

Rus

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

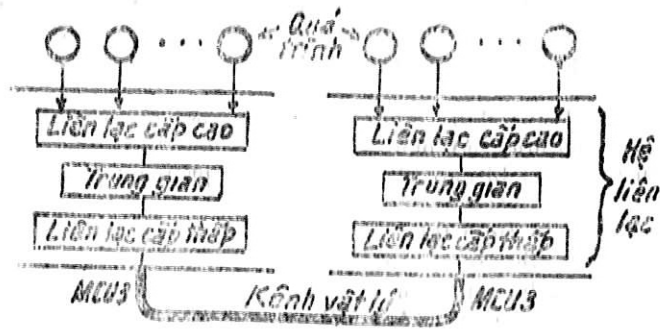
NGUYỄN NGỌC KIẾN
ĐOÀN THANH VỊNH
VŨ DUY MÃN

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 và 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 cửa 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, dị 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.

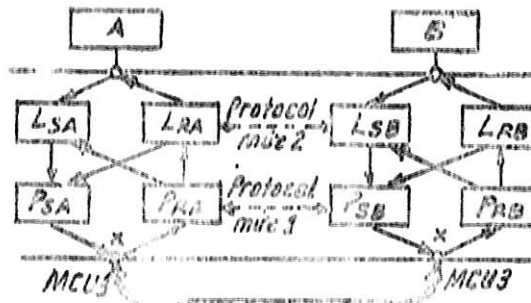


Hình 1

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 ở tầng 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

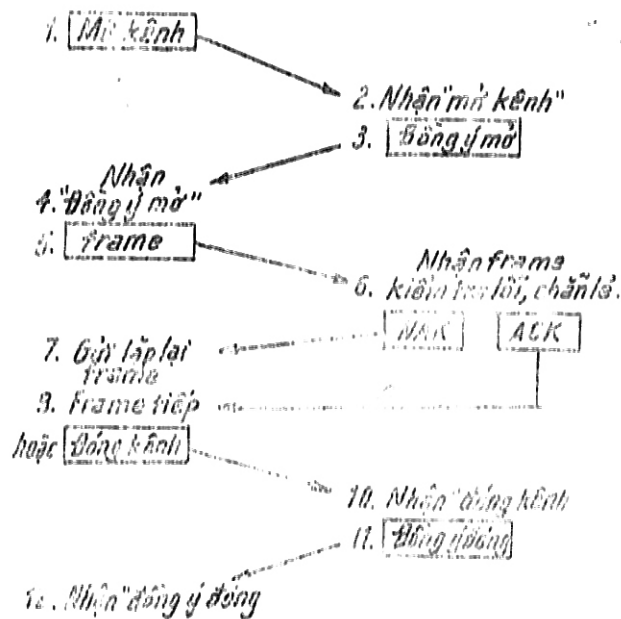
Nhu cầu trao đổi ở mỗi bên được qui tụ lại trong một đối tượng tổng quát là A và B và liên lạc cấp thấp phải đảm bảo chuyển các tin báo từ A sang B và ngược lại mỗi khi có yêu cầu. Công việc để đáp ứng một yêu cầu như vậy gọi là một hoạt động gửi - nhận. Các thực thể của mức 2 có nhiệm vụ phối hợp với 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). Cấp LSA-LRB phối hợp với nhau theo protocol mức 2 để thực hiện hoạt động gửi - nhận từ A sang B (tương tự cấp 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 lại một số lần khi có lỗi. Mỗi đoạn tin trực tiếp chứa 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 khôi 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. Weitzman 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 MCU 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.