cho phép chứa các bìa mở rộng của họ PC/XT/AT. Một số vi mạch riêng của IBM cũng làm cho việc bắt chước PS/2 trở nên rất khó khăn.

Tuy vậy PS/2 là một họ máy cá nhân rất mạnh. Công nghệ hàn linh kiện trên bề mặt SMD (Surface Mounted Devices) làm cho mật độ mạch in và dữ liệu tăng gấp bội (MTBF = 50 năm!!). Với MCA việc cải tiến quản lý các kênh DMA đã đạt tốc độ truyền dữ liệu qua đĩa cứng nhanh lên bốn lần so với máy AT. Các bộ điều khiển màn hình kiểu mới như MCGA và VGA vừa tương hợp với các chuẩn cũ (MDA, CGA và EGA) lại vừa biểu thị nhanh gấp đôi. Song tốc độ tính toán nói chung còn thấp hơn các máy 32 bit khác và OS/2 thực ra chưa khai thác được sức mạnh của 80386. IBM đã thôi sản xuất họ PC/XT và trong thời gian chuyển tiếp hiện nay chỉ còn dùng AT, riêng chiếc PS/2 đầu tiên là 8530 vẫn chạy với DOS 3.3 để tương hợp MS-DOS và những chiếc mạnh hơn mới dứt hẳn để theo OS/2. Một bất ngờ nhỏ là hệ điều hành đa nhiệm Prologue 286 (ra đời từ 1984) lại chạy được trên chiếc PS/2 - 8560/

Trong các ngoại vi mạnh của PS/2 có thể lưu ý những ổ đĩa cứng tốc độ cao theo chuẩn ESDI và đặc biệt là bộ đọc đĩa quang số. Cùng với màn hình đồ họa cao cấp những máy mạnh trong họ PS/2 và MAC II tỏ ra thích hợp trong CAD hai chiều, nhất là do giá rẻ hơn những Work Station chuyên nghiệp kiểu Apollo DN 3000, DN 4000 hoặc DEC Vaxstation 2000 v.v... Tất nhiên PS/2 được thiết kế chủ yếu cho tự động hóa văn phòng; khả năng ghép kính tế giữa chúng hiện thực hơn so với họ PC/XT/AT, chỉ ít ta cũng có thể nối một cách đơn giản qua cửa song song hai chiều có sẵn.

Chắc rằng PS/1 và OS/2 dễ trở thành những tiêu chuẩn tương lai trong vi tin học, không phải do tính cách mạng trong kiến trúc mà trước hết là vì sức mạnh tài chính của IBM. Không đầy hai tháng sau đã xuất hiện họ máy OGIVAR dùng 80386 và tương hợp PS/2. Nhưng họ PC/XT/AT cũng còn phổ biến vài năm nữa do giá thành cực thấp của những sản phẩm bắt chước ở Đông Nam Á (PC \approx 400 đô la, XT \approx 1000 đô la, AT \approx 1500 đô la) và khả năng ứng dụng rộng rãi của những phần mềm có sẵn chạy dưới MS-DOS. Ngay những hãng lớn như Olivetti, Compaq và Wyse vẫn tiếp tục dùng MS-DOS và BUS AT như cũ trong các máy 16/32 bit vừa ra đời năm nay hoặc cuối năm ngoài, mặc dù họ tuyên bố sẽ chấp nhận OS/2. Ta cũng không loại trừ khả năng có những máy tương lai bao gồm cả BUS AT và BUS MCA.

Bảng đặc trưng kỹ thuật của MAC II và PS/2

	CPU : MC 68020/16 MHz, Đồng xử lý số học MC 68881
M	RAM : 1 MBytes, có thể mở rộng lên 8 MBytes
A	ROM : 256 KBytes Giá bán lẻ : từ 55.000F trở đi.
C	Liên lạc : 2 cửa ghép nối tiếp theo chuẩn RS 422 (CCITT V 11)
II	BUS : NUBUS, 6 ổ cắm châu Âu 96 tiếp điểm
A	Ghép ngoại vi : Chuẩn SCSI : 7 thiết bị, chuẩn FDB : 16 thiết bị
P	Ổ đĩa mềm : 2 ổ 3,5 inch, đĩa 2 mặt mật độ kép sức chứa 800 KB
P	Ổ đĩa cứng : 1 ổ bên trong máy sức chứa 20, 40 hoặc 80 MB
P	Màn hình : đồ họa 640x480 pixel, 256 mức sáng hoặc màu sắc
L	Bàn phím : tách rời như màn hình, 104 phím bấm, phím số riêng
E	Hệ điều hành : Finder 5.2, thực đơn di động, cửa sổ
	Mạng cục bộ : Apple Talk và EtherNet
	Phụ kiện : con chuột, máy in laser, băng từ, 4 đầu ra loa

IBM - P8/2

Kiểu máy	CPU/	RAM	BUS	DMA	Điều khiển màn đồ họa	Đĩa cứng	Đĩa mềm	Đĩa cứng	Hệ điều hành	Giá bán lẻ (trừ bàn phím - màn hình)
8530-002	8086/8 MHz	640 K	AT 16 bits	2 kênh	MCGA: đơn sắc 640×480, hoặc 320×200/256 màu khác nhau	3	2× 720K		DOS 3.3	10693F 4-1987
8530-021							1× 720K	1× 20M		15073F 4-1987
8550	80286/10 MHz	1-7M	16 bits MCA	15 kênh trong đó có 8 kênh đồng hành	VGA: 640×480/16 màu, hoặc 320×200/256 màu, hoặc EGA	3		1× 20M	DOS 3.3 + OS/2	25709F 4-1987
8560-041		1-15 M					1× 44M	37507F 5-1987		
8560-071							1,44 M	1× 70M ESDI		41354F 7-1987
8580-041	80386/16 MHz	1-20 M	16/32 bits MCA	1024×768/256 màu (bìa đặc biệt phụ thêm ngoài VGA có sẵn trên bìa mẹ). Giá bìa 19713 F	7, trong đó có 3 đĩa 32 bits	1× 1,44 M	1× 44M	DOS 3.3 + OS/2 + AIX (1988)	45734F 9-1987	
8580-071		2-20 M					1× 70M ESDI		53865F 9-1987	
8580-111							1× 115M ESDI		68264F 12-1987	

TÀI LIỆU THAM KHẢO

1. La famille Macintosh, Apple Computer France, 2-3-1987.
2. Minis et Micros: N° 270, 1-5-1987; N°282, 15-6-1987.
3. Décision Informatique: N°147, 20-6-1987; N° 144, 8-6-1987.
4. Inf. PC: N°31, 5-1987.
5. Electronique Hebdo: N°31, 5-3-1987; N°22, 12-3-87.

TIN TỨC HOẠT ĐỘNG

Hội thảo: CÔNG NGHỆ PHẦN MỀM - 87

Được sự chỉ đạo của Ban chủ nhiệm chương trình tin học quốc gia, sự ủng hộ và tài trợ của nhiều cơ sở nghiên cứu và ứng dụng tin học - đặc biệt của CP-84, Viện Khoa học tính toán và điều khiển đã phối hợp với Trung tâm Máy tính Bộ Quốc phòng, một số trường Đại học và cơ quan nghiên cứu, tổ chức Hội thảo « Công nghệ phần mềm - 87 » tại Bãi Cháy từ 24 đến 28-8-1987. Hơn 40 đại biểu đã nghe, thảo luận và trao đổi về các báo cáo và các vấn đề sau:

- Tổng quan và công nghệ phần mềm (CNPM);
- R - Công nghệ và những vấn đề liên quan;

- Các công cụ đại số và logic của CNPM;
- CNPM và Trí tuệ nhân tạo;
- Những quan niệm và nội dung cơ bản của CNPM trong khối SEV;
- Suy nghĩ về CNPM và một số kiến nghị về việc tổ chức triển khai ở Việt Nam.

Hầu hết các đại biểu dự Hội thảo đều nhất trí nhận định rằng CNPM là một lĩnh vực có ý nghĩa chi phối và bao trùm toàn bộ các hoạt động nghiên cứu và ứng dụng của tin học, cần có kế hoạch, biện pháp đầu tư và triển khai thích đáng.

H.T.

GIỚI THIỆU SÁCH

NGUYỄN XUÂN HUY, LẬP TRÌNH SONG NGỮ PASCAL-BASIC Hà Nội, NXB Thống kê, 1978, 126 trang

Sự phát triển mạnh mẽ của tin học và kỹ thuật tính toán đòi hỏi ngày càng nhiều người biết lập trình cho máy tính. Ở nước ta, trên các máy vi tính, đang được sử dụng và khai thác phổ biến hơn cả là các ngôn ngữ lập trình PASCAL và BASIC. Sách « Lập trình song ngữ PASCAL-BASIC » ra đời trong hoàn cảnh ở ta còn thiếu tài liệu tiếng Việt dạy học lập trình, mà số người mong muốn sử dụng máy vi tính ngày càng tăng. Nội dung cơ bản của cuốn sách là:

Giới thiệu những thao tác cơ bản biến đổi một đoạn chương trình viết trên PASCAL sang đoạn chương trình BASIC tương đương.

- Giới thiệu kỹ thuật lập trình theo phương pháp tính chế từng bước xuất phát từ phân tích bài toán cho đến khâu mã hóa trên ngôn ngữ cụ thể.

- Biện diễn các thuật toán cơ bản như tìm kiếm trong mảng, tìm phần tử lớn nhất, tìm ước số chung lớn nhất, sắp xếp trong (8 phương pháp khác nhau), sắp xếp ngoài ở dạng có thể thực hiện được ngay trên máy tính.

Người dùng mới biết một trong hai ngôn ngữ kể trên, khi đọc cuốn sách này có thể dễ dàng biết thêm một ngôn ngữ mới nhằm mở rộng tầm hiểu biết và khai thác tối hơn tài nguyên tính toán sẵn có. Tác giả đã khéo vận dụng phương pháp so sánh để trình bày những khác biệt cơ bản của hai ngôn ngữ PASCAL và BASIC. Ngữ nghĩa của ngôn ngữ này được biểu diễn qua ngữ nghĩa của ngôn ngữ kia và chính việc trình bày trở nên rõ ràng, chặt chẽ và đơn giản hơn so với phương pháp « Đơn » ngữ đang được dùng phổ biến.

Ở nước ngoài cách tiếp cận này cũng được sử dụng rộng rãi, chẳng hạn khi giới thiệu PL/I trên cơ sở ALGOL và FORTRAN [1, 2] hoặc khi nghiên cứu có hệ thống các ngôn ngữ lập trình nói chung.

Một ưu điểm khác của cuốn sách là nó có thể được dùng để tra cứu các thuật toán cơ bản, nhất là các thuật toán sắp xếp và thuật toán tìm kiếm nhị phân. Cuốn sách được in loát đẹp, ít lỗi.

Xin giới thiệu « Lập trình song ngữ PASCAL-BASIC » với bạn đọc làm tin học.

TÔ TUẤN

cus

TRAO ĐỔI

VỀ CÁC TIÊU CHUẨN SẢN PHẨM PHẦN MỀM

NGUYỄN XUÂN HUY

I - ĐẶT VẤN ĐỀ

Trong nội dung của đề tài "Quy thuật toán và chương trình" thuộc Chương trình tin học nhà nước giai đoạn 1986-1990 số đề cấp đến yêu cầu sớm ban hành những qui định về cấu trúc của hệ sơ và các tiêu chuẩn của những sản phẩm phần mềm trên máy tính để làm cơ sở cho việc nghiệm thu các đối tượng thuộc phạm vi quản lý của qui thuật toán và chương trình quốc gia.

Mục đích của bài này là thử đề xuất một vài quan niệm về sản phẩm phần mềm và những tiêu chuẩn của nó.

Trong phần II đề nghị một định nghĩa và một bảng phân loại các sản phẩm phần mềm dùng trên các họ máy tính Von Neumann hiện hành.

Phần III là những đề xuất về tiêu chuẩn của một sản phẩm phần mềm.

Phần cuối cùng, phần IV nêu một số ý kiến sơ bộ về việc vận dụng những tiêu chuẩn nêu ở phần III trong việc đánh giá một sản phẩm phần mềm cụ thể.

II - SẢN PHẨM PHẦN MỀM LÀ GÌ?

Sản phẩm phần mềm (hay sản phẩm chương trình) là một (bộ) chương trình thực hiện một nhiệm vụ tương đối độc lập nhằm phục vụ cho một ứng dụng cụ thể việc quản lý hoạt động của máy tính hoặc áp dụng máy tính trong các hoạt động kinh tế, quốc phòng, văn hóa, giải trí.

Thí dụ về sản phẩm phần mềm:

- Hệ điều hành MS DOS 2.00
- Chương trình dịch FORTRAN EC
- Hệ soạn thảo văn bản WORDSTAR 3.0
- Hệ quản trị cơ sở dữ liệu DBASE III PLUS
- Trò chơi Deathlon
- Hệ phát triển FT 86K.

Đối với các họ máy tính hoạt động theo các nguyên lý Von Neumann hiện nay có thể tạm chia các sản phẩm phần mềm thành 6 nhóm sau đây:

Nhóm 1: Các hệ điều hành bao gồm:

- 1.1. Các driver
- 1.2. Các monitor
- 1.3. Các hệ quản lý tệp
- 1.4. Các hệ thống quản lý thư viện chương trình dịch

Nhóm 2: Các chương trình biên dịch và thời kỳ dịch như PASCAL (CPCOL, LISP, BASIC, FORTRAN, ADA, C, v.v...)

.
Nhóm 3: Các chương trình hệ thống bao gồm:

3.1. Các chương trình soạn thảo văn bản

3.2. Các chương trình phục vụ thiết bị tạo khuôn cho đĩa sao chép, điều khiển thiết bị ngoại vi

3.3. Các chương trình mở rộng các chức năng quản lý tệp: sắp xếp, sao chép, cập nhật v.v...

3.4. Các chương trình đồ thị

.

Nhóm 4: Các hệ quản trị cơ sở dữ liệu, trí thức và các chương trình mở rộng tương ứng.

Nhóm 5: Các chương trình ứng dụng có tính chất hệ thống

5.1. Các chương trình xử lý dữ liệu đa năng

5.2. Các bộ chương trình phục vụ cho các yêu cầu tính toán cơ sở

5.3. Các chương trình tối ưu hóa

5.4. Các hệ expert (chuyên gia) và các hệ tương tự

5.5. Các hệ mô phỏng

5.6. Các lớp ngôn ngữ dữ liệu và trí thức phục vụ cho việc khai thác các cơ sở dữ liệu và trí thức

5.7. Các hệ tự động hóa quản lý chuyên ngành

5.8. Các hệ tự động hóa thiết kế

5.9. Các hệ thống tự học

5.10. Các hệ thống dạy học

5.11. Các hệ thống tự động phát sinh, kiểm sửa chương trình

5.12. Các hệ chương trình nhận dạng, phân tích và tổng hợp tiếng nói, hình ảnh, tín hiệu

5.13. Các hệ chương trình điều khiển quy trình và các thiết bị công nghiệp

.

Nhóm 6: Các chương trình tiện ích

6.1. Các chương trình xử lý bảng điện tử

6.2. Các chương trình chuyển đổi (liên dịch) ngôn ngữ, dịch chéo, khôi phục

6.3. Các chương trình trò chơi

.

III - CÁC TIÊU CHUẨN CHO MỘT SẢN PHẨM PHẦN MỀM

Một sản phẩm phần mềm cần được xem xét theo các tiêu chuẩn sau đây:

1. **Tính đúng đắn:** Tính đúng đắn của sản phẩm thể hiện ở chỗ sản phẩm đó thực hiện chính xác những mục tiêu và chức năng được đề xuất khi thiết kế. Thí dụ, chức năng của sản phẩm SF là sắp xếp một tệp có dung lượng (số bản ghi) tùy ý, theo một trường tùy ý với chiều tăng, giảm tùy ý thì những tình huống sau đây sẽ làm mất tính đúng đắn của chương trình SF:

- Không thực hiện với những tệp trống (tệp không chứa bản ghi nào).

- Không thực hiện hoặc thực hiện sai với những tệp có trên 100 trường hoặc có trên 10^4 bản ghi.

- Không thực hiện hoặc thực hiện sai với những trường có chiều dài lớn hơn 125 byte trong trường hợp yêu cầu sắp xếp theo trật tự giảm dần.

Tính đúng đắn của sản phẩm được xác định qua những căn cứ sau đây:

Tính đúng đắn của thuật toán: Thuật toán phải được chứng minh trên cơ sở toán học hoặc tác giả của sản phẩm phải chỉ rõ nguồn tài liệu cung cấp thuật toán có kèm theo chứng minh.

— Tính tương đương của chương trình với thuật toán đã cho. Thuật toán có thể đúng nhưng chương trình lập ra không tương đương với thuật toán, nên khi thực hiện sẽ cho kết quả sai. Trong trường hợp này, người ta nói là việc lập mã (lập trình) cho thuật toán bị sai.

— Tính đúng đắn của chương trình có thể được chứng minh trực tiếp qua văn bản chương trình.

— Tính đúng đắn của chương trình có thể được khẳng định dần dần qua việc kiểm thử (testing), qua việc áp dụng trong một khoảng thời gian đủ lớn, trên một diện khá rộng, với tần suất sử dụng cao.

— Các tác giả của sản phẩm cần cung cấp đầy đủ những luận chứng và kết quả xác nhận tính đúng đắn của sản phẩm. Những tư liệu đó bao gồm:

+ Thuật toán có kèm chứng minh và chú thích số dòng lệnh tương ứng trong chương trình.

+ Chương trình có kèm các luận đề phục vụ cho việc kiểm chứng.

+ Các phương pháp và kỹ thuật kiểm thử chương trình.

+ Các kết quả chạy thử.

+ Các giấy chứng nhận, nhận xét của các cá nhân và cơ sở đã khảo sát hoặc áp dụng chương trình.

2. Tính khoa học: Tính khoa học của sản phẩm được thể hiện qua những mặt sau:

a) Khoa học về cấu trúc: Bản thân sản phẩm được chia thành những đơn vị cân đối, không trùng lặp nhau về chức năng, có quan hệ hữu cơ với nhau, có thể tổ hợp thành nhiều khả năng. Bản thân thuật toán và chương trình được thiết kế và cài đặt một cách có cấu trúc.

b) Khoa học về nội dung: các thuật toán dựa trên những thành tựu mới của toán học và tin học, có cơ sở chặt chẽ. Các chức năng và nhiệm vụ do sản phẩm thực hiện có giá trị khoa học cao.

c) Khoa học về hình thức thao tác: Tên của các lệnh phải hợp lý, thể hiện tính lô-gic và phù hợp với tư duy tự nhiên của người dùng. Thí dụ, một hệ thống dùng một số lệnh bằng tiếng Anh, một số lệnh bằng tiếng Việt sẽ được xem là một hệ thống không đảm bảo tính khoa học về hình thức thao tác. Trong trường hợp như vậy, cách giải quyết tối là thiết kế 2 chế độ thao tác: một bằng tiếng Anh, một bằng tiếng Việt. Ở mức linh hoạt cao hơn, có thể cho phép người dùng tự chuyển đổi tên lệnh theo ý muốn.

Các thông báo lỗi phải rõ ràng và bao gồm những chú giải sau: lỗi số mấy, nội dung lỗi, nơi xảy ra lỗi, cách giải quyết

3. Tính hữu hiệu: Tính hữu hiệu của một sản phẩm mềm được xác định qua những tiêu chuẩn sau:

a) Hiệu quả kinh tế hoặc ý nghĩa, giá trị thu được do áp dụng sản phẩm đó.

b) Tốc độ xử lí của sản phẩm (v) tính bằng tỷ lệ giữa khối lượng đối tượng được xử lí (m) và tổng số đơn vị thời gian cần thiết xử lí khối lượng trên (t):

$$v = \frac{m}{t}$$

c) Giới hạn tối đa của sản phẩm hay miền xác định của chương trình, được xác định qua khối lượng tối đa của các đối tượng mà sản phẩm đó quản lý, thí dụ, hệ QF có thể quản lý tối đa 10^6 bản ghi trong một tệp, mỗi bản ghi có thể có số trường tối đa là 255.

d) Dung lượng tối đa của miền nhớ trong mà chương trình sử dụng, thí dụ, hệ QF sử dụng 48K bộ nhớ trong, từ địa chỉ... đến địa chỉ...

4. Tính sáng tạo: Tính sáng tạo của sản phẩm thể hiện qua các đặc điểm sau:

a) Sản phẩm phần mềm đó được sản xuất lần đầu tiên trên thế giới.

b) Sản phẩm phần mềm đó được sản xuất phục vụ cho những đặc thù riêng của Việt Nam, như xử lí văn bản tiếng Việt, bộ chương trình nhận dạng các loại men gốm, hoa văn trang trí v.v..

c) Sản phẩm phần mềm đó có những điểm khác về mặt nguyên lý so với các sản phẩm truyền thống hiện hành

d) Sản phẩm phần mềm đó mang những ưu thế nổi bật so với các sản phẩm cùng loại hiện hành

5. **Tính an toàn:** Sản phẩm (trong trường hợp thấy cần thiết) có cơ chế bảo mật và bảo vệ các đối tượng do nó phát sinh hoặc quản lý. Bản thân sản phẩm cũng được đặt trong một cơ chế bảo mật nhằm chống sự sao chép trộm hoặc làm biến dạng sản phẩm đó.

6. **Tính toàn vẹn:** Tính toàn vẹn của sản phẩm thể hiện qua các chức năng sau:

- Không gây ra những nhập nhằng trong quá trình thao tác, đảm bảo tính nhất quán về cú pháp, có cơ chế ngăn ngừa việc phát sinh ra những đối tượng (dữ liệu, tri thức, đơn thể v.v...) sai qui cách hoặc mâu thuẫn với các đối tượng có sẵn.

- Có cơ chế khôi phục lại toàn bộ hoặc một phần những đối tượng thuộc diện quản lý của sản phẩm trong trường hợp có sự cố như hỏng máy, mất điện đột ngột.

- Có các cơ chế sao chép đề cất giữ, dự phòng.

7. **Tính đầy đủ:** Tính đầy đủ của sản phẩm được đánh giá qua tập các chức năng mà sản phẩm đó đảm nhiệm. Tập các chức năng cần và nên thỏa mãn các tính chất sau đây:

- Tính **đối xứng:** Nếu có thao tác phát sinh thì nên có thao tác hủy bỏ và ngược lại. Thí dụ, một hệ thống quản lý các bằng biểu đã có thao tác xen thêm một số cột vào bằng hiện có thì cần và nên có thêm thao tác ngược lại là xóa (hoặc bỏ) một số cột từ bằng hiện có.

- Tính tiêu chuẩn: Sản phẩm cần đạt được một số tiêu chuẩn tối thiểu được thừa nhận trên thị trường thế giới hoặc trong khoa học.

Thí dụ, một hệ quản lý tệp cần có các chức năng tối thiểu sau đây:

+ Phát sinh tạo lập các tệp

+ Hủy bỏ các tệp

+ Đổi tên các tệp

+ Sao chép các tệp

+ Tờ chức thư mục cho các tệp

+ Nạp và cập nhật dữ liệu vào tệp

+ Có thể quản lý được các tệp văn bản, các tệp lưu trữ chương trình trong một (hoặc một số) hệ điều hành

+ Các lệnh thao tác tệp có thể hoạt động như những đơn thể độc lập cũng như những vào trong một số ngôn ngữ lập trình cao cấp.

8. **Tính độc lập:** Một sản phẩm phần mềm nên và cần thể hiện được tính độc lập đối với các đối tượng sau:

a) Độc lập đối với thiết bị: Sản phẩm đó có thể cài đặt một cách dễ dàng trên nhiều loại máy và có thể quản lý được nhiều loại thiết bị. Những sửa đổi để thích nghi là không đáng kể.

b) Độc lập với cấu trúc của đối tượng mà sản phẩm đó quản lý: Thí dụ, một hệ quản trị cơ sở dữ liệu có tính độc lập cao sẽ không đòi hỏi người dùng phải viết lại chương trình ứng dụng khi chuyển từ việc quản lý các tệp tuần chỉ sang việc quản lý các tệp tuần tự.

c) Độc lập với nội dung của đối tượng mà sản phẩm đó quản lý. Thí dụ, đơn thể COPY có thể sao chép các tệp, không phụ thuộc gì vào nội dung cụ thể của các tệp đó ra sao; tệp trống nay không, tệp lưu trữ văn bản hay chương trình, tệp lưu trữ mã nhị phân hay mã trung gian v.v...

9. **Tính phổ dụng:** Sản phẩm phần mềm có thể áp dụng cho nhiều lĩnh vực theo nhiều chế độ làm việc khác nhau.

10. **Tính đơn giản:** Sản phẩm có tính đến những yếu tố tâm lý của đông đảo người dùng như:

- Dễ thao tác

- Dễ học và dễ hoàn thiện

- Ngôn ngữ (hoặc tên các lệnh, thực đơn, thông báo, v.v...) trong sáng, dễ hiểu.

11. **Tính dễ phát triển, hoàn thiện:** Sản phẩm có thể mở rộng cho các phương án khác hoặc mở rộng, tăng cường về mặt chức năng một cách dễ dàng.

IV - VẬN DỤNG

Các tiêu chuẩn trình bày trong mục III cần được lượng hóa cụ thể. Một trong những phương pháp lượng hóa tiêu chuẩn là định ra thang điểm. Thang điểm này sẽ bao gồm từng nhóm điểm cho mỗi tiêu chuẩn. Trong mỗi tiêu chuẩn từng mục của tiêu chuẩn đó sẽ được qui định một điểm cụ thể. Việc qui định số lượng điểm cho từng tiêu chuẩn và từng mục của mỗi tiêu chuẩn sẽ được cân đối theo trọng số thể hiện mức độ quan trọng của tiêu chuẩn này so với các tiêu chuẩn khác.

Sau khi đã qui định thang điểm, hội đồng nghiệm thu sẽ qui định mức nghiệm thu cho một sản phẩm mềm, mức nghiệm thu sẽ được căn cứ vào tổng số điểm mà sản phẩm đó đạt được tính theo toàn bộ các tiêu chuẩn và hoặc một số tiêu chuẩn quan trọng. Thí dụ mức nghiệm thu thấp nhất cho một sản phẩm phần mềm là

- Tổng số điểm phải đạt trên 160/250, trong đó
- Tiêu chuẩn 3 phải đạt trên 15/20, và
- Tiêu chuẩn 6 phải đạt trên 10/20,

Nhận ngày 17-3-1987

РЕЗЮМЕ

ОБ НОРМАХ ДЛЯ ОЦЕНКИ ПРОГРАММНЫХ ПРОДУКТОВ

В заметке предлагается набор норм для оценки программных продуктов. По каждой норме перечисляются и выясняются основные требования к программам.

Рассматриваются следующие требования: правильность, научность, эффективность, инициативность, безопасность, целостность, полнота независимость, универсальность, простота и легко усовершенствование

HỘI THẢO QUỐC TẾ VỀ TOÁN HỌC TÍNH TOÁN TẠI THÀNH PHỐ HỒ CHÍ MINH, THÁNG 4/1988

Viện khoa học Việt Nam phối hợp với Hội đồng quốc tế về Toán học trong các nước đang phát triển (gọi tắt là ICOMIDC) sẽ tổ chức một Hội thảo quốc tế về Toán học tính toán tại thành phố Hồ Chí Minh, từ ngày 25 đến 29-4-1988.

Hội thảo khoa học này có mục đích chủ yếu là thúc đẩy và tăng cường sự trao đổi khoa học, sự hiểu biết lẫn nhau và sự hợp tác giữa các nhà toán học trong các nước đang phát triển, đặc biệt là trong vùng Đông nam và Nam châu Á, với nhau cũng như với các nhà toán học trên thế giới. Chủ đề của hội nghị là các vấn đề toán học của tính toán, từ các phương pháp giải tích số, các phương pháp giải các bài toán đại số, phương trình vi phân, các phương pháp xấp xỉ, các bài toán mô phỏng, cho đến các phương pháp giải các bài toán có cấu trúc rời rạc, việc phân tích hiệu quả và độ phức tạp của các thuật toán, những vấn đề lý thuyết chung về tính toán, các cấu trúc tính toán, các thuật toán tính song song, v.v...

Hội thảo được sự ủng hộ về tinh thần cũng như giúp đỡ về tài chính của một số tổ chức khoa học quốc tế, như: Trung tâm quốc tế về Vật lý lý thuyết tại Trieste, Italia; UNESCO, Hội Toán học quốc tế, Hội Toán học Đông Nam Á, Học viện công nghiệp châu Á, v.v...

Hơn 100 cán bộ nghiên cứu và ứng dụng toán học trong nước ta đã đăng ký tham dự hội thảo, trong đó có hơn 60 cán bộ đăng ký thông báo khoa học tại hội thảo.

Số khách nước ngoài được mời đọc bài giảng hoặc đăng ký tham dự hội thảo vào khoảng 60 người từ hơn 20 nước, bao gồm các nước trong vùng Đông Nam Á, Ấn Độ, các nước xã hội chủ nghĩa và một số nước khác như Pháp, Mỹ, Anh, Canada, Nhật, v.v...

Chương trình khoa học của Hội thảo bao gồm hơn 30 báo cáo mới (mỗi báo cáo trình bày 45 phút) của các nhà toán học từ nhiều nước khác nhau và hơn 70 thông báo khoa học (mỗi thông báo trình bày trong 20 phút). Về các thuật toán số trong đại số tuyến tính có các báo cáo của M. Maesynski (Ba Lan), E.M. Lagare (Philippin); về phương trình vi phân thường có các báo cáo của D.E. Daykin, E.L. Ortiz (Anh), T. Mitsui (Nhật); về phương trình đạo hàm riêng và ứng dụng có các báo cáo của F. Michel (Canada), M. Frémond (Pháp), J.A. Donaldson (Mỹ), O. Fujitwara (Nhật), K.H. Hoffmann (CHLB Đức); về việc chính qui hóa đối với các bài toán không chỉnh có các báo cáo của R. Gorenflo (CHLB Đức), Đ.Đ. Ánh (Việt Nam); về các phương pháp xấp xỉ và mô phỏng có các báo cáo của B. Sendov (Bungari), H.H. Tuq (Canada); về quan hệ giữa ngẫu nhiên và lý thuyết tính toán có các báo cáo của N.N. Vakhramein (Liên Xô), H.N. Phien (AIT); về tối ưu hóa và qui hoạch toán học có các báo cáo của H. Tuy (Việt Nam), Iu.G. Evtushenko (Liên Xô); về các phương pháp biến đổi Fourier nhanh có báo cáo của N.X. Lộc (Việt Nam); về các bài toán rời rạc có các báo cáo của Asharya (Ấn Độ), S. Skyum (Đan Mạch), L. Budaeh (CHDC Đức), J.M. Steyaert (Pháp), P.Đ. D'ệu (Việt Nam); về lý thuyết tính toán và độ phức tạp có các báo cáo của M. Nivat (Pháp), R. Srimoney và D.L. Văn (Ấn Độ và Việt Nam), J. Diaz (Tây Ban Nha); về những vấn đề toán học của lập trình có các báo cáo của G. Huet (Pháp), J. Demetrovics (Hungari); về các thuật toán tính song song có các báo cáo của J. Miklosko (Tiệp Khắc), I. Lavallé (Pháp) v.v... Các thông báo khoa học của cán bộ toán học Việt Nam cũng như của nhiều khách nước ngoài chứa đựng nhiều kết quả nghiên cứu phong phú và mới mẻ trong các lĩnh vực của toán học tính toán. Ngoài các buổi báo cáo khoa học, trong chương trình Hội thảo cũng sẽ có một số buổi dành cho việc thảo luận về sự hợp tác khoa học giữa các nhà toán học trong các nước đang phát triển, đặc biệt trong khu vực; trong đó có buổi trao đổi ý kiến và thảo luận về một dự án hình thành "Mạng lưới khu vực về Toán ứng dụng", một hình thức hợp tác nghiên cứu theo một số chủ đề lựa chọn của các nhà toán học trong khu vực.

Hội thảo khoa học về Toán học tính toán lần này là một dịp để anh chị em Toán học nước ta có điều kiện tiếp xúc rộng rãi với các bạn đồng nghiệp trong khu vực cũng như trên thế giới nói chung, do đó mở rộng các quan hệ trao đổi và hợp tác, nhằm thúc đẩy sự phát triển nghiên cứu và ứng dụng toán học ở nước ta.

P.Đ.D.

MỤC LỤC

(СОДЕРЖАНИЕ – CONTENTS)

	<i>Trang</i>
ĐỖ VIỆT ANH – Một số phương pháp tự động hóa tổng hợp văn bản Некоторые методы автоматизированного синтеза документов. Some methods of document automatic synthesis.	1
NGUYỄN VĂN TAM – Thẻ CPU-Z80 trên bus VT-64. Карта CPU-Z80 на линии VT-64. Card CPU-Z80 on the bus VT-64.	5
NGUYỄN VĂN LƯU, HỒ THUẬN – Cài đặt ngôn ngữ PADRE bằng kỹ thuật tiền xử lý. Имплементация PADRE методом предпроцессора. On the PADRE's implementation by preprocessor technique.	9
NGUYỄN NGỌC KIÊN, ĐOÀN THANH VINH, VŨ DUY MÃN – Thực hiện liên tục cấp thấp trong xây dựng hệ đa xử lý từ hai máy tính mini 16. Реализация низко-уровневой коммуникации в конструировании мульти-обрабатывающей системы из двух ЭВМ мини 16. Realisation of low-level communication in construction of the multi-processing system from two computers MINI 16	16
NGUYỄN NHƯ CẬN – Hệ TV-86. Система TV-86. System TV-86.	21
NGUYỄN CHÍ CÔNG – PS/2 và những tiêu chuẩn tương lai. PS/2 и стандартные продукты в будущее. PS/2 and standards in future.	23
HỒ THUẬN – Hội thảo công nghệ phần mềm.	25
GIỚI THIỆU SÁCH (НОВЫЕ КНИГИ-BOOK REVIEW)	
Lập trình song ngữ PASCAB-BASIC của Nguyễn Xuân Huy.	26
NGUYỄN XUÂN HUY – Về các tiêu chuẩn sản phẩm phần mềm. Об нормах для оценки программных продуктов. On the norms for evaluation of software products.	27
Giới thiệu hội thảo quốc tế về toán học tính toán tại Tp HCM, tháng 4-1988	32

Người gửi đăng bài cần chú ý

1. Bài gửi đăng phải viết rõ ràng, có đánh số từng trang. Nếu là công trình nghiên cứu, không dài quá 8 trang viết tay, có phần tóm tắt nội dung không dài quá nửa trang bằng một trong hai thứ tiếng: Nga, Anh. Nếu là bài giới thiệu tổng quan, dài không quá 16 trang viết tay. Dịch đầu đề các bài ra tiếng Nga và tiếng Anh.

2. Các công thức, kí hiệu trong bài cần viết ngay ngắn, chân phương, đúng vị trí và tỉ lệ to nhỏ. Số hiệu đánh số các công thức thống nhất ghi ở cuối dòng bên phải các công thức.

3. Các hình nên vẽ riêng trên giấy trắng. Trong bài có ghi chỗ đặt hình và chú thích cho từng hình.

4. Những từ dùng nên thống nhất với các quy định từ điển toán học do Ủy ban Khoa học và kỹ thuật nhà nước xuất bản. Những từ mới cần được ghi chú thích dưới trang viết các từ tương ứng với chúng bằng tiếng nước ngoài.

5. Tên tác giả, nhà xuất bản, địa danh thuộc hệ chữ La tinh thì dùng nguyên gốc, nếu thuộc hệ chữ Nga thì chuyển sang hệ chữ Việt như sau: (А) → A, (Б) → B, (В) → V, (Г) → G, (Д) → D, (Е, Э) → E, (Ж) → YO, (К) → J, (З) → Z, (И, Ё) → I, (К) → K, (Л) → L, (М) → M, (Н) → N, (О) → O, (П) → P, (Р) → R, (С) → X, (Т) → T, (У) → U, (Я) → YA, (Ю) → YU, (ѳ) → U, (Ц) → TX, (Ш, Щ) → S, (Ч) → CH, (Ф) → F, (Х) → KH. Bỏ các dấu Ъ, Ь.

6. Tài liệu tham khảo và trích dẫn ghi theo thứ tự sau: Nếu là bài báo: Tên tác giả, tên bài báo, tên báo, tập, số, năm xuất bản, từ trang ... đến trang. Nếu là sách: Tên tác giả, tên sách, nhà xuất bản, nơi xuất bản, năm xuất bản, từ trang... đến trang.

7. Bài gửi phải có ghi họ tên, địa chỉ cơ quan và số điện thoại liên lạc với tác giả.

8. Bài gửi đăng và thư gửi Tòa soạn gửi theo địa chỉ: Tạp chí Khoa học tính toán và điều khiển, 70 Trần Hưng Đạo, Hà Nội. Dãy số: 52825.

9. Bài không đăng không trả lại bản thảo.

TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN

In 1300 cuốn, khổ 19 × 26,5. Số in 19/T2 tại Nhà máy in Điện Hồng, Hà Nội. In xong và nộp lưu chiểu tháng 3-1988.