

THIẾT KẾ VÀ CÀI ĐẶT HỆ CHƯƠNG TRÌNH ĐIỀU KHIỂN TERMINAL VÌ TÍNH CHUYÊN DỤNG MP 1000 VÀO HỆ ĐIỀU HÀNH CỦA MÁY TÍNH MINI 16

ĐỖ MINH THÁI

1. ĐẶT VẤN ĐỀ

Hệ vi tính chuyên dụng MP1000 là một thiết bị đầu cuối thông minh được điều hành bởi MP8080A với hệ mềm chuyên dụng của nó cho phép thực hiện các chức năng vẽ đồ thị trên các màn hình màu. Máy tính điện tử MINI 16 là một hệ mini, được thiết kế hướng vào các nhu cầu xử lý tin ở chế độ thời gian thực (real-time) và trực tiếp (on-line), hoạt động với các hệ điều hành D ϕ S (Disk Operating System) và MAS (Multi-Application System), là thiết bị điều khiển trung tâm trong một qui trình điều khiển khép kín. Hai hệ này đã được ghép nối với nhau về mặt kỹ thuật phần cứng [13] và hoạt động theo chế độ chủ-tớ. Mọi sự điều khiển từ MINI 16 đối với MP 1000 đều thông qua một ngôn ngữ gọi là « ngôn ngữ trao đổi », Ngôn ngữ này bao gồm các loại tin báo (message) khác nhau, được xây dựng trên cơ sở các thủ tục trao đổi đã được qui ước sẵn và đã được cài đặt về phía MP 1000. Vấn đề đặt ra là phải cài đặt được thủ tục trao đổi này cho phía MINI 16, tức là mở rộng các hệ điều hành của nó nhằm điều khiển thêm một ngoại vi chuyên dụng, và hơn thế, phải thiết kế và cài đặt hệ chương trình sao cho, theo một nghĩa nào đó, « tối ưu » đồng thời theo các chỉ tiêu :

- « tối ưu » về cấu trúc hệ,
- « tối ưu » về độ phức tạp thuật toán (thời gian tính, dung lượng bộ nhớ, độ dài chương trình), trong đó, chỉ tiêu về thời gian tính được ưu tiên.

Bài báo này trình bày phương pháp thiết kế và giải pháp cài đặt trên cơ sở vận dụng và phát triển một phương pháp thiết kế phần mềm cũng như tiến trình thực hiện hướng theo các mục đích trên và đánh giá sản phẩm đạt được.

2. PHƯƠNG PHÁP THIẾT KẾ HỆ CHƯƠNG TRÌNH ĐIỀU KHIỂN THIẾT BỊ MP 1000.

2.1. Về các chỉ tiêu chất lượng đối với một hệ mềm

Theo [5], hơn 80% giá thành của một sản phẩm phần mềm trong khoảng « thời gian sống » của nó thuộc về phần kiểm tra và bảo hành. Do vậy các chỉ tiêu chất lượng hệ mềm đã trở thành một trọng tâm được chú ý trong những năm gần đây [11], đặc biệt là các chỉ tiêu dễ thay đổi và mở rộng, thuận tiện cho việc cài đặt và hiệu chỉnh.

Với đối tượng thiết kế cụ thể đề cập ở đây, sau khi cài đặt xong chương trình điều khiển thiết bị MP 1000 (gọi tắt là DRMP) vào D ϕ S ta còn phải thay đổi để cài đặt vào MAS và như vậy phải thiết kế hệ chương trình có chi phí là ít nhất.

2.2. Về phương pháp thiết kế theo cấu trúc và vấn đề áp dụng đối với DRMP

Phương pháp thiết kế theo cấu trúc (structured design) do L. L. Constantine khởi xướng [10]. Tính ưu việt của phương pháp này đối với các bài toán thiết kế phức tạp đã được khẳng định trong [7]. Constantine cho rằng phương pháp này làm cho việc viết chương trình, hiệu chỉnh và thay đổi dễ hơn, nhanh hơn và rẻ hơn bằng cách giảm độ phức tạp cấu trúc

của hệ thống, đồng thời phương pháp này cũng tương thích với kỹ thuật lập trình theo cấu trúc (structured programming).

Nghiên cứu và vận dụng các tư tưởng chính của thiết kế theo cấu trúc cho việc thiết kế hệ chương trình điều khiển hệ vi tính chuyên dụng MP1000 nhằm cài đặt vào các hệ điều hành của MINI16, chúng tôi chia quá trình thiết kế làm 3 giai đoạn:

(1) Thiết kế tổng thể

(2) Thiết kế chi tiết

(3) mô phỏng logic điều khiển của hệ chương trình, đồng thời phát triển thêm một bước phương pháp này bằng cách đưa tư tưởng « tối ưu hóa » hệ thống đồng thời theo các chỉ tiêu tối ưu về cấu trúc hệ và tối ưu về độ phức tạp thuật toán (theo nghĩa thời gian tính, dung lượng bộ nhớ, độ dài chương trình) một cách xuyên suốt trong toàn bộ quá trình phát triển hệ mềm từ giai đoạn phân tích đến thiết kế, cài đặt và hiệu chỉnh.

Trong các giai đoạn thiết kế, ở giai đoạn (1) và (2) chúng tôi chủ yếu vận dụng các tư tưởng đề cập trong [10], [16], [6] đồng thời xác định tần số sử dụng các đơn thể (modula) nhằm tập trung thực hiện tối ưu hóa về độ phức tạp thuật toán trong các giai đoạn cài đặt và hiệu chỉnh cho các đơn thể có tần số sử dụng cao.

Tiếp theo hai giai đoạn (1) và (2) thường người ta bắt tay ngay vào giai đoạn viết chương trình, điều đó có thể kéo theo nhiều sai sót của giai đoạn thiết kế không bị loại bỏ. Theo Schulz [9], vận dụng kinh nghiệm quá trình công nghệ thiết kế phần cứng, người ta nên thực hiện thêm hai giai đoạn nữa, đó là giai đoạn xác định logic điều khiển của hệ chương trình và mô phỏng logic điều khiển đó bằng một « ngôn ngữ thiết kế » hoặc bằng một ngôn ngữ nào đó tương đương để thực hiện trên MTDI—thực tế thì chưa có một ngôn ngữ thiết kế thực sự nào cho phương pháp thiết kế theo cấu trúc mà chỉ đang ở dạng đề nghị [17]. Với đối tượng thiết kế đặc thù là DRMP, tận dụng một kinh nghiệm thực tế là trước khi bắt tay vào viết chương trình cài đặt vào hệ điều hành bắt buộc người lập chương trình phải trải qua giai đoạn kiểm tra hoạt động thật sự của thiết bị, chúng tôi chủ trương ghép hai giai đoạn của Schulz vào giai đoạn (3) và tận dụng việc kiểm tra hoạt động của thiết bị vào việc mô phỏng hoạt động của hệ chương trình sẽ được cài đặt sau đó.

Dưới đây là những nét chung nhất mô tả từng giai đoạn thiết kế đối với DRMP.

Giai đoạn 1: Nhiệm vụ cần thực hiện trong giai đoạn này là phân chia các chức năng của hệ Chương trình theo cách từ trên xuống (top down).

Vấn đề cần giải quyết đối với DRMP là thể hiện thủ tục trao đổi giữa MINI 16 với MP 1000 thông qua bộ điều khiển thiết bị MCU3 với thủ tục vào-ra và chế độ ngắt đã được qui định cho các kênh trao đổi thông tin của nó (programmed channel/I/O processor). Dưới góc độ của một hệ quản lý vào-ra ta có thể chia nhiệm vụ cần giải quyết ra làm hai chức năng:

1. Thể hiện thủ tục trao đổi.

2. Thực hiện các thủ tục vào-ra cơ sở phù hợp với MINI 16 và bộ điều khiển MCU3.

Ta có thể biểu diễn các chức năng trên trong hệ quản lý vào-ra của DQS (MAS) theo cấu trúc phân cấp như trong h. 1.

trong đó:

Mức 1. 1: làm các thao tác vào-ra cơ sở theo nghĩa của MINI16, tức là hoặc input hoặc output, đảm bảo chế độ trao đổi thông tin và chế độ ngắt của bộ điều khiển MCU3.

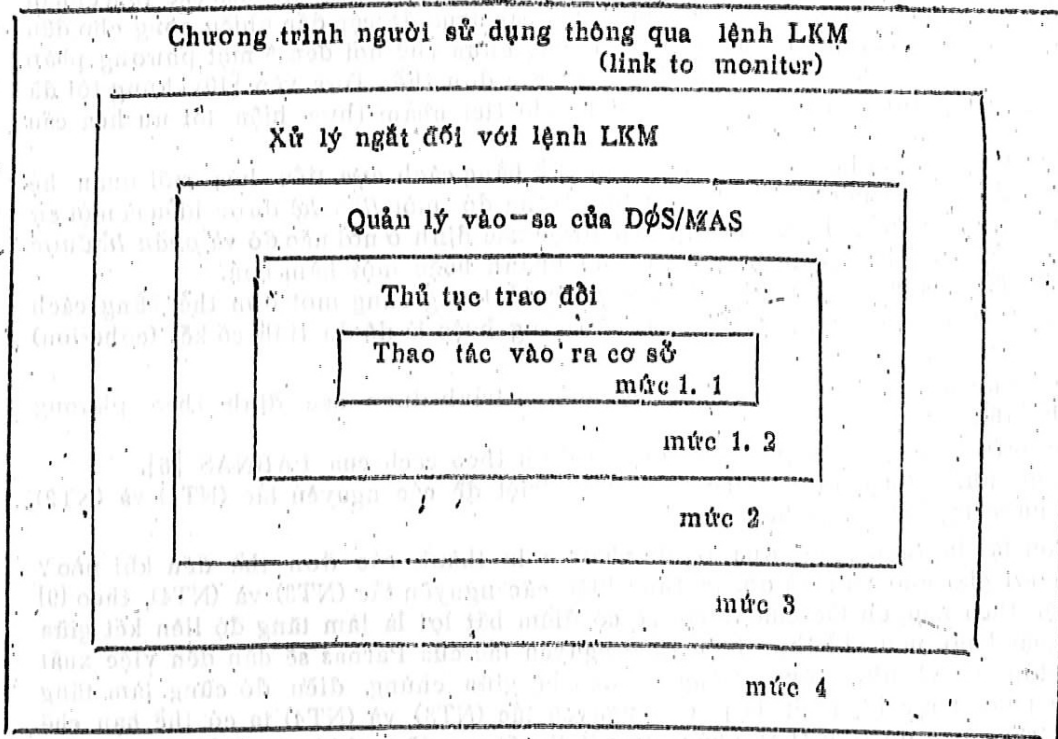
Mức 1. 2: Nhận các tham số từ chương trình người sử dụng, thể hiện các loại thủ tục trao đổi cho từng dạng tin báo (xem chi tiết trong [15]). Ta có thể biểu diễn sơ đồ tổng quát theo chức năng của DRMP như trong h. 2.

Giai đoạn 2: Giai đoạn này có hai nhiệm vụ [9]:

1. Tiếp tục phân chia các chức năng theo cấu trúc phân cấp từ trên xuống.

2. Xác định dòng điều khiển và dòng dữ liệu giữa các đơn thể.

Vấn đề được đặc biệt quan tâm giải quyết ở giai đoạn này là: tối ưu hóa cấu trúc và tối ưu đồng thời theo các chỉ tiêu về giảm độ phức tạp cấu trúc và giảm độ phức tạp thuật toán.



Hình 1. Cấu trúc phân cấp hệ quản lý vào-ra với DRMP

Mục đích của vấn đề tối ưu hóa ở đây là tìm ra một cấu trúc cho hệ chương trình đảm bảo tính đơn giản theo nghĩa khi ta biểu diễn các mối liên hệ giữa các đơn thể bằng đồ thị với các đơn thể là các đỉnh và các cung là dòng điều khiển thì đồ thị đó phải là đơn giản nhất. Theo [10], phương pháp thiết kế theo cấu trúc cho phép giảm độ phức tạp của hệ chương trình (theo nghĩa cấu trúc) bằng cách chia hệ thống thành các đơn thể chức năng đơn giản, độc lập và có thể sử dụng lại được.

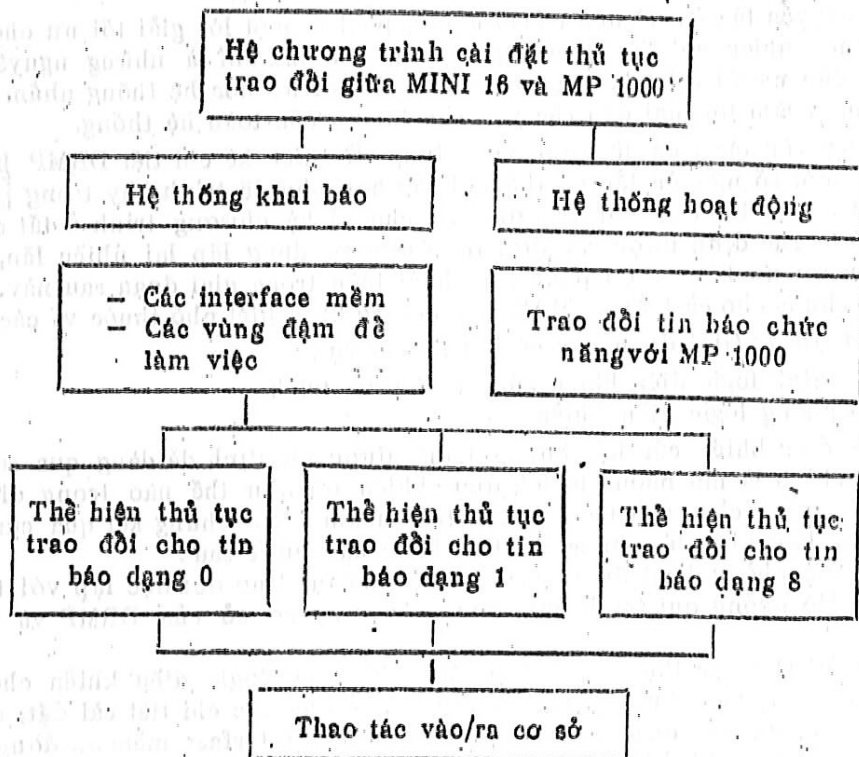
Mức phân cấp
Vấn đề cần giải quyết
0

Đối tượng của hệ chương trình
1

Chức năng
2

Chức năng
3

Chức năng
4



Hình 2. Sơ đồ cấu trúc tổng quát của hệ chương trình DRMP

Vấn đề nảy sinh là phân chia thành các đơn thể theo cách nào và theo các tiêu chuẩn nào. Vấn đề này đã được lý thuyết về lập trình theo cấu trúc đề cập đến nhiều, song cho đến nay vẫn chưa có một phương pháp chung nào và càng chưa thể nói đến « một phương pháp tối ưu » (nếu có) cho việc phân chia hệ thống thành các đơn thể. Dựa vào [10] chúng tôi đã đề ra một số nguyên tắc được tuân theo khi thiết kế chi tiết nhằm thực hiện tối ưu hóa cấu trúc hệ mềm:

(NT1) Cực tiểu hóa số liên hệ giữa các đơn thể bằng cách cực tiểu hóa mối quan hệ giữa các phần tử không ở trong cùng một đơn thể, trong đó, một liên hệ được hiểu là một sự qui chiếu đến một nhân nào đó, hoặc một địa chỉ được xác định ở nơi nào đó và phần tử được hiểu là một phần của đơn thể, ví dụ một lệnh, một nhánh hoặc một hàm con.

(NT2) Cực đại hóa mối quan hệ giữa các phần tử trong cùng một đơn thể bằng cách tăng « độ ràng buộc » (binding) của nó, trong đó độ ràng buộc là độ đo tính co kết (cohesion) của một đơn thể.

(NT3) Các mức phân cấp cao nhất của hệ chương trình được xác định theo phương pháp của Wirth [16].

(NT4) Các mức phân cấp thấp hơn được xác định theo cách của PARNAS [6].

Ta dễ dàng nhận thấy rằng, việc thực hiện triệt để các nguyên tắc (NT1) và (NT2) sẽ làm giảm độ phức tạp cấu trúc hệ thống.

Vấn đề tồn tại là thực hiện quá trình phân chia thành các đơn thể đến khi nào? Để có được một lời giải cho vấn đề đó, ta thực hiện các nguyên tắc (NT3) và (NT4), theo [9] nếu chỉ thực hiện theo nguyên tắc của Wirth sẽ có điểm bất lợi là làm tăng độ liên kết giữa các chức năng, mặt khác nếu chỉ thực hiện theo nguyên tắc của Parnas sẽ dẫn đến việc xuất hiện quá nhiều đơn thể và như vậy sẽ tăng số liên hệ giữa chúng, điều đó cũng làm tăng độ phức tạp cấu trúc. Thực hiện kết hợp các nguyên tắc (NT3) và (NT4) ta có thể hạn chế các nhược điểm trên. Vấn đề tiếp theo chúng ta xét là tối ưu đồng thời các chỉ tiêu về giảm độ phức tạp thuật toán.

Một thực tế là nếu chúng ta chỉ thực hiện đơn thuần tối ưu hóa về cấu trúc thì kết quả thường dẫn đến việc tăng độ phức tạp thuật toán, đó là giảm tính hữu hiệu của hệ chương trình. Cho đến nay cũng chưa có phương pháp chung nào giải quyết vấn đề đó. Để có một lời giải « dung hòa » cho các yêu cầu tối ưu đồng thời, ta bổ sung các nguyên tắc sau:

(NT5) Xác định các đơn thể được dùng đến nhiều lần để tập trung tối ưu hóa độ phức tạp thuật toán, đặc biệt nhằm giảm thời gian thực hiện.

(NT6) Xác định các đơn thể ít được dùng đến để có thể lưu vào bộ nhớ phụ, khi cần mới gọi vào bộ nhớ chính.

Các nguyên tắc (NT1) đến (NT6) không phải là một lời giải tối ưu cho bài toán tối ưu đồng thời theo nhiều chỉ tiêu theo nghĩa toán học mà chỉ là những nguyên tắc mang tính chất hướng dẫn người thiết kế thực hiện ở bước tối ưu hóa hệ thống nhằm tránh được việc tối ưu bộ phận làm tổn hại đến các yêu cầu tối ưu của toàn hệ thống.

Các nguyên tắc trên đã được vận dụng để thiết kế chi tiết DRMP bằng cách đưa ra và tuân thủ một số nguyên tắc cụ thể, chi tiết hơn như đã trình bày trong [15]. Kết quả nhận được là một sơ đồ thiết kế chi tiết đầy đủ cho cả hệ chương trình ở tất cả các mức (xem [14]). Từ đó ta xác định được các đơn thể được sử dụng lặp lại nhiều lần, đó là đối tượng để thực hiện tối ưu hóa về độ phức tạp thuật toán trong giai đoạn sau này. Về cơ bản, thiết kế này dùng chung cho cả DĐS và MAS, trừ một số khác biệt nhỏ thuộc về cách cài đặt cụ thể.

Giai đoạn 3: Giai đoạn này có hai nhiệm vụ:

- 1) Xác định logic điều khiển của hệ chương trình.
- 2) Mô phỏng logic điều khiển.

Logic điều khiển của hệ chương trình được xác định dễ dàng qua sơ đồ thiết kế chi tiết. Vấn đề chính là mô phỏng logic điều khiển đó như thế nào trong điều kiện chưa có ngôn ngữ mô phỏng cho thiết kế theo cấu trúc nhằm kiểm chứng kết quả của giai đoạn thiết kế chi tiết. Để làm việc đó, chúng tôi tiến hành các bước sau:

(MP1) Mô phỏng hoạt động của các loại thủ tục trao đổi độc lập với thiết bị MP1000.

(MP2) Mô phỏng qui trình hoạt động vào - ra cơ sở của DRMP và thử nghiệm với thiết bị.

Bước (MP1) được thực hiện bằng cách thể hiện logic điều khiển cho các loại logic điều khiển bằng một hệ chương trình, trong đó bỏ qua các chi tiết cài đặt, chỉ lưu tâm đến dòng điều khiển giữa các đơn thể, mối liên hệ với các interface mềm ... đồng thời các thông tin nhận được từ thiết bị thật được mô phỏng bằng chương trình. Hệ chương trình mô phỏng

này được viết ở ngôn ngữ Assembler và do vậy được chạy dưới sự giám sát của chương trình hiệu chỉnh DEB (debuging processor) trong DỚS, cho phép dừng tại mọi điểm và kiểm tra từng đoạn hoạt động của hệ chương trình. Bước này, cũng như toàn bộ giai đoạn thiết kế tổng thể và chi tiết được thực hiện độc lập với thiết bị (trước khi có thiết bị) trên cơ sở tài liệu mô tả kỹ thuật và hướng dẫn sử dụng. Kết quả là logic điều khiển cho cả hệ chương trình (phần không liên quan đến thiết bị) đã được kiểm tra. Việc cài đặt dòng điều khiển ở đây cũng sẽ được sử dụng trực tiếp cho việc cài đặt cả hệ chương trình sau này.

Bước (MP2) được thực hiện như sau: Sau khi thiết bị đã được ghép nối với MINI16 và đã được kiểm nghiệm hoạt động, chúng tôi xây dựng qui trình vào - ra giống như qui trình sẽ cài đặt sau này vào DỚS, điểm khác duy nhất là ở chỗ các lệnh vào - ra được dùng trực tiếp, còn trong hệ điều hành thì nó được xây dựng bởi một thư viện để dùng chung cho cả hệ. Hệ chương trình này chạy trong DỚS, hoạt động trực tiếp với thiết bị thông qua MCU3 với cả hai kênh truyền tin (programmed channel và IOP). Các lệnh dừng (HLT) của MINI 16 được cài vào các điểm thích hợp để có thể kiểm tra logic điều khiển. Sau khi hoạt động, hệ chương trình mô phỏng này được sử dụng để thử nghiệm hoạt động của từng loại thủ tục. Kết quả cho phép một khẳng định được hoạt động thực tế của thủ tục trao đổi như hướng dẫn, mặt khác chỉ ra được một số sai khác giữa hoạt động thực với tài liệu mô tả và chỉ ra một số chức năng không có trong hình trạng thật của thiết bị, giúp ích cho giai đoạn cài đặt cụ thể. Nhiều đơn thể có thể dùng lại trong hệ DRMP.

3. GIẢI PHÁP CÀI ĐẶT HỆ CHƯƠNG TRÌNH DRMP.

Nói một cách tổng quát, nhiệm vụ của giai đoạn này là tạo ra được một hệ chương trình hoàn chỉnh theo thiết kế, được cài vào hệ DỚS (MAS) mà thông qua đó người sử dụng có thể điều khiển thiết bị MP1000 như đối với các ngoại vi thông dụng khác. Giải pháp cài đặt phải đạt được các yêu cầu sau:

(CĐ1) Thể hiện đầy đủ và chính xác các thủ tục trao đổi cho các dạng tin báo của ngôn ngữ trao đổi MP1000.

(CĐ2) Hoạt động đồng bộ trong hệ quản lý vào - ra của DỚS (MAS).

(CĐ3) Tối ưu hóa độ phức tạp thuật toán.

Các yêu cầu (CĐ1) và (CĐ2) đã được thực hiện trên cơ sở giải pháp chung đã được xây dựng bởi nhóm «Lập trình hệ thống» trên máy MINI16 nhằm cài các chương trình điều khiển ngoại vi chuyên dụng vào các hệ điều hành DỚS (MAS) bằng cách tận dụng hệ quản lý vào - ra có sẵn của các hệ điều hành (xem [1], [2], [4]). Một vài đặc thù của DRMP đã được trình bày trong [13].

Yêu cầu (CĐ3) đã được thực hiện như sau:

(TU1) Tận dụng tối đa các chức năng dùng chung có sẵn.

(TU2) Tận dụng tối đa khả năng của thiết bị. Ở bước 1, đã sử dụng kênh truyền tin «programmed channel» với tốc độ 9600 baud. Sau đó chuyển sang tận dụng kênh truyền tin «IOP», trong đó các bước của hoạt động vào / ra đã được «cứng hóa», cho phép tốc độ truyền tin giữa bộ nhớ với MCU3 là 1660 Kb/s. Các kết quả bước đầu cho thấy thời gian thực hiện các thủ tục cho các tin báo chức năng cơ bản giảm xuống gần một nửa. Ngoài ra ta còn giảm được 4 đơn thể với 130 dòng lệnh Assembler với 66 ô nhớ.

(TU3) Tận dụng tối đa các kinh nghiệm lập trình Assembler và sử dụng bộ lệnh MINI16. Các kinh nghiệm này đã được đúc rút bởi nhóm «Lập trình hệ thống» trên MINI16 và được vận dụng để thực hiện tối ưu độ phức tạp thuật toán cho các đơn thể có tần số sử dụng lại cao. Hệ chương trình DRMP đã được cài đặt vào DỚS, sau khi đã thực hiện tối ưu hóa còn khoảng 2000 dòng lệnh, chiếm một dung lượng bộ nhớ \approx 4Kb.

4. KẾT LUẬN

Hệ chương trình DRMP đã hoạt động đồng bộ trong hệ DỚS và cho kết quả mong muốn. Đối với hệ MAS theo cách thiết kế như trình bày ở mục 2, phần lớn công việc cũng đã được thực hiện, vấn đề còn lại là bổ xung một số chi tiết cài đặt và cho hoạt động thử nghiệm đồng hợp đánh giá sự đồng bộ với hoạt động của hệ vào/ra trong MAS. Ở đây cũng có thể tận dụng các kinh nghiệm cài đặt của các ngoại vi chuyên dụng khác [3] [4]. Tư tưởng của phương pháp thiết kế theo cấu trúc, thực chất, như tác giả của nó đã khẳng định trong [10] là không mới, bởi vì trong các lĩnh vực kinh tế và kỹ thuật khác người ta đã áp dụng nó từ lâu [9]. Tuy vậy, trong lĩnh vực kỹ thuật phần mềm, một lĩnh vực gắn liền với tư duy con người, thì đây có thể là chìa khóa cho các nghiên cứu tiếp theo sâu sắc hơn. Đóng góp

của chúng tôi ở đây là chi tiết hóa phương pháp thiết kế hệ mềm theo cấu trúc, vận dụng thử nghiệm cho đối tượng cài đặt cụ thể là hệ chương trình điều khiển terminal vi tính chuyên dụng và phát triển phương pháp này, đặc biệt trong phương diện đưa tư tưởng tối ưu hóa đồng thời hai chỉ tiêu về độ phức tạp cấu trúc và độ phức tạp thuật toán một cách xuyên suốt trong toàn bộ quá trình phát triển hệ mềm và đưa ra các nguyên tắc để thực hiện quá trình tối ưu đó.

Xin chân thành cảm ơn các đồng nghiệp trong nhóm « Lập trình hệ thống » trên máy MINI16 đã có những trao đổi quý giá về giải pháp cài đặt DRMP. Xin cảm ơn các đồng nghiệp trong Seminar « Đồng bộ hóa » (Viện Khoa học Tính toán và Điều khiển) đã có những góp ý và gợi ý bổ ích để hình thành nên bài báo này.

Nhận ngày 1-4-1985

TÀI LIỆU DẪN

1. N.N. Kiên, Cài đặt chương trình điều khiển thiết bị biến đổi ảnh-số vào hệ điều hành D ϕ S của máy tính MINI16. - Báo cáo khoa học Đơn vị 40208 1980.
2. N.N. Kiên, L.N. Đẩu, Đ.M. Thái, Phát triển các chức năng vào-ra chuyên dụng cho một hệ điều hành thời gian thực nhiều mục đích. - Báo cáo đăng ký Hội nghị Toán học Việt Nam lần thứ 3, 1985.
3. N.N. Kiên, Về giải pháp cài đặt chương trình điều khiển bộ biến đổi ảnh-số vào hệ điều hành MAS của máy tính MINI16. - Đăng ký báo cáo khoa học, Đơn vị 40208, 1984.
4. L.N. Đẩu, Về giải pháp cài đặt chương trình điều khiển bộ biến đổi điện-số vào hệ điều hành MAS của máy tính MINI 16. - Báo cáo khoa học, Đơn vị 40208, 1982.
5. Nixdorf E., Von Hardware zur Software- ein Biedeutungs- wandel. - Applied Informatics 2(1982).
6. Parnas D.L., A technique for Softwaremodule Specification with exemple. - CACM 15, 5(1972).
7. Schulz A., Software- Entroufsmethoden: Ein Vergleich des LIT ϕ S-Verfahren mit der Jackson-Methode. - Applied Informatics 1 (1979).
8. Schulz A., Methoden des Softwareentwurfs iund strukturierte Programmierung. - Walter de Gruyter Verlag, Berlin, NewYork 1979.
9. Schulz A., Untersuchungen zur Anwendbarkeit der strukturierten Programmierung in der Betriebsinformatik Applied Informatics 8 (1977).
10. Steven W.P., Meyers G.I., and Constantine L.L., Structured Design. - IBM System Journal, Vol. 13 (1974); pp. 115-139.
11. Herrlich O., Lindner W., Strukturierte Programmierung-BSB B.G. Teubner Verlagsgesellschaft, Leipzig, 1981.
12. Terminal de visualisation MP1000, Vol. I, II, III.
13. Đỗ Minh Thái, Đ.T Vĩnh, Về một giải pháp ghép nối MTĐT MINI16 với hệ vi tính chuyên dụng MP1000. - Nội san KHKH, 2 (1983), Đơn vị 40208.
14. Đỗ Minh Thái, Thiết kế và cài đặt hệ chương trình điều khiển hệ vi tính chuyên dụng MP1000 vào hệ điều hành của MTĐT MINI 16. Tài liệu thiết kế, 1983, Đơn vị 40208.
15. Đỗ Minh Thái, Phương pháp thiết kế và cài đặt hệ chương trình điều khiển hệ vi tính chuyên dụng MP1000 vào hệ điều hành của MTĐT MINI 16. - Báo cáo khoa học, Hội thảo khoa học về tin học 10-1983.
16. Wirth N., Program development by stepwise refinement. - CACM14, 4 (1971).
17. Zygment Ryznar S & DL - Structured Design Language (Proposal). Applied Informatics 12 (1981).

ABSTRACT

Design and Implementation of the program system controlling the special purpose microcomputing terminal MP 1000 to the operating system of the computer MINI 16

The paper deals with the problem of design and implementation of the program system controlling the special purpose microcomputing terminal MP 1000 to the operating system of the computer MINI 16 on the basis of the application and of the extension of a software design: methode for fulfilling (in some sense) simultaneously the following criteria:

- « optimization » of system structure complexity.
- « optimization » of algorithmic complexity.