

Mục lục

Chương I:	3
tổng quan về hệ điều hành MS-DOS	3
I. Khái niệm hệ điều hành.....	3
II. Đặc điểm hệ điều hành MS - DOS	3
III. Cấu trúc của hệ điều hành MS – DOS.....	3
IV. Ưu điểm và hạn chế của hệ điều hành MS – DOS.....	4
1. Ưu điểm.....	4
2. Hạn chế.....	4
Chương II:.....	6
Các thành phần của hệ điều hành MS - DOS	6
I. Quản lý tiến trình.....	6
1. Khái niệm quá trình.....	6
2. Thực hiện quá trình.....	6
3. Quản lý tiến trình.....	6
Làm việc với các quá trình.....	7
II. Quản lý bộ nhớ ngoài.....	11
Các driver khôi	12
III. Hệ thống bảo vệ.....	28
1. Dùng phục vụ năm(format track đĩa) để bảo vệ chống sao chép: ...	29
2. Phục vụ 90 HEX: (AT) Thiết bị bận:	30
3. Undelete: Lệnh được sử dụng khi muốn cứu các tập tin bị xoá. ..	31
4. Defrag: Lệnh được sử dụng để sắp xếp lại khối lưu trữ của tập tin, tránh tình trạng phân mảnh đĩa:	31
5. Scandisk: Lệnh dùng để phân tích, sửa chữa các lỗi về đĩa hay tập tin, thư mục.	32
7. SHOWMEM.ASM	32
IV. Quản lý bộ nhớ trong.....	33
V. Quản lý file	37
1. File là gì?.....	38
FILENAME.EXT	38
2. Quản lý File.....	39
VI. Hệ THỐNG VÀO RA.....	41
1. Thiết bị nhập dữ liệu.....	41
2. Thiết bị đầu ra.....	44
VII. Hệ thông dịch.....	48

Lời mở đầu

Máy tính đã trải qua một chặng đường dài trong một thời gian tương đối ngắn và phần lớn tiến trình này là nhờ vào những tiến bộ trong công nghệ hệ điều hành. Suốt 30 năm qua, sự tiến hoá trong hệ điều hành làm cho các máy tính dễ sử dụng, dễ hiểu hơn, linh hoạt hơn và đáng tin cậy hơn. Trong những năm 80, hầu như mọi máy tính PC đều chạy hệ điều hành DOS và các máy tính Macintosh sử dụng Mac DOS. Ngày nay, người sử dụng có nhiều khả năng lựa chọn hệ điều hành hơn. Nhưng việc lựa chọn này không phải lúc nào cũng dễ dàng. Mặc dù, phần lớn các máy tính thế hệ mới đều có một phiên bản tiêu dùng của Window, nhưng số lượng các ngoại lệ tăng dần.

Sau hệ điều hành Unix, ban đầu được phát triển bởi Bell Labs vào những năm 70 MS-DOS (Microsoft Disk Operatinh System) của Microsoft cùng với PC-DOS (Personal Computer Disk Operatinh System) của IBM, đã từng là hệ điều hành phổ biến nhất trong số tất cả các hệ điều hành PC. Có một số lượng rất lớn các phần mềm chạy trong DOS, và có rất nhiều hệ thống PC dựa trên Intel chạy DOS.

DOS được phát triển trong những năm 70 và được phân bố trên một số máy PC thương mại thời kì đầu. Tuy thống trị suốt những năm 70 nhưng DOS phải đấu tranh rất nhiều để giành được ưu thế. Đối thủ khó chịu nhất của nó là một hệ điều hành có tên là CP/M (Control Program for Microprocessor). Tuy nhiên, DOS đã chiến thắng trong những cuộc chiến giành thị trường hệ điều hành thời kì đầu vì IBM đã chọn và cấp phép cho nó thay vì CP/M như là một hệ điều hành chuẩn dành cho IBM PC. Nó trở hệ điều hành dành cho thị trường khổng lồ của các máy tính tương thích IBM.

Chương I: Tổng quan về hệ điều hành MS-DOS

I. Khái niệm hệ điều hành.

Hệ điều hành là cầu nối trung gian giữa các chương trình với phần cứng máy tính. Bởi vì, khi chạy chương trình chúng ta cần sử dụng bộ nhớ, bộ hiển thị, các ổ đĩa và các thiết bị khác của máy tính chẳng hạn như máy in. Ngoài ra, trong mạng máy tính hệ điều hành đóng vai trò trung gian giữa máy tính cá nhân với các thiết bị khác trên mạng. Có thể nói hệ điều hành là hệ chương trình điều khiển mọi hành động của máy tính điện tử. Hệ điều hành thực hiện chương trình từ một trong ba nguồn sau:

- Điều khiển các bộ phận của máy tính (vi xử lý, bộ nhớ, cửa vào/ra) do thực hiện chương trình trong các tệp chứa trên đĩa.

- Làm việc theo lệnh điều khiển của người điều khiển máy truyền lệnh qua bàn phím.

- Làm việc theo yêu cầu của thiết bị ngoài (yêu cầu ngắt chương trình bằng tín hiệu – ngắt cứng) hoặc bởi lệnh ngắt chương trình INT nh viết trong chương trình – ngắt mềm.

II. Đặc điểm hệ điều hành MS - DOS .

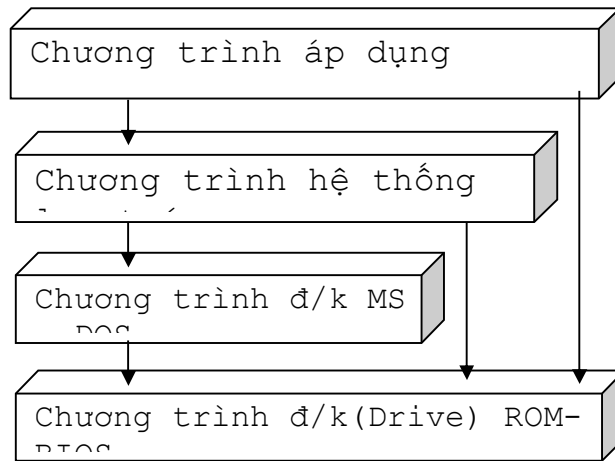
MS – DOS ra đời trên cơ sở của CP/M để phục vụ cho vi xử lý 16 bit, bắt đầu từ Intel 8088.

Cp/M (Control Program for Microcomputer) là hệ điều hành đầu tiên cho vi xử lý 8 bit loại 8085, ghi trên đĩa mềm do Gary kindall (1974) sau khi vi xử lý Intel ra đời (1973). Sau đó CP/M được cải tiến cho vi xử lý 8085 (1977) và Z80 (1976).

MS – DOS ra đời năm 1978 do hãng Microsoft viết lại trên cơ sở CP/M 86. Đến năm 1982, MS - DOS được viết lại có cải tiến cho các vi xử lý từ 8088, 8086. hệ viết trên hợp ngữ 8086. Hệ có khả năng vẽ đồ thị, xử lý văn bản, quản lý tệp, quản lý cơ sở dữ liệu(Database II).

III. Cấu trúc của hệ điều hành MS – DOS.

Bởi vì MS - DOS có cấu trúc theo lớp nên các giao diện và các chức năng của chúng không được phân chia rõ ràng. Ví dụ các chương trình áp dụng có khả năng truy cập đến các chương trình con vào/ra cơ sở nhằm viết trực tiếp lên màn hình và hiện lên trên đĩa.



Cấu trúc theo lớp

IV. Ưu điểm và hạn chế của hệ điều hành MS – DOS.

1. Ưu điểm.

Trong số những sức mạnh của MS - DOS gồm có độ tin cậy cao và tính ổn định. Trên một hệ thống được cấu hình đúng đắn DOS và các chương trình của DOS chạy rất tốt, những sự cố sập đổ và treo rất hiếm khi xảy ra. Hầu hết những người sử dụng chỉ cần học một tập lệnh nhỏ. Tuy dấu nhắc của DOS không phải là một giao diện tao nhã, nhưng nó không khó sử dụng lắm một khi mà người sử dụng đã thành thạo các lệnh dùng thường xuyên.

DOS thoả đáng cho các máy PC tương thích IBM trong đầu những năm 80 nhưng những giới hạn của nó đã trở lên đáng chú ý hơn khi máy PC trở lên mạnh mẽ hơn.

2. Hạn chế.

Vì DOS có cấu trúc theo lớp nên nó có bộ mặt dễ bị tổn thương ở các chương trình bị lỗi, điều này gây ra sự che chắn toàn bộ hệ hay xoá đĩa khi những chương trình của người dùng không chạy.

Trong DOS, người sử dụng chỉ có thể nạp mỗi lần một chương trình đơn vào bộ nhớ, muốn làm việc với chương trình thứ hai phải đóng chương trình đầu lại – một tiến trình thường gây trở ngại cho năng suất.

DOS chỉ hỗ trợ một người sử dụng và bộ xử lý đơn.

Vì DOS không áp đặt giao diện của một ứng dụng phải trông ra sao và hoạt động như thế nào, nên nhiều nhà phát triển đã tạo ra rất nhiều

giao diện đặc trưng chương trình. Một số ứng dụng xuất hiện như các màn hình thuần văn bản, người dùng phải ghi nhớ các cú nhấn phím hoặc phải sử dụng các phím chức năng. Các ứng dụng khác cố gắng sử dụng các hệ thống menu thô sơ vốn gây khó khăn cho những người dùng DOS không có một con chuột.

DOS được thiết kế chỉ để nhận diện được 640 Kb RAM do vậy không thể quản lý những khối lượng RAM lớn hơn thường được tìm thấy trên các máy PC ngày nay. Kết quả là phải sử dụng các tiện ích để truy suất bộ nhớ vượt quá giới hạn 640 Kb.

DOS được thiết kế cho các CPU 8 bit và 16bit, nó có thể lợi dụng kiến trúc 32 bit của các chip 486, Pentium và các chip mới hơn. DOS buộc các máy tính có hiệu suất cao phải hoạt động ở tốc độ thấp hơn khả năng của chúng.

Phần cứng khó lắp đặt và định cấu hình trong DOS vì mỗi thiết bị đòi hỏi một chương trình điều khiển đặc trưng. Thông thường, các ứng dụng của DOS khác nhau sử dụng các trình điều khiển khác nhau cho cùng một thiết bị.

Các tệp tin của DOS giới hạn bởi 8 kí, cộng thêm một “phần mở rộng” gồm 3 kí tự sau dấu chấm. Trong khi đó Window95, 98 hỗ trợ các tệp tin bao gồm 256 kí tự.

Chương II:

Các thành phần của hệ điều hành MS - DOS .

I. Quản lý tiến trình.

1. Khái niệm quá trình.

Quá trình là sự biến đổi trạng thái dưới sự tác động của chương trình.

2. Thực hiện quá trình.

Một số tài nguyên nhất định như: bộ nhớ thiết bị phần cứng, giờ CPU... nên khi một quá trình hoạt động nó sẽ đòi hỏi 2 điều kiện cơ bản là:

- + Những yêu cầu tối thiểu để quá trình thực hiện.
- + Phải đảm bảo điều kiện để thực hiện song song các quá trình.

3. Quản lý tiến trình.

Khi có nhiều quá trình hoạt động song song, do đó có thể nảy sinh các quá trình con, việc quản lý quá trình phải thực hiện các công việc sau:

a. Tạo và xoá bỏ các quá trình của user và của hệ thống.

MS - DOS chỉ thị dòng lệnh bằng cách thực hiện trên màn hình dấu nhắc lệnh. Bạn gõ các lệnh ở dấu nhắc lệnh để qui định nhiệm vụ mà bạn muốn MS – DOS làm. Mỗi lệnh bao gồm một tập các chỉ thị. Để yêu cầu MS – DOS thi hành một lệnh, bạn gõ lệnh sau đó bấm enter.

Gõ một lệnh:

Dấu vạch sáng nhấp nhấp nháy mé dưới dòng lệnh được gọi là dấu nhắc. Dấu nhắc chỉ cho bạn biết nơi phải gõ lệnh. Khi bạn gõ 1 kí tự vào, dấu nhắc sẽ di chuyển sang khoảng trống bên phải. Nếu gõ nhầm, bấm phím Backspace để xoá kí tự sát bên trái dấu nhắc. Nếu bạn muốn gõ lại một lệnh, hãy bấm ESC. Dấu nhắc sẽ di chuyển đến đầu dòng kế tiếp và bạn có thể bắt đầu lại.

MS – DOS có những phím hiệu chỉnh để lặp lại hoặc thay đổi 1 lệnh mà bạn đã gõ trước đó như:

F1: Hiện thị lệnh trước đó, bấm mỗi lượt ra 3 kí tự.

F3: Hiện thị toàn bộ lệnh trước đó(một lượt).

MS – DOS còn có chương trình DOSKey để truy tìm, thay đổi và dùng lại các lệnh. Có nhiều cách để truy tìm lại các lệnh này khi đã nạp DOSKey nhưng cách dễ nhất là bấm phím mũi tên lên.

Cách thức MS-DOS đáp ứng một lệnh

MS – DOS đáp ứng một lệnh theo nhiều cách khác nhau. MS – DOS có thể thông báo cho bạn biết là lệnh bạn yêu cầu đã thực hiện hoàn tất, hay có lỗi vì bạn gõ sai lệnh. Đôi khi MS – DOS nhắc bạn soát lại một lệnh.

Ngoài ra MS – DOS có sẵn những chỉ dẫn hỗ trợ cho mọi lệnh để bạn tra cứu. Để thêm thông tin chỉ dẫn bạn gõ tên lệnh theo sau là /?, hoặc gõ HELP theo sau là tên lệnh. Nếu bạn gõ HELP mà không có lệnh theo sau, MS – DOS sẽ hiện thị danh sách mọi lệnh MS – DoS và công dụng của chúng.

b. Ngừng, huỷ bỏ một qua trình

Bạn có thể ngừng kết xuất của một lệnh khi đang thực hiện, bằng cách bấm tổ hợp phím CTRL+S hoặc PAUSE. Sau đó, bấm bất kì phím nào trừ phím PAUSE để tiếp tục xuất thông tin. Bạn có thể tuỳ ý lặp lại nhiều lần việc tạm ngưng và tiếp tục xuất tin như vậy

Còn khi muốn ngăn MS – DOS hoàn tất 1 lệnh thì hãy bấm CTRL+BREAK hay CTRL+C. Khi đó lệnh sẽ thôi dùng dấu nhắc đợi lệnh lập tức xuất hiện.

Làm việc với các quá trình

Một trong những tính năng của MS – DOS là khả năng nó có thể chạy cùng một lúc 2 hay nhiều chương trình hay nhiều hơn. Khi bạn chạy các chương trình cùng một lúc, chúng sẽ dùng các tài nguyên ở máy của bạn theo nhiều cách khác nhau.

MS – DOS Shell cũng là một công cụ để kết cấu các chương trình thành các nhóm mục chương trình. Khi bạn thêm 1 chương trình vào nhóm, bạn có thể đặc tả thông tin mà chương trình ấy sẽ dùng đến, mỗi khi bạn gọi nó.

+ *Gọi thực hiện một chương trình*

Có 4 cách để gọi thực hiện một chương trình

- Nếu một chương trình được liệt kê trong một nhóm có mặt trong danh sách chương trình thì chọn một mục chương trình.

- Bạn có thể gọi thực hiện 1 chương trình trong danh sách tệp tin bằng cách chọn tệp tin hay một tệp tin kết với chương trình đó

- Gọi thực hiện chương trình bằng cách dùng lệnh RUN từ menu File và gõ tệp tin chương trình vào.

- Một cách khác nữa là làm việc tại dấu nhắc đợi lệnh

+ *Chuyển đổi giữa các chương trình.*

Bạn có thể chạy cùng 1 lúc nhiều chương trình và dễ dàng chuyển đổi giữa chúng bằng cách cho chạy đặc phận Task Swapper.

- Chạy nhiều chương trình

- Gọi chạy chương trình đầu tiên bằng cách bấm đúp tại tên chương trình của nó ở danh sách tệp tin hay ở danh sách chương trình. Hoặc bấm phím mũi tên lên, xuống để chọn tên chương trình rồi bấm enter. Chương trình sẽ xuất hiện trên màn hình của bạn.

- Bấm CTRL+ESC để trở lại MS – DOS Shell. Tên chương trình mà bạn vừa mới gọi thực hiện sẽ xuất hiện trên Active Task list.

- Gọi chạy chương trình khác, chương trình thứ 2 sẽ xuất hiện trên màn hình, mặc dù bạn chưa rời bỏ chương trình thứ nhất.

- Thêm 1 chương trình và Active task list.

- Từ danh sách tệp tin hay danh sách chương trình, hãy chọn tên tệp tin chương trình mà bạn muốn thêm và Active Task list

- Bấm và giữ phím SHIFT, rồi bấm đúp tại tên chương trình hoặc bấm SHIFT+ENTER

- Để chuyển tới chương trình khác, từ MS – DOS Shell.

Bấm đúp tại tên chương trình trên Active Task list hoặc bấm phím mũi tên để chọn chương trình mà bạn muốn chuyển qua, rồi bấm ENTER.

- Xoay vòng các chương trình trong Active Task list từ MS – DOS Shell.

- Bấm và giữ phím ALT, trong khi bấm TAB, chương trình đầu tiên trong Active Task list sẽ xuất hiện trên màn hình. Nếu bạn muốn chuyển

sang 1 chương trình nằm xa mé dưới danh sách Active Task list thì tiếp tục nhấn phím ALT và bấm phím TAB. Sau khi chương trình đầu tiên xuất hiện, chỉ có tên chương trình kế tiếp mới hiện ra trên đỉnh màn hình. Để chọn một chương trình bạn nhả phím ALT ra.

- Bấm và giữ phím ALT trong khi bấm phím TAB. Tên của chương trình kế tiếp trong Active Task list sẽ xuất hiện trên đỉnh màn hình. Để chuyển sang chương trình khác, dừng nhả phím ALT mà chỉ tiếp tục bấm phím TAB cho đến khi chương trình bạn muốn chuyển qua được hiện ra. Để chọn chương trình ấy hãy nhả phím ALT.

- Để chuyển đến MS – DOS Shell từ bất cứ chương trình nào.

- Bấm phím CTRL+ESC.

- Hoặc giữ phím ALT trong khi bấm phím TAB nhiều lần cho đến khi chữ MS – DOS Shell hiện lên ở đỉnh màn hình. Để trở về MS – DOS Shell, bạn nhả phím ALT ra.

- Kết thúc một chương trình

Khi đặc phận Task Swapper đang được phép chạy, thì bạn có thể chạy đồng thời nhiều hơn 1 chương trình. Các chương trình bạn gọi thi hành đều được liệt kê trong Active Task list bạn phải rời bỏ khỏi chương trình này.

- Để rời bỏ một chương trình có trong danh sách Active Task list .+Từ MS – DOS Shell chuyển sang chương trình bạn muốn.

- + Dùng lệnh Exit
- Để rời bỏ một chương trình
- + Chuyển sang MS – DOS Shell.
- + Từ Active Task list chọn chương trình bạn muốn rời bỏ
- + Từ menu file, chọn delete hay bấm phím DEL.

c. *Cơ chế đồng bộ các quá trình*

Nếu bạn có một tập tin thường sử dụng cùng 1 chương trình cụ thể nào đó, bạn có thể tiết kiệm được thời gian bằng cách kết các tập tin với chương trình. Khi đó, mỗi lần bạn mở tập tin đã được kết, chương trình sẽ khởi động, kèm theo nạp các tập tin đó vào

- Kết các tập tin với một chương trình
 - Chọn thư mục có chứa chương trình bạn muốn kết với một kiểu tập tin nào đó.
 - Từ danh sách tập tin, bạn chọn tên tập tin của chương trình ấy
 - Từ menu file, chọn Associate, hộp đối thoại Associate file sẽ xuất hiện
 - Trong hộp Extensions(tên mở rộng) bạn gõ tên mở rộng của tập tin mà bạn muốn kết với chương trình vừa chọn
 - Chọn nút OK.

Chú ý: Bạn có thể kết tập tin với chương trình bằng cách chọn trước một tập tin và sau đó mới chỉ định tên chương trình

- Chạy một tập tin đã được kết nối với 1 chương trình khác
 - Từ menu file, chọn RUN, hộp đối thoại RUN sẽ xuất hiện
 - Gõ vào tên đường dẫn và tên của tập tin của chương trình mới mà bạn muốn dùng, theo sau là tên tập tin đã kết.
 - Chọn nút OK
- Gỡ bỏ 1 liên kết giữa 1 chương trình và một kiểu tập tin
 - Chọn tập tin có liên kết mà bạn muốn gỡ bỏ
 - Từ menu file, chọn Associate, hộp đối thoại Associate File sẽ xuất hiện. Tên chương trình được hiển thị trong hộp văn bản
 - Bấm phím BACKSPACE để xoá tên chương trình
 - Chọn nút OK hoặc nhấn ENTER

II. Quản lý bộ nhớ ngoài.

Rất nhiều công việc mà DOS thực hiện thay cho người sử dụng lại không được người sử dụng thấy, do vậy, người sử dụng thường dễ đánh giá thấp sức mạnh và tính phức tạp của hệ điều hành. Thí dụ: Một số lớn các cấu trúc dữ liệu mà DOS dùng để quản lý bộ nhớ khối như đĩa cứng hay đĩa mềm không được người sử dụng nhìn thấy. Mặc dù sự am hiểu các cấu trúc này không có ý nghĩa thực tế đối với đại đa số người sử dụng, nhưng chúng tôi sẽ cho các bạn thấy các bí mật của DOS, sẽ giới thiệu cho các bạn cấu trúc và hoạt động của các cấu trúc dữ liệu quan trọng.

Dưới con mắt của người sử dụng, DOS gọi các bộ nhớ khối như là gọi các volume, một chữ cái đặc biệt được gán cho mỗi volume. Đối với các ổ đĩa mềm, đó là các chữ A, B. Trong khi đó, đĩa cứng thường được ký hiệu bằng các chữ C. Một bộ nhớ khối có thể có nhiều volume (ở đây ta nói tới sự phân chia đĩa). Tất nhiên, sự phân chia một đĩa mềm thành nhiều volume chẳng có ý nghĩa gì lớn vì dung lượng của nó tương đối nhỏ. Ngược lại, ta có thể chia đĩa cứng thành nhiều volume. Thí dụ: Như khi ta muốn làm việc không chỉ dưới hệ điều hành DOS mà còn dưới unit(xenix). Mỗi hệ điều hành này có thể có nhiều volume riêng của mình.

Mỗi volume có thể có tên, tuy nhiên điều này không bắt buộc. Tên volume được hiện trong DIR, tên volume được đặt trong khi tạo khuôn(format) volume. Mỗi volume còn có một thư mục gốc, thư mục này chứa nhiều thư mục con và các file. Một loạt các hàm được gọi qua ngắt 21h và những hàm cho phép người sử dụng truy nhập đến các File và các thư mục con. Khi ta gọi các hàm này, ta không cần phải quan tâm đến các đặc tính của bộ nhớ khối, có nghĩa là dung lượng của nó, các cấu trúc của nó, cách điều khiển truy nhập, công việc này là thuộc trách nhiệm của các driver thiết bị của DOS.

Trong các hệ điều hành cũ trước đây các Driver thiết bị được nạp vào mã của hệ điều hành một cách cứng nhắc đến mức sau đó không thể thay đổi hoặc làm cho chúng thích ứng với một máy mới. Rất may là thế hệ 20 của DOS đã cho ra đời một phương pháp mềm dẻo để ghép các Driver thiết bị vào DOS.

Để xây dựng một Driver thiết bị không được quên rằng còn phải tuân thủ các quy tắc, một chương trình con với một sự khác biệt là một chương trình điều khiển thiết bị không bắt đầu từ địa chỉ offset 100h mà là 0h.

Thao tác nạp DOS bao gồm trước tiên là nạp và cài đặt các Driver NUL, \$ CLOCK, CON, AUX và PRN, cũng như các Driver cho ổ đĩa mềm và đĩa nén có cho ở ổ cứng. Các Driver được lập trực tiếp cho bộ nhớ, cái này sau cái kia và liên kết với nhau theo kiểu một chuỗi.

Các Driver điều khiển các bộ nhớ ngoài như đĩa nén đĩa cứng cũng được thay thế dễ dàng.

Các driver khối

Về nguyên tắc, các driver điều khiển thiết bị khối dùng để liên lạc với các thiết bị khối như đĩa cứng, đĩa mềm hay streamer. Cho nên chúng không chuyên chỉ một ký tự mà là một số lượng các ký tự mỗi lần gọi hàm t gọi số lượng ký tự là một khối. Trong một số trường hợp, nhiều khối có thể được gọi thông qua nhiều lần gọi hàm kích thước của khối thay đổi tùy theo các bộ nhớ khối nhưng tất nhiên là nhỏ hơn kích thước bộ nhớ khối.

Khác với driver thiết bị ký tự, một driver thiết bị khối có thể có nhiều kiểu nhiều thiết bị đồng thời cũng có thể chia một thiết bị thành nhiều đơn vị logic. ví dụ: Một đĩa cứng 40 MB có thể phân chia thành hai ổ đĩa cứng 20 MB gọi là C và D. ở đây, các đĩa logic không mang tên thiết bị, cũng không mang tên File mà chỉ đơn giản là ký hiệu bằng các chữ cái. Ký hiệu ở đĩa không thể được chọn bởi Driver vì rằng các ký hiệu này phụ thuộc vào tính hằng bên trong của chuỗi các driver thiết bị. Nếu một Driver thiết bị đảm bảo nhiều thiết bị logic, thì các thiết bị có tên là các chữ cái liên tiếp nhau. Cho nên đĩa cứng logic thứ hai buộc phải gọi là D, chứ không phải là B. Mỗi thiết bị logic có một bảng phân bố File (FAT-File Allocation Table) và một thư mục gốc. Đối với các driver thiết bị khối, không có sự phân biệt chế độ Cooked và RAM như đối với driver ký tự chúng luôn đọc và viết một số lượng xác định các khối và chúng không xử lý các dữ liệu cần truyền.

Có nhiều cách để truy nhập đến một Driver thiết bị. Ta có thể truy nhập đến các driver ký tự thông qua các hàm FCB hay Handle một cách

bình thường, bằng cách chỉ ra tên của driver ở cho tên file, thí dụ: 'CON'. Ta cũng có thể truy nhập điểm của các Driver khởi thông qua các hàm bình thường của DOS (quản lý file thư mục...) bằng cách chỉ ra tên mới ở chỗ thiết bị cũ. Tuy nhiên điều này sẽ thừa nên ta dùng lệnh ASSIGN như là chúng ta đã giải thích, vì khi đó ta có thể dùng các tên cũ.

Tất nhiên, cũng còn có các cách khác để truy nhập đến các Driver thiết bị.

Bây giờ ta quay lại với Volume. Mỗi volume được DOS chia thành các sector, mỗi sector chứa một số xác định các byte (về nguyên tắc là 512 byte). Chúng được đặt nối tiếp nhau, các sector được đánh số liên tục bắt đầu từ 0. Tuy nhiên, các lời gọi hàm bằng ngắt 21h không bao giờ liên quan đến các sector mà đến File. Do vậy, DOS phải đổi các lời gọi File thành lời gọi sector. Để làm việc này, nó sử dụng các thư mục và một cấu trúc dữ liệu gọi là FAT. Hai loại thông tin này nằm trong bộ nhớ khối. Sau khi các sector logic được xác định, điều khiển trao cho driver thiết bị, nó chuyển các số sector logic thành các địa chỉ vật lý, vì các bộ nhớ khối như đĩa cứng, đĩa mềm được chia thành các rãnh, mà rãnh lại chứa một số xác định các sector nên cần phải xác định không chỉ số thứ tự sector vật lý, mà còn phải xác định một số thứ tự rãnh, số thứ tự đầu đọc/ghi tương ứng.

Cấu Trúc Của Một Volume

Sector boot – sector khởi động
File Allocation Table thứ nhất (FAT)
Một hay nhiều bản sao FAT
Thư mục gốc của Volume
Vùng dữ liệu cho các phai và thư mục con

Như chúng ta đã chỉ ra ở phần trên, mỗi volume được chia thành các vùng, một số vùng để nhận cấu trúc dữ liệu của DOS, một số vùng khác để nhận các File. Kích thước các vùng này thay đổi tùy theo loại bộ nhớ khối hay hãng sản xuất, cho nên, mỗi volume còn chứa các gọi là boot

sector hay sector khởi động. Sector này chứa các thông tin quan trọng để truy nhập đến các vùng và các cấu trúc, sector boot được DOS đặt trong quá trình tạo khuôn volume. Nó luôn có một cấu trúc duy nhất và được đặt ở sector 0, để DOS có thể tìm thấy nó và giải nghĩa nó chính xác. Hình sau mô tả cấu trúc Sector Boot.

Cấu Trúc Sector Boot Của Một Volume

Địa chỉ	Nội Dung	Kiểu
+ 00h	Lệnh nhảy tới thủ tục boot khởi động	3 Byte
+ 03h	Tên hãng sản xuất và thế hệ của DOS	8 Byte
+ 0Bh	Số Byte trên một sector	1 Từ
+ 0Dh	Số sector trên một cluster	1 Byte
+ 0Eh	Số sector để dành	1 Từ
+ 10h	Số bảng FAT	1 Byte
+ 11h	Số đề mục trong thư mục gốc	1 Từ
+ 13h	Số sector trong Volume	1 Từ
+ 18h	Byte mô tả đĩa	1 Byte
+ 16h	Số sector trong một Fat	1 Từ
+ 18h	Số sector trên một rãnh	1 Từ
+ 1Ah	Số đầu đọc/ghi	1 Từ
+ 1Ch	Khoảng cách từ sector đầu tiên của Volume đến sector đầu tiên của Data	1 Từ
+ 1Eh - 1Fh	Thủ tục khởi động	482 Byte
	Độ dài	512 Byte

Sector này chứa tất cả các thông tin cần thiết để khởi động DOS. Khác với nhiều người nghĩ, DOS không có mặt trong bộ nhớ ngay sau khi bật máy đầu tiên, cần nạp DOS vào sau đó là khởi động. Để làm việc này, Bios thực hiện khởi tạo hệ thống sau khi bật máy và cũng chính nó, sau khi công việc khởi tạo kết thúc, nạp vào bộ nhớ sector 0 của đĩa (đĩa mềm hay đĩa cứng) sau đó, nó truyền điều khiển cho chương trình nạp Hệ điều hành nằm trong sector này.

Do vậy, sector boot luôn chứa lệnh JUMP ở địa chỉ 0. Lệnh này có thể là một lệnh nhảy bình thường hay lệnh “SHORT JUMP” (lệnh nhảy gần). Thường dành cho lệnh nhảy dài 3 Byte, trong khi một lệnh “SHORT JUMP” chỉ dài 2 Byte. Do đó, sau lệnh này luôn là lệnh NOP, lệnh này không làm gì cả và chiếm 1 Byte, để cho cả 3 Byte của trường đều được điền. Sau trường này là một loạt các trường chung cung cấp các thông tin chính xác về cấu trúc của đĩa. Trường đầu tiên dài 8 Byte, nó chứa tên hãng sản xuất máy tính, và thế hệ của DOS đã được tạo khuôn đĩa. Các trường tiếp theo chứa các thông tin về khuôn vật lý của đĩa, có nghĩa là số Byte trên một sector, số sector trên một rãnh..., ở đây còn có kích thước cấu trúc dữ liệu của DOS, các cấu trúc này được đặt trên đĩa. Các thông tin này rất quan trọng đối với các hàm của ROM-BIOS hay BIOS – DOS, các hàm này được thực hiện ở mức thấp nhất để truy nhập trực tiếp đến đĩa mềm hay đĩa cứng. Cho nên vùng này được gọi là khối tham số của BIOS (BPB). Sau BPB là 3 trường nữa, các trường này không được DOS sử dụng, nhưng chúng có thể cung cấp các thông tin bổ sung về đĩa cho driver thiết bị.

Sau vùng này là cái mà ta gọi là BOOT STRAP (cái xỏ giày), lệnh nhảy ở đầu sector chính là nhảy đến đây. Chính thủ tục này chịu trách nhiệm và khởi động DOS.

Sau sector Boot có thể có nhiều sector được dùng để dành, chúng có thể được dùng để chứa phần tiếp theo của mã chương trình BOOT STRAP. Số lượng các sector này được viết trong BPB ở địa chỉ 0Eh. Số 1 ở đây chỉ ra rằng không có sector để dành và đây là trường hợp đối với đa số PC.

Số Lượng Sector Của Một Cluster

Máy	Sector/cluster
Đĩa một mặt và đĩa dung lượng cao	1

Đĩa hai mặt	2
Đĩa cứng của AT	4
Đĩa cứng của XT	8

Khi DOS muốn tạo các File mới hay tăng kích thước của các File đã tồn tại thì chắc chắn nó phải biết các sector còn tự do trên đĩa. DOS lấy các thông tin này từ một cấu trúc gọi là FAT(File Allocation Table). FAT được đặt ngay sau vùng để dành, nếu không có vùng để dành thì nó nằm ngay sau boot sector. Mỗi đề mục của FAT tương ứng với một trang, một trang có thể chứa một hay nhiều sector đặt nối tiếp nhau. Số lượng các sector tạo nên một trang được đặt trong ô nhớ 0Dh của sector Boot, số này là một thành phần của bản tham số BIOS, giá trị của trường này luôn là lũy thừa của 2. Đối với các đĩa cứng của AT, trường này luôn chứa giá trị 8 nên trang sẽ chứa 8 sector kề nhau. Bảng sau cho thấy số lượng các sector tạo thành một trang thay đổi theo các loại đĩa như thế nào.

Việc nhóm các sector thành một trang là do cách mà DOS dùng để ghi các File. Thực vậy, DOS không tự động chọn các sector kế tiếp nhau để ghi File, vì sẽ tiếp kiệm hơn nếu chia nhỏ các File sao cho có thể dùng tất cả các sector còn tự do, ngay cả khi chúng đứng kế tiếp nhau. Những phương pháp này làm giảm truy nhập của File, vì đầu từ đọc/ghi cần phải dịch lại mỗi khi truy nhập, do vậy, cần tránh việc chia nhỏ các File và người ta đã nhóm nhiều sector liên tục thành một trang, như vậy có thể cho rằng các sector của một trang là một đơn vị File, để cho dễ hiểu ta xét ví dụ sau:

Giả sử nếu ta không nhóm các sector thành một trang thì một File dài 24 sector hoàn toàn riêng biệt. Và đầu từ đọc/ghi phải đổi vị trí 24 lần để đọc hết File. Nhưng nếu ta nhóm, thí dụ, 4 sector thành một trang thì có thể tiếp kiệm được thời gian truy nhập nhưng lại lãng phí về kích thước. Thật vậy, mỗi File chiếm ít nhất một trang, và trong không ít trường hợp như thế là lãng phí, thí dụ, ta xét File AUTOEXEC.BAT, nó thường chứa không quá 150 Byte. Một sector đã là quá thừa để ghi File này, ta thừa đến gần 400 Byte. Còn đối với đĩa cứng AT, một File nhỏ nhất cũng chiếm cả một trang, có nghĩa là $4 \times 512 = 2084$ Byte và như

vậy, chỉ khoảng 10% của trang được dùng. Do vậy, hơn 1,5 KB bị lãng phí đối với các File nhỏ hơn 512 Byte, và với các file mà trang cuối cùng không bị chiếm quá 512 Byte.

Và bây giờ chúng ta quay lại với bảng FAT. Kích thước đề mục của bảng FAT là 12 Bit trong các thế hệ 1 và 2 của DOS. Trong thế hệ 3 của DOS, kích thước này phụ thuộc vào số lượng trang trên đĩa. Nếu có hơn 4079 trang, 16 bit được dùng, nếu không thì 12 bit được dành cho một đề mục. Do vậy, giá trị này cần được tính trước, khi truy nhập bảng FAT. Để làm việc này, ta dùng các thông tin được cung cấp trong khối tham số BIOS, tổng số sector trên đĩa được chỉ ra ở ô nhớ 13h. Chỉ cần chia giá trị này cho số sector trên một trang là nhận được số trang trong volume.

Hai đề mục đầu tiên của bảng FAT được để dành. Tùy theo kích thước hai đề mục này có thể chiếm 24 Bit(3 Byte) hay 32 Bit(4 Byte). Byte đầu tiên chứa cái mà ta gọi là Byte mô tả đĩa, trong khi các byte tiếp theo luôn chứa giá trị FFh. Byte mô tả đĩa cùng nằm ở địa chỉ 15h của BPB. Mã của nó có thể là:

Mã Của Byte Mô Tả Đĩa (Media Descriptor)

Mã	Đĩa
F0h	Đĩa mềm 3 1/2", 2 mặt, 80 rãnh, 18 sector/rãnh
F8h	Đĩa cứng
F9h	Đĩa mềm 5 1/4", 2 mặt, 80 rãnh, 5 sector/rãnh Đĩa mềm 3 1/2", 2 mặt, 80 rãnh, 9 sector/rãnh
FAh	Đĩa mềm 5 1/4", 1 mặt, 80 rãnh, 8 sector/rãnh Đĩa mềm 3 1/2", 1 mặt, 80 rãnh, 8 sector/rãnh
FBh	Đĩa mềm 5 1/4", 2 mặt, 80 rãnh, 8 sector/rãnh Đĩa mềm 3 1/2", 2 mặt, 80 rãnh, 8 sector/rãnh
FCh	Đĩa mềm 5 1/4", 1 mặt, 40 rãnh, 9 sector/rãnh
FDh	Đĩa mềm 5 1/4", 2 mặt, 40 rãnh, 9 sector/rãnh
Feh	Đĩa mềm 5 1/4", 1 mặt, 40 rãnh, 8 sector/rãnh

FFh	Đĩa mềm 5 1/4", 2 mặt, 40 rãnh, 8 sector/rãnh
-----	---

Các Khuông Của Đĩa 5 1/4"

Tồn Tại Từ Thế Hệ Của DOS	1.0	1.10	2.0	2.0	2.0
Byte mô tả đĩa	FEh	FFh	FCh	FDh	F9h
Số đầu đọc/ghi	1	2	1	2	2
Số rãnh trên một mặt	40	40	40	40	80
Số sector/rãnh	8	8	9	9	15
Số Byte/sector	512	512	512	512	512
Số sector/cluster	1	2	1	2	1
Số sector để dành	1	1	1	1	1
Số sector cho FAT	1	1	2	2	7
Số bảng FAT	2	2	2	2	2
Số sector cho thư mục gốc	4	7	4	7	14
Số đề mục trong thư mục gốc	64	112	64	112	224
Tổng số sector	320	640	360	720	2400
Số sector dành cho các File	313	630	351	708	2371
Số cluster	313	315	351	354	2371
Tổng dung lượng	160 KB	320 kb	180kb	360 kb	1,2 M
Dung lượng dành cho các File	156,5kb	315 kb	178,5	354 kb	1,185M

Bảng này trình bày các dạng khuôn của loại đĩa 5 1/4" mà các thế hệ của DOS đảm bảo.

Đối với loại đĩa 3 1/2", có dạng khuôn sau:

Các Khuôn Của Đĩa 3 1/2"

Chắc chắn bạn sẽ tự hỏi, tại sao các đề mục của FAT lại phải chiếm đến 12 hay 16 Bit, nếu như nó chỉ xác định xem trang đã bị chiếm hay chưa.

Chỉ một Bit là đủ để thông báo trang đã bị chiếm (giá trị 1) hay còn tự do (giá trị 0).

Tồn Tại Từ Thế Hệ Của Dos	3.0	3.30
Byte mô tả đĩa	F9h	F0h
Số đầu từ đọc ghi	2	2
Số rãnh trên một mặt	80	80
Số sector/rãnh	9	18
Số byte/sector	512	512
Số sector/cluster	2	1
Số sector để dành	1	1
Số sector cho một FAT	3	9
Số bảng FAT	2	2
Số sector để dành cho thư mục gốc	7	14
Số đề mục cho thư mục gốc	112	224
Tổng số sector	1440	2880
Số sector dành cho file	1426	2847
Tổng số cluster	720	2880
Tổng dung lượng	720 kb	1.5 M
Dung lượng dành cho File	713	1.4 M

Nhưng vấn đề lại không phải như vậy, bạn có thể tự đoán ra rằng các đề mục của FAT không phải chỉ có nhiệm vụ chỉ ra một trang đã bị chiếm hay còn tự do, mà còn phải chỉ ra dữ liệu của mỗi File nằm ở những trang nào. Đề mục của thư mục sẽ cho DOS biết trang đầu tiên của File nằm ở đâu, số thứ tự của trang bằng số thứ tự của đề mục của bảng FAT. Mỗi đề mục của bảng FAT chỉ ra số thứ tự của trang tiếp theo của File theo như hình sau, ta sẽ thấy có một xâu được hình thành nó cho phép định vị, theo một thứ tự.

Đề mục của FAT tương ứng với trang cuối cùng của một File, phải chứa một mã đặc biệt để chỉ cho DOS biết File đã kết thúc. Bảng sau cho biết ý nghĩa nội dung của một đề mục bảng FAT.

Mã	ý nghĩa
(0)000h	Cluster tự do
(F)F00h...(F)FF6h	Cluster để dành
(F)FF8h...(F)FFFh	Cluster bị hỏng, không dung cluster cuối cùng của một File.
(X)XXh	Cluster tiếp theo của file

Số hexa trong ngoặc đơn liên quan đến đề mục FAT có độ dài 16 bit DOS cho phép lưu trữ nhiều bản sao y hệt nhau của FAT trên đĩa. Ưu điểm ở đây là nó cho phép thay bản FAT chính bằng bản sao trong trường hợp bản chính bị hỏng. Rõ ràng, điều này là nên làm.

Thi dụ: lệnh CHKDSK cho phép kiểm tra xem các bản sao của FAT có giống nhau không.

Chúng ta đã từng nhắc đến thư mục, bây giờ là lúc chúng ta nghiên cứu cấu trúc của nó.

Thư mục gốc của một volume đặt ngay sau bản sao cuối cùng của FAT, cũng như các thư mục khác, nó chứa các đề mục của 32 Byte. Các đề mục này cung cấp các thông tin về File, thư mục con và tên của volume. Số lượng cực đại các đề mục trong thư mục gốc được đặt trong BPB tại địa chỉ 11h. Giá trị này được lệnh Format quyết định trước khi mô tả các trường của cấu trúc này, ta xét hìh sau:

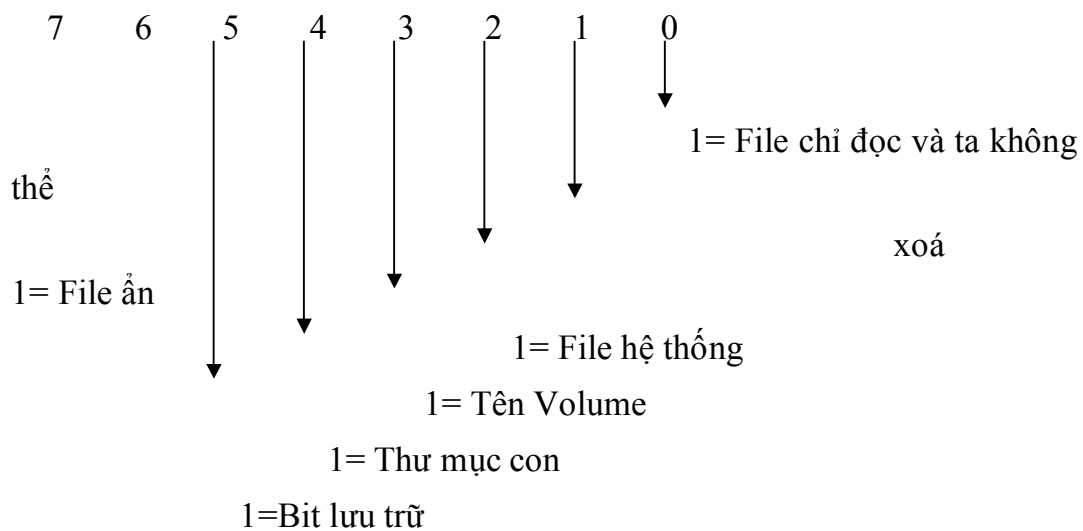
Địa Chỉ	Nội Dung	Kiểu
00h	Tên File	8 Byte
08h	Phần mở rộng của File	3 Byte
0Ch	Thuộc tính File	1 Byte
16h	Để dành	10 Byte
18h	Giờ của lần thay đổi cuối cùng	1 Từ
1Ah	Ngày của lần thay đổi cuối cùng	1 Từ

1Ch	Cluster đầu tiên của File	1 Từ
	Kích thước File	1 Từ kẹp
	Độ dài	32 Byte

8 Byte đầu tiên chứa tên File. Nếu tên này ngắn hơn 8 Byte, nó sẽ được đầy bằng những dấu trắng(mã ASCII 32)..Nếu đề mục thư mục không chứa các thông tin liên quan đến File, giả sử vì nó được sử dụng cho một mục đích khác, thì byte đầu tiên trên File, có nghĩa là byte đầu tiên của đề mục, nó sẽ nhận một mã đặc biệt, sẽ nhận một mã đặc biệt trường thứ hai dài 3 byte chứa phần mở rộng của File, ở đây cũng vậy, trường được làm bằng các ký tự trắng. Dấu chấm giữa tên File và phần mở rộng không được lưu trong đề mục

Mã	ý nghĩa
00h	Đề mục cuối cùng của thư mục
2Eh	File liên quan đến thư mục con hiện tại, nếu có một 2Eh nữa tiếp theo nó liên quan đến thư mục mẹ
E5h	File đã bị xoá

Cấu Trúc Của Byte Thuộc Tính Trong Đề mục Của Một File



ý nghĩa của các Bit 0 đến Bit 4 là khá rõ ràng. Ngược lại, Bit 5 yêu cầu thêm 1 và giải thích bổ sung, Bit này được gọi là bit lưu

trữ(archive), vì nó yêu cầu sao bảo vệ. Một chương trình sao bảo vệ, thí dụ: chương trình Backup của DOS, sẽ lưu trữ File đã tạo ra hay bị thay đổi, Bit này được đặt bằng 1 có nghĩa là File đã bị thay đổi và cần phải sao bảo vệ lại nó.

Sau trường thuộc tính là trường để dành mà DOS sẽ dùng cho các thao tác bên trong của nó.

Các trường giờ và ngày cho biết File được tạo hay thay đổi lần cuối vào lúc nào. Hai thông tin này được ghi trong đề mục dưới dạng một từ(2 byte), nhưng khuôn của chúng là khác nhau.

Khuôn của trường giờ trong đề mục File

Bit 0..4: giây, mỗi đơn vị là hai giây.

Bit 5..10: phút

Bit 11..15: giờ

Khuôn của trường ngày trong đề mục File

Bit 0..4: ngày trong tháng

Bit 5..8: tháng (1 đến 12)

Bit 9..15: năm (so với năm 1980)

Trường tiếp theo cho biết số thứ tự của trang chứa những dữ liệu đầu tiên của File (trang đầu tiên của File). Giá trị này đồng thời cũng chỉ ra số thứ tự của đề mục trong bảng FAT, đề mục này chỉ ra số thứ tự của trang tiếp theo của File. Như vậy, trường trang trong đề mục File của thư mục, tạo thành mắt xích đầu tiên của chuỗi cho phép tìm ra tất cả các trang của File.

Kích thước file tính bằng Byte được ghi dưới dạng 1 từ kép (2 từ) từ thấp được đặt trước, công thức sau cho phép tính kích thước thực của File

Kích thước File= từ thấp+từ cao+ 65536.

Hai đề mục “thư mục con” Và “tên volume” của thư mục xứng đáng được có một thí dụ riêng. Tên volume chỉ tồn tại trong thư mục gốc. Nếu nó được ký hiệu bằng bit 3 của những thuộc tính trong đề mục. Tên File trong một đề mục loại này được coi là tên của đĩa, nó sẽ được hiện lên màn hình khi gọi các lệnh DIR và TREE.

Nếu bit 4 của trường thuộc tính bằng 1, có nghĩa là đề mục này liên quan đến thư mục con. Trong một đề mục loại này, các trường tên File

và phần mở rộng nhận tên các thư mục con, các trường ngày và giờ nhận ngày và giờ tạo các thư mục con và trường độ dài File luôn chứa giá trị 0. Trường chỉ ra trang đầu tiên của File cho biết trang nào chứa các đề mục của thư mục con này. Cũng như đối với một File bình thường, đề mục tương ứng của FAT chỉ ra trang tiếp theo của thư mục này, nếu thư mục dài hơn một trang. Đây không phải là trường hợp đối với thư mục gốc, mặc dù bản thân nó có thể chiếm nhiều sector hay trang, bởi vì các trang này luôn luôn nối tiếp nhau. Các trang thư mục gốc không thể liên hệ với nhau thông qua bảng FAT, bởi vì FAT chỉ trỏ đến vùng dữ liệu của đĩa, có nghĩa là trỏ đến vùng chỉ có thể nhận các File và các thư mục con chứ không phải là thư mục gốc. Phương pháp mà chúng ta mô tả tiết lộ hai khía cạnh quan trọng trong việc quản lý File của DOS, các File của bộ nhớ khối được phân biệt dựa vào thư mục, nhưng các File của một thư mục của một thư mục lại không nhất thiết phải nằm trong một vùng xác định của bộ nhớ khối mà chúng có thể được phân bố ở những chỗ rất khác nhau trong bộ nhớ khối.

Khi một thư mục con tạo ra, hai đề mục con lập tức xuất hiện với các tên '.' Và '..' các đề mục này chỉ mất đi khi toàn bộ thư mục con bị xoá, tức là khi nó không có mặt trong các thư mục của đĩa này. File đầu tiên '.' chỉ thư mục hiện tại, và các trang của nó chứa số thứ tự của trang đầu tiên của thư mục hiện tại và trường trang của nó chứa số thứ tự của trang đầu tiên thư mục hiện tại. Đề mục thứ hai (File '..') chỉ ra cái gọi là thư mục mẹ, nó là đầu mỗi lần ra mỗi quan hệ giữa thư mục con hiện tại với các thư mục còn lại trong cây thư mục. Nếu thư mục mẹ là gốc, thì trường trang có giá trị là 0. Chính đề mục này cho phép DOS tìm ra con đường dẫn đến thư mục gốc, vì mỗi lần tìm thư mục mẹ của thư mục con ta luôn tiến dần đến thư mục gốc.

Mặc dù DOS xử lý thư mục con và tên là Volume như là các File, ta không thể truy nhập đến nó thông qua các hàm thao tác File. Tôi chỉ có thể khuyên các bạn đừng bao giờ có mưu toan làm việc đó.

Còn bây giờ, tôi quay lại với vấn đề nghiên cứu cấu trúc bộ nhớ khối trong môi trường DOS. Sau thư mục gốc mà chúng ta vừa mô tả là vùng của các File, nó chiếm phần còn lại của bộ nhớ khối, vùng này nhận các File cũng như các thư mục con. Mỗi một trang của vùng tương ứng với một đề mục của FAT, mà FAT thì có thể bị thay đổi mỗi khi các file

được thao tác. Nếu một File lớn hơn lên, DOS sẽ giành thêm một trang tự do nữa để lưu các dữ liệu tiếp theo của File. Đề mục FAT tương ứng với trang cuối cùng của File giờ đây sẽ bị thay thế bằng một đề mục FAT khác trỏ đến trang mới, còn đề mục của trang mới sẽ nhận dấu hiệu kết thúc File. Để tìm một trang có thể nhận dữ liệu mới của File, các thế hệ 1 và 2 của DOS sẽ kiểm tra FAT từ đầu đến cuối để tìm ra trang đầu tiên. Thế hệ 3 của DOS ứng dụng một phương pháp phức tạp hơn để tìm ra các trang tự do gần trang cũ, như vậy sẽ giảm được thời gian truy nhập File. Khi một File bị cắt ngắn hay bị xoá, các trang được giải phóng sẽ được đánh dấu là tự do (các đề mục của chúng chứa giá trị 0) trong bảng FAT. Do vậy, chúng có thể sử dụng để tạo File mới hay tăng kích thước của một File đã có.

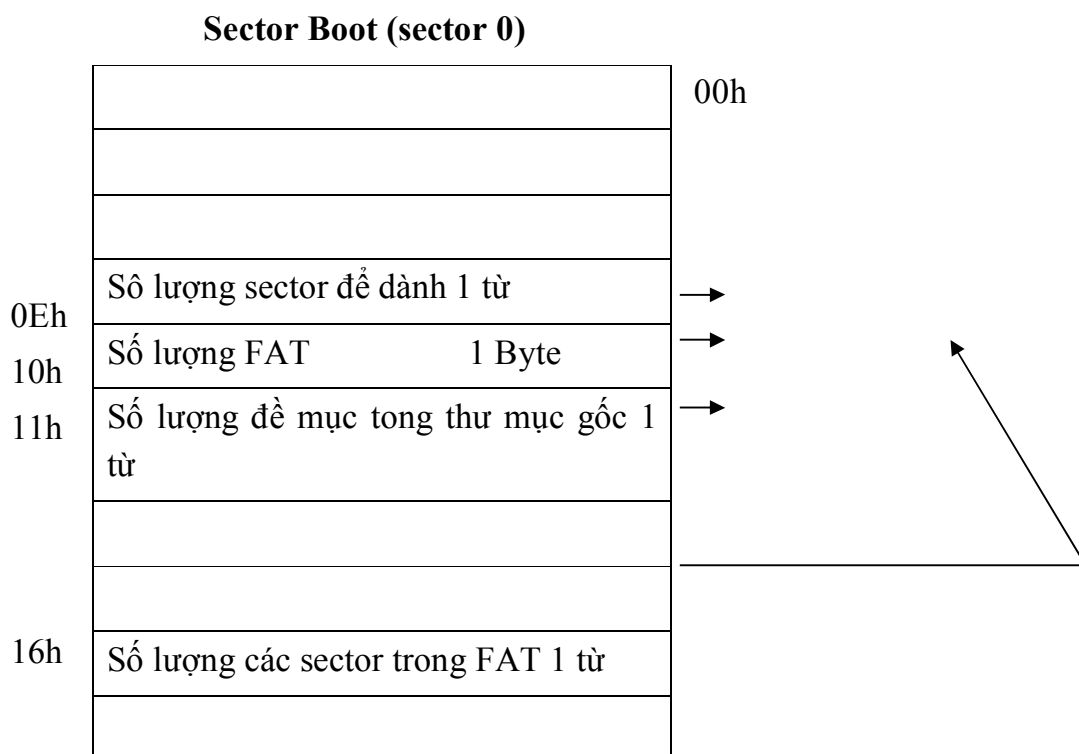
Chúng không dừng lại ở chỗ giải thích một cách lý thuyết. Bây giờ chúng ta sẽ tìm hiểu xem làm thế nào để truy nhập đến các sector của File thông qua FAT.

Để bắt đầu chúng ta cần xác định trang đầu tiên của InTel và đề mục đầu tiên của FAT, ta lấy thông tin của trang đầu tiên của File trong đề mục File, số này tương ứng với số đề mục của FAT. Ta để ý là hai đề mục đầu tiên của FAT đã bị chiếm và không dùng để trỏ đến trang của File (nó dùng cho byte mô tả đĩa). Do vậy, đề mục thứ 3 của FAT tương ứng với trang 0 trên đĩa. Cho nên, giá trị trong trường trang đầu tiên phải trừ đi 2, thì mới thực sự trỏ đến trang đầu tiên trên đĩa. Để nhận được số sector đầu tiên, ta nhân số trang nhận được với số sector trong một trang.

Tuy nhiên chúng ta không thể có ngay số trang phù hợp, vì số thứ tự trang trong thư mục File cũng như trong đề mục FAT không được tính từ sector đầu tiên của đĩa, mà tính từ sector đầu tiên của vùng dữ liệu đĩa.

Như ta đã thấy ở đầu, vùng này chỉ được bắt đầu sau thư mục gốc của Volume. Do vậy, cần cộng số thứ tự của sector tính được với số thứ tự của sector đầu tiên của vùng dữ liệu, số thứ tự của sector đầu tiên của vùng dữ liệu có thể tính dựa vào thông tin trong bảng tham số BIOS. Như chúng ta đã biết, vùng dữ liệu của đĩa cứng sau Boot, FAT, các bảng sao của nó và cuối cùng là thư mục gốc. Số lượng sector để dành (ở đây tính cả sector boot) được thông báo ở ô nhớ 0Eh của sector boot.

Còn ô nhớ 16h chỉ ra số sect_or của một bảng FAT. Giá trị này được nhân với số lượng bảng FAT, số này ở ô nhớ 1Ah. Cuối cùng, ta lấy số lượng các đề mục của thư mục gốc ở trong từ có địa chỉ 11h của boot sector, nhân giá trị này với 32(số byte trên một đề mục), sau đó, chia kết quả cho 512(số byte trên một sector), ta sẽ nhận được số lượng các sector của thư mục gốc. Cộng tất cả các kết quả lại ta nhận được số thứ tự của sector đầu tiên của vùng dữ liệu.



Hình : Tính Sector Đầu Tiên của Vùng Dữ liệu

Cộng tổng này với số thứ tự logic của sector đầu tiên của File, ta nhận được số thứ tự sector tuyệt đối của sector đầu tiên của File. Sector thứ nhất và một số sector tiếp theo tạo thành trang thứ nhất, tùy theo lượng sector của mỗi trang.

Nếu ta muốn xác định trang tiếp theo và đề mục tiếp theo của FAT thì chúng ta phải đọc thông tin trong bảng FAT. Việc tiến hành công việc này phụ thuộc chủ yếu vào kích thước đề mục của bảng FAT. Thật vậy, để xác định trang tiếp theo, nếu kích thước của đề mục FAT là 16 bit, trong trường hợp này, chỉ cần nhân số trang với 2. Tính số này là địa chỉ tương đối (offset) so với đầu của bảng FAT (mỗi Byte của bảng

FAT là một đơn vị bắt đầu từ 1) hai Byte liên tiếp tính từ địa chỉ offset chứa số thứ tự của trang tiếp theo. Nếu kích thước đề mục FAT là 12 Bit thì số thứ tự trang phải nhân được với 1,5, phần nguyên của tích số này sẽ được dùng như là địa chỉ offset. Nếu tích số này là nguyên thì ta sẽ đọc đề mục từ offset đó, nếu tích số không phải là nguyên, từ địa chỉ offset ta dịch thêm 4 bit về bên phải và đọc đề mục từ đó (chú ý bit thấp viết trước bit cao).

Như vậy ta đã biết trang đầu tiên của file. Nếu giá trị của đề mục đầu tiên của FAT nằm giữa FF8h và FFFh đối với FAT 12 bit, hay giữa FFF8h và FFFFh đối với FAT 16 bit thì có nghĩa là đã đến cuối file. Nếu Không, thao tác cần được thực hiện lại từ đầu để tìm trang tiếp theo.

CD – ROM – Một Thiết Bị Đặc Biệt

Chỉ sau vài năm thâm nhập vào lĩnh vực sản phẩm chất lượng cao (Hi-Fi) và với sự bùng nổ thị trường sau đó, công nghiệp đĩa laser ngày nay đã tấn công thị trường PC. Phải công nhận rằng một ổ đĩa CD-ROM và một PC là một sự kết hợp thú vị vì đĩa laser cung cấp cho PC một bộ nhớ khối ngày nay mới chỉ là đọc được, nhưng nó lại thay được và cho phép lưu trữ trên rãnh duy nhất của nó 660 MB dữ liệu, văn bản hay hình ảnh... phải chăng CD-ROM mở ra một kỷ nguyên mới cho tin học? Điều này không phải là không có thể vì trong lĩnh vực Hi-Fi, ta đã biết rằng giá của các đĩa laser sẽ là chấp nhận được với sự ra đời của đĩa dung lượng lớn. Nhưng vấn đề đặt ra với phần mềm, DOS chưa có một chuẩn bị gì để đảm bảo một thiết bị loại này.

Đĩa CD-ROM là một bộ nhớ khối, các thông tin được đọc ra theo từng khối, ta có thể cho rằng nó được điều khiển bởi một driver khối, nhưng điều này lại không đúng và ở đây xuất hiện các phức tạp, vì DOS yêu cầu thiết bị khối một số điều kiện mà ổ đĩa CD-ROM không có.

Dung lượng của CD-ROM không cho phép sát nhập nó thông qua driver khối, vì rằng dung lượng của một thiết bị khối bị hạn chế trước thế hệ 4.0 là 32 MB, ngoài ra, CD-ROM không có File Allocation Table (bảng phân bố File), vì không thể viết vào các sector và không có ssector nào được dành cho FAT. Thay vào đó, CD-ROM có bảng các File, trong đó có địa chỉ bắt đầu của thư mục con và các File.

Để viết một driver CD-ROM, có lẽ nên dùng đến driver ký tự, đối với các driver này, DOS không đặt bất kỳ điều kiện gì về cấu trúc thiết bị, nhưng mặt khác, các driver này sẽ đặt khó khăn trong việc liên lạc với ổ đĩa CD-ROM, vì chúng không có số liệu theo khối mà theo từng Byte và không mang ký hiệu thiết bị và mang một tên File như "CON" hay "NUL". Dù sao đi chăng nữa đọc FAT vốn không tồn tại trong CD-ROM. Driver sẽ không mang tên thiết bị, mà lấy tên của mình trong File cấu hình CONFIG.SYS. Vì rằng driver trong CD-ROM chủ yếu là do các hãng sản xuất CD-ROM cung cấp, nên nó mang tên theo kiểu SONY.SYS hay HITACHI.SYS, và lời gọi nó trong File CONFIG.SYS được viết theo cách sau:

```
Driver= HiTachi.sys/d:CDR1
```

Driver thiết bị chọn tên CDR1 như là tên của ổ đĩa CD-ROM

Sau thủ tục khởi tạo, Driver mới thể hiện đúng là mình, là một driver khối, nó bổ sung thêm một số hàm đặc biệt để đảm bảo hoạt động cho CD_ROM nhưng đối với DOS, nó vẫn là một driver ký tự và không thể kiểm tra thư mục của CD-ROM hoặc truy nhập đến các File.

Để vượt qua trở ngại này, một chương trình nội chúng (TSR) gọi là MS CDEX (Microsoft CD-ROM Extention) được cung cấp để đảm bảo hoạt động cho ổ đĩa CD-ROM. Chương trình này được gọi trong File Autoexec.bat.

Trong dòng lệnh, tên của driver CD-ROM phải được chuyển từ tham số cho chương trình.

MS CDEX/D: CDR1

Trước tiên, MS CDEX mở driver thông qua hàm Open của DOS, cung cấp cho nó một ký hiệu thiết bị, MSCDEX làm cho DOS tin rằng CD-ROM là một ổ đĩa ở xa trong mạng. Kể từ thế hệ 3.1, DOS đảm bảo mạng.

Các ổ đĩa của mạng được DOS xử lý như là File lớn, các File này có thể chứa đến hơn 32 MB. Ngoài ra các thiết bị này không được DOS gọi một cách trực tiếp, nó thông qua cái mà ta gọi là bộ đổi hướng. Phần nội chú của MS CDEX được cài trong bộ đổi hướng. Bộ đổi hướng này thông dịch mọi lời gọi, nếu xuất hiện một lời gọi hướng tới ổ đĩa CD-ROM, nó không được truyền cho bộ đổi hướng theo cách truyền thông, mà mỗi lệnh được chuyển thành lời gọi các hàm của Driver CD-ROM. Như vậy, mối liên hệ giữa DOS và ổ đĩa CD-ROM được thực hiện một cách hoàn hảo và sự truy nhập đến các thư mục con, các File có thể cho phép vào bất kỳ lúc nào.

III. Hệ thống bảo vệ.

Hệ điều hành có thể được coi như là bộ phân phối tài nguyên của máy tính. Nhiều tài nguyên của máy tính như thời gian sử dụng CPU, vùng bộ nhớ, vùng lưu trữ tập tin, thiết bị nhập xuất... được các chương trình yêu cầu để giải quyết vấn đề. Hệ điều hành hoạt động như một bộ quản lý các tài nguyên và phân phối chúng cho các chương trình và người sử dụng khi cần thiết. Do có rất nhiều yêu cầu, hệ điều hành phải giải quyết vấn đề tranh chấp và phải quyết định cấp phát tài nguyên cho những yêu cầu theo thứ tự nào để hoạt động của máy tính là hiệu quả

nhất. Một hệ điều hành cũng có thể được coi như là một chương trình kiểm soát việc sử dụng máy tính, đặc biệt là các thiết bị nhập xuất.

Để đảm bảo an toàn hệ thống và các dữ liệu đi cùng thì các chương trình và dữ liệu phải được bảo vệ ở nhiều mức bằng các khoá khác nhau.

Trong một hệ thống nhiều người sử dụng và cho phép tiến trình diễn ra đồng thời, các tiến trình phải được bảo vệ đối với những hoạt động khác. Do đó hệ thống cung cấp cơ chế để đảm bảo rằng tập tin, bộ nhớ, CPU, và những tài nguyên khác chỉ truy xuất bởi những tiến trình có quyền.

VD: Bộ nhớ đảm bảo rằng tiến trình chỉ được thi hành trong phạm vi địa chỉ của nó. Bộ thời gian đảm bảo rằng không có tiến trình nào độc chiếm CPU. Cuối cùng các thiết bị ngoại vi cũng được bảo vệ.

Hệ thống bảo vệ là một cơ chế kiểm soát quá trình truy xuất của chương trình, tiến trình, hoặc người sử dụng với tài nguyên của hệ thống. Cơ chế này cung cấp cách thức để mô tả lại mức độ kiểm soát.

Hệ thống bảo vệ cũng làm tăng độ an toàn khi kiểm tra lỗi trong giao tiếp giữa những hệ thống nhỏ bên trong.

Hệ thống bảo vệ phải làm các công việc như bảo vệ các tài nguyên sử dụng chung, phát hiện và ngăn chặn các khả năng sai sót của các quá trình.

Sau đây là một số cách bảo vệ hệ thống:

1. Dùng phục vụ năm(format track đĩa) để bảo vệ chống sao chép:

Track có thể được format theo đủ kiểu, nhưng hầu hết các hệ điều hành chỉ có thể đọc được một số kiểu nhất định. Do đó, hầu hết các cơ chế bảo vệ chống sao chép đều dựa vào kiểu chống format khác thường khiến cho hệ điều hành không thể đọc hay ghi dữ liệu lên được. Ta có thể chọn từ nhiều phương pháp bảo vệ chống sao chép sau đây:

- Sắp xếp lại thứ tự các sector nhằm làm thay đổi thời gian truy cập theo cách mà cơ chế bảo vệ chống sao chép có thể phát hiện được.
- Nén nhiều sector hơn vào track(10 là giới hạn cho sector 512 byte)
- Bỏ qua một số hiệu sector
- Cộng vào số hiệu sector một số lạ(Ví dụ: Cho ra R=22)
- Chỉ định một hoặc nhiều sector có kích thước khác thường.
- Ghi sai các giá trị C và H

Bạn có thể chọn bất kỳ phương pháp nào trên đây để chống sao chép hoặc thay đổi đặc tính làm việc của đĩa mềm. Tuy theo cách chọn, một đĩa mềm format thông thường có thể chứa đựng đặc tính bất thường mà DOS hoàn toàn không thấy, và như vậy không thể dùng các lệnh thông thường để phát hiện ra cơ chế bảo vệ chống sao chép này.

2. Phục vụ 89 HEX: (AT) đặt kiểu bảo vệ.

- Phục vụ này đặt kiểu bảo vệ cho các máy AT. Ta đặt số hiệu ngắt cứng ứng với IRQ0 vào BL (IRQ1-7 ứng với 7 ngắt kế tiếp) và số hiệu ngắt cứng ứng với IRQ8 vào BH (IRQ9-F ứng với 7 ngắt kế tiếp). Cặp ES:SI được trỏ đến bảng mô tả chung GDT cho kiểu bảo vệ, có cấu trúc sau:

Offset (Hex)	Mô tả
0	= 0
8	Mô tả GDT
10	Mô tả IDT
18	DS
20	ES
28	SS
30	CS

CX = offset trong kiểu bảo vệ CS cần nhảy đến.

Chú ý: BL và BH phải là bội số của 8

2. Phục vụ 90 HEX: (AT) Thiết bị bận:

Phục vụ 90 Hex cho phép trình điều khiển thiết bị ngăn ngừa hệ điều hành thực hiện hoạt động vào/ra thiết bị. Nó sẽ móc phục vụ này và ngăn hoạt động vào/ra tiếp theo cho đến khi nhận được tín hiệu hoàn thành qua phục vụ 91 Hex.

Các trình điều khiển đĩa, bàn phím, máy in đều dùng phục vụ này. Kiểu thiết bị được thông báo trong thanh ghi AL (0 = đĩa cứng; 1 = đĩa mềm; 2 = bàn phím; 80 hex = mạng; FD hex = mô-đun đĩa mềm khởi động; FE hex = máy in) Cặp ế:BX trỏ đến khối yêu cầu (Request block) cho kiểu thiết bị từ 80 đến BF Hex.

Chú ý: Kiểu thiết bị được chia ra như sau:

- 00 đến 7F: thiết bị chỉ xử lý mỗi lúc một yêu cầu (non-reentrant)

- 80 đến BF: Thiết bị xử lý được nhiều yêu cầu đồng thời (reentrant) ES:BX đến khối yêu cầu duy nhất.
- C0 đến FF: Thiết bị mà Hệ điều hành phải đợi trong khoảng thời gian xác định trước khi tiếp tục điều khiển.

Các chương trình đa nhiệm cần móc phục vụ này để cho phép các quá trình khác thực hiện khi BIOS đang chờ việc vào/ra thiết bị được hoàn thành.

Khi phục vụ này trở về, nếu cờ nhớ CF=CY=1 là thời gian chờ đã trôi qua. Nếu CF=NC=0 là trình điều khiển thiết bị đang chờ. VD các trình điều khiển đĩa đều móc phục vụ này để cài đặt thời gian chờ(timeout): nếu phục vụ trở về với CF=1 nghĩa là thời gian chờ đã trôi qua mà thiết bị không trả lời.

3. Undelete: Lệnh được sử dụng khi muốn cứu các tập tin bị xoá.

Khi bạn xoá một tập tin, thực ra MS-DOS không xoá bỏ dữ liệu trong tập tin này, thay vì thế, nó chỉ đánh dấu tập tin này là đã bị loại bỏ để MS-DOS có thể dùng lại vùng nhớ vốn bị tập tin này chiếm chỗ trên đĩa. Dữ liệu cũ vẫn còn nguyên trên đĩa cho đến khi MS-DOS ghi dữ liệu mới của một tập tin khác lên chính vùng đó.

Cấu trúc lệnh:

Undelete [[ô đĩa] [đường dẫn] [tên tập tin] [/DT/DS/DOS]].

4. Defrag: Lệnh được sử dụng để sắp xếp lại khối lưu trữ của tập tin, tránh tình trạng phân mảnh đĩa:

Cấu trúc lệnh:

DEFRAG [ổ đĩa:] [/F] [/S[:] thứ tự] [/B] [/Skiplhigh]

5. *Scandisk*: **Lệnh dùng để phân tích, sửa chữa các lỗi về đĩa hay tập tin, thư mục.**

Cấu trúc lệnh:

SCANDISK [ổ đĩa : [ALL] [CHECKONLY] /Autoflx [/NoSave] [/Custom] [/Surface] [/mono] [/NoSummary]].

7. **SHOWMEM.ASM**

SHOWMEM là một chương trình in ra bản đồ của bộ nhớ RAM.DOS, quản lý RAM bằng các MCB(Memory Control Block: khối điều khiển bộ nhớ).

SHOWMEM là một chương trình ngắn nhưng rất hữu dụng.

- Cho biết địa chỉ các chương trình thường trú. Như thế có thể tiến hành khảo sát “động” về chúng

- Dựa vào các MCB có thể lần tìm một số thông tin cần thiết đang nằm trong bộ nhớ

- Phát hiện Virus. Nếu máy bị nhiễm virus thì khi gọi SHOWMEM, Bạn sẽ thấy một trong các MCB hay số A000 đã thay đổi. Các file-virus, loại virus gắn vào tập tin, dễ phát hiện vì chúng thường làm tăng kích thước tập tin, nhưng các Boot-virus khó phát hiện hơn, vì ẩn náu trong sector đầu tiên của đĩa, là Boot sector cho đĩa mềm và Master Boot cho đĩa cứng, nơi ít ai chú ý. Nhưng may mắn là các Boot-virus đều chiếm vùng bộ nhớ cao, trước khi DOS được nạp vào(chương trình LOCK.ASM cũng làm thế) do đó bạn sẽ thấy phần bộ nhớ dành cho DOS bé hơn A000(bằng 9FC0 hay 9F80 chẳng hạn). Chương trình LOCK.ASM cho phép vừa giấu vừa bảo vệ đĩa cứng khỏi các Boot-virus này. Xem lượng bộ nhớ giành cho DOS có bị giảm so với bình thường, nếu không giảm thì kết thúc chương trình và có thể khẳng định máy chưa bị nhiễm Boot-virus. Nhưng còn file-virus, bạn nên đặt SHOWMEM ở dòng cuối cùng trong tập tin AUTOEXEC.BAT để khi thấy có MCB bị sửa hoặc số A000 nhỏ đi một cách khó hiểu thì kết luận được máy đang nhiễm file-virus. Bạn phải lấy một chương trình antivirus để trị virus hoặc phải duyệt tất cả các tập tin được CONFIG.SYS và AUTOEXEC.BAT gọi đến, nhằm phát hiện tập tin bị lây nhiễm.

- Phải tìm cho ra giá trị ROM-BIOS chuẩn vì có thể virus đã chặn ngắt 13 trước DOS và làm lệnh INT31h hoạt động sai. Sau khi phục hồi ngắt 13 chuẩn bạn có thể yên tâm dùng lệnh INT13h để so sánh Master Boot hiện tại với một bản lưu chuẩn của nó. Nếu thấy

giống, thì chắc là trình điều khiển thiết bị này đã nhiễm loại file-virus. Bạn phải thay nó. Nếu thấy khác mà không giải thích được, thì có nghĩa máy đang nhiễm Boot-virus: bạn chỉ việc ghi đè bản lưu chuẩn lên Master-Boot rồi dùng lệnh `JMP FFFF:0` để khởi động lại.

8. LOCK.ASM

LOCK.ASM hợp dịch ra dạng LOCK.COM, cần tập tin FLOOPBOOT.BIN, PUTBACK.BIN, NEWPART.BIN. Cũng cần thư viện các macro STDMAC.INC, hãy lấy một đĩa mềm đã Format bằng DOS vào ổ đĩa A rồi cho chạy chương trình LOCK. LOCK sẽ thực hiện các công việc chính:

- Ghi FLOPBOOT.BIN lên Bootsector của đĩa mềm
- Lưu Master Boot gốc đĩa cứng vào sector thứ 5 của đĩa mềm
- Ghi PUTBACK.BIN vào các sector thứ 2 và của đĩa mềm
 - Ghi NEWPART.BIN lên Master Boot của đĩa cứng vào sector thứ 4 của đĩa mềm.

Chương trình LOCK.ASM dùng để dấu đĩa cứng nếu không khởi động bằng đĩa mềm đặc biệt trên_ không thể tạo bản sao bằng lệnh DISKCOPY của DOS. Ngoài ra nó còn được thiết kế để tiêu diệt các Boot-virus. Mỗi khi đĩa cứng bị nhiễm các Boot-virus và bạn khởi động máy bằng đĩa mềm đặc biệt thì sẽ thông báo hỏi bạn có muốn ghi đè Master Boot không? Hơn nữa, LOCK cũng mã hoá các thông tin quan trọng trong Master Boot để tránh sự xâm nhập đĩa cứng ngoài ý muốn của bạn.

IV. Quản lý bộ nhớ trong.

Dưới quan điểm của DOS, bộ nhớ trung tâm của PC, mà kích thước của nó có thể đạt tới 640 KB, được chia làm 2 vùng:

Vùng thứ nhất được DOS quản lý, và do vậy ta gọi nó là vùng Hệ Điều Hành, nó bắt đầu từ ô nhớ thấp nhất, có nghĩa là ô nhớ 000:000, nó chứa bảng vector ngắt cũng giống như một số bảng, vùng đệm, các biến trong và phần mã nội trú của hệ điều hành. Ngoài ra còn có các chương trình điều khiển thiết bị được phép vào hệ thống, và chúng ta có thể gọi như các gọi một hàm nào đó của DOS. kích thước của vùng nhớ này phụ thuộc vào thể hệ của DOS và kích thước của các chương trình điều

khởi thiết bị được cài đặt và một số yếu tố khác. ví dụ như số lượng các vùng đệm đĩa.

địa chỉ	Nội dung	Kiểu
+00h	ID (“Z” = MCB cuối cùng, “M”= còn có các MCB tiếp theo)	1 byte
+01h	Địa chỉ đoạn của PSP tương ứng.	1 từ
+03h	Số lượng các paragraph trong vùng nhớ được cung cấp.	1 từ
+05h	Không dùng	1 từ
+10h	Vùng nhớ được cấp.	11 byte
	Tổng độ dài	16 bytes

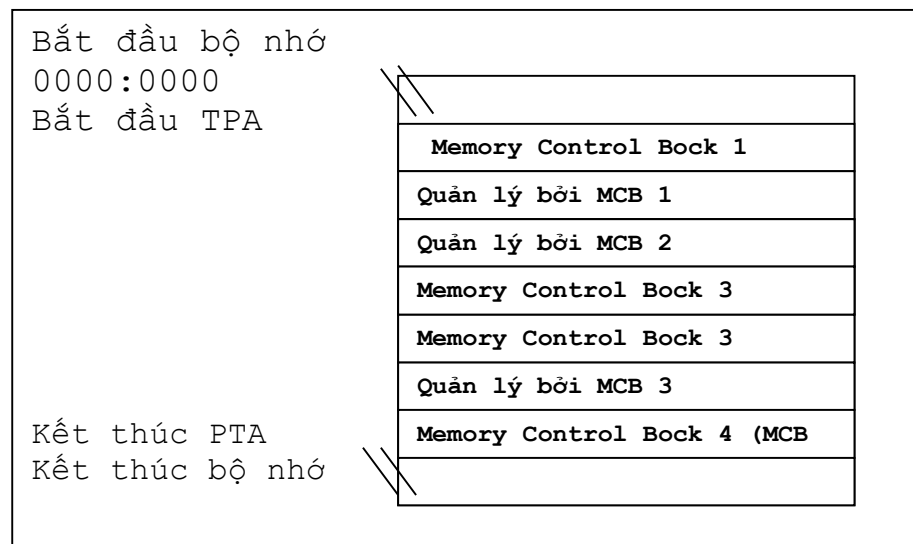
Vùng thứ 2 lớn hơn nhiều và nó được gọi là Transient Program Area (TPA) - vùng chương trình tạm thời, có nghĩa là vùng của các chương trình cần chạy cũng như các khối môi trường tương ứng, và nó bắt đầu ngay sau vùng Hệ Điều Hành. Tùy theo yêu cầu bộ nhớ của các chương trình, DOS cung cấp cho mỗi chương trình một vùng nhớ, vùng nhớ này được quản lý nhờ một khối dữ liệu đứng ngay trước vùng nhớ. Khối dữ liệu này gọi là Memory Control Block (MCB) – khối điều khiển bộ nhớ. Nó luôn có kích thước là 16 byte (1 paragraph), nó luôn bắt đầu từ một địa chỉ là bội nguyên của 16 và đứng ngay trước vùng nhớ được cung cấp cho chương trình.

Trong các hàm cung cấp bộ nhớ, DOS luôn làm việc với địa chỉ đoạn của MCB có thể dễ dàng tính được bằng cách lấy địa chỉ đoạn của vùng nhớ trừ 1.

Khối điều khiển bộ nhớ gồm 3 trường:

Trường thứ 1 : Người sáng lập MS – DOS Mark Zbikowski quy định trường này chỉ chứa một trong hai ký tự, “M” chỉ ra rằng sau MCB này còn có MCB khác, trong khi “Z” có nghĩa đây là MCB cuối cùng trong bộ nhớ.

Trường thứ 2: chứa địa chỉ đoạn PSP (File Control Block) của chương trình. Địa chỉ này chỉ có nghĩa khi vùng nhớ được cấp là khối môi trường của một chương trình. Trường này chỉ ra địa chỉ PSP của chương trình đó, và như vậy, tạo ra một khối liên hệ. Ngược lại, nếu vùng nhớ là một PSP, thì trong đại đa số các trường hợp, trường hợp này chính vùng nhớ của chương trình.



Quản lý bộ nhớ bằng các MCB

Trường thứ 3: Trường này của MCB là đáng quan tâm hơn cả nó chỉ ra kích thước của vùng nhớ được cấp theo đơn vị paragap. Thực vậy, vì MCB tiếp theo bắt đầu ngay sau vùng nhớ được cấp, nên trường này chỉ ra khoảng cách đến MCB tiếp theo. Mỗi một MCB chỉ ra MCB tiếp theo một cách gián tiếp, do vậy, ta nhận được một chuỗi cho phép tìm lại danh sách tất cả các MCB. Khi một chương trình được nạp và thực hiện bằng hàm EXEC của DOS, trước tiên hàm này yêu cầu hai vùng dữ liệu thông qua một trong các hàm khác của DOS, trước tiên hàm này yêu cầu hai vùng dữ liệu thông qua một trong các hàm khác của DOS. Vùng thứ nhất dùng để nhận khối môi trường, trong khi vùng thứ hai nhận bản thân chương trình và cả PSP tương ứng. Tuy nhiên, hàm EXEC rất khó đánh giá kích thước vùng nhớ cần dành cho chương trình. Đối với chương trình COM lại còn khó hơn nữa, vì nó được lưu trong đĩa dưới dạng như là trong bộ nhớ và nó không chứa thêm một thông tin nào về bản thân nó. Do vậy, DOS tự động dành cho cả bộ nhớ tự do cho một chương trình COM.

Phương pháp này đã chứng minh ưu việt của mình trong giai đoạn khởi tạo DOS, nhưng nó nảy sinh các vấn đề. Trong thế hệ đầu tiên của DOS, không thể có quá một chương trình trong bộ nhớ, còn ngày nay thì ngược lại, thường xảy ra là một chương trình nạp một chương trình khác vào bộ nhớ làm chương trình nội trú. Mà điều này sẽ không thực hiện được nếu không còn bộ nhớ tự do, ví dụ sau khi chương trình COM được nạp, cho nên cần phải làm sao để mọi chương trình COM giải phóng bộ nhớ mà không cần mà ngay sau khi được nạp vào bộ nhớ.

Một chương trình COM chỉ có thể được nạp nếu còn đủ bộ nhớ tự do. Có nghĩa là bộ nhớ ít tự do nhất cũng bằng kích thước chương trình cộng 256 byte của PSP và ít nhất 2 byte cho ngăn xếp. Cũng cần quan tâm đến việc, bộ nhớ dành cho chương trình COM phải thực sự đủ cho nó. Trên thực tế, ít khi xảy ra một trường hợp một chương trình nào đó

có thể hoạt động trong điều kiện tối thiểu như trên, vì chương trình nào lại chỉ có 2 byte ngăn xếp ?

Đối với chương trình EXE thì lại khác, nó có một loạt các thông tin đứng trước chương trình. Các thông tin này được chương trình liên kết LINKER cung cấp, chúng luôn đi kèm theo chương trình, ngay cả ở trong đĩa. Do vậy, hàm EXEC có thể suy ra từ các thông tin này, cần bao nhiêu bộ nhớ cho mã, dữ liệu và ngăn xếp. Khối thông tin này của một chương trình EXE gọi là Header còn chứa các thông tin khác cho biết bao nhiêu bộ nhớ bổ sung cần trừ tính cho chương trình EXE. Tuy nhiên, các thông tin này không chỉ ra chính xác cần bao nhiêu byte, mà nó xác định giới hạn dưới và trên của bộ nhớ bổ sung. Chương trình nạp EXE cố gắng dành bộ nhớ cho chương trình theo giới hạn trên, nếu đó không thể được, nó dành bộ nhớ theo giới hạn dưới, hoặc dành tất cả bộ nhớ còn lại cho chương trình. Nhưng nếu giới hạn dưới cũng không được thoả mãn, thì thao tác nạp sẽ dừng lại và điều khiển được trả lại cho chương trình đã gọi hàm EXEC.

Sau khi thực hiện chương trình, điều khiển cũng được trao cho chương trình gọi, nhưng trước đó, hàm EXEC được giải phóng vùng nhớ đã được phân phối, với điều kiện là chương trình gọi không phải là nội trú.

V. Quản lý file

1. File là gì?

MS DOS hệ điều hành đơn nhiệm, chỉ xử lý được một chương trình tại một thời điểm, sử dụng giao diện dòng lệnh (Người dùng phải gõ lệnh tại đầu nhắc).

DOS là hệ điều hành quản lý đọc ghi đĩa, các thông tin được lưu trữ, quản lý trên đĩa theo hệ thống Thư mục (Directory) và Tập tin (File).

Tập tin (File)

Là đơn vị lưu trữ thông tin đã được mã hoá. Trong DOS, các File được lưu trữ trên đĩa và được nhận diện bằng tên xác định. Tên File được lưu trữ dưới dạng Text. Cấu trúc của tên File như sau:

FILENAME.EXT

F	I	L	E	N	A	M	E	E	X	T
---	---	---	---	---	---	---	---	---	---	---

Phần tên File chứa tối đa 8 ký tự, không chứa các ký tự đặc biệt. Tên File thường được đặt có ý nghĩa nhất định để có thể xác định được ý nghĩa của nội dung File ví dụ như File có tên Readme cho ta biết rằng

Phần mở rộng (Extension) gồm 3 ký tự không chứa ký tự đặc biệt thường được biểu thị loại File. Ngăn cách với tên File bằng dấu chấm

Loại File: Các File trong DOS thường được phân loại như sau bằng phần mở rộng của File:

Phần mở rộng	Loại File
.COM (Command)	Các File chứa mã lệnh.
.EXE (Executable)	Các File có thể xử lý như các lệnh, chương trình.
.BAT (Batch)	File dạng Text chứa các dòng gọi lệnh xử lý tuần tự.
.SYS (System)	File hệ thống.
.TXT (Text)	File chứa các ký tự thường là các File hướng dẫn.
Nói chung, các File chính là các loại lệnh và File text.	

Ký tự thay thế: thường được dùng để tìm kiếm, hiển thị, xoá, sao chép...

Ký tự	Chức năng
*	Thay thế cho một nhóm các ký tự. Ví dụ *.txt biểu diễn các File có tên File bất kỳ và có phần mở rộng chung là .TXT.
?	Thay thế cho một ký tự tại vị trí nó đứng. Ví dụ ????.doc biểu diễn các File có tên bất kỳ gồm 3 ký tự và phần mở rộng là .DOC.

Thuộc tính (Attribute): File thường có một số thuộc tính mà người dùng hoặc chương trình có thể áp đặt cho nó. Các thuộc tính có thể có là:

Thuộc tính	ý nghĩa
R (Read Only)	Thuộc tính chỉ đọc (không được sửa chữa nội dung File)
A (Archive)	Thuộc tính lưu trữ.
S (System)	Thuộc tính hệ thống (các File dành cho hệ thống).
H (Hidden)	Thuộc tính ẩn.

Đường dẫn (path): Chỉ dẫn tới thư mục hoặc File xác định, ký tự phân cách trong đường dẫn là “\”. Ví dụ **C:\Windows\Fonts\vntime.ttf** xác định File vntime.ttf nằm trong các thư mục có đường dẫn lần lượt là **C:\Windows\Fonts**

2. Quản lý File

Để quản lý File hệ điều hành cung cấp cho người lập trình các hàm cơ bản nhất. Chủ yếu là xung quanh các hàm như:

- + Tạo và xoá File.
- + Mở và đóng File.
- + Đọc và ghi File.
- Các hệ điều hành hoàn thiện như DOS còn chứa một loạt các hàm khác để quản lý File. Trong DOS, đó là các hàm cung cấp các thông

tin quan trọng về một File và các hàm đặc biệt, chẳng hạn như hàm đổi tên File. Một trong những đặc điểm của DOS liên quan đến vấn đề này là: mỗi một hàm tồn tại dưới hai dạng. Nguyên nhân là do sự tương thích. Đó là tương thích CP/M thì có một hàm tương ứng tương thích UNIX.

+ Tương thích CP/M: Các hàm tương thích với CP/M gọi là FCB, vì chúng dựa trên một cấu trúc dữ liệu gọi là FCB_ File Control Block (khối quản lý File). DOS sử dụng cấu trúc dữ liệu này trong khi làm việc với một File để lưu vào đây các thông tin quan trọng về File đó. Tuy nhiên, vai trò của người lập trình là phải dành chỗ cho cấu trúc dữ liệu này trong chương trình của mình, để thông qua nó mà gọi các hàm như: mở, đóng, đọc, ghi...

Thế nhưng, các hàm FCB chỉ cho phép truy nhập đến các File nằm trong thư mục hiện thời, lý do là:

- Các hàm FCB được xây dựng là do vấn đề tương thích với các hàm của CP/M.
- Vì CD/M không là hệ thống File hình cây, nên các hàm FCB không đảm bảo được các đổi mới của DOS 2.0.

+ Tương thích UNIX: Khác với việc truy nhập File thông qua FCB. Các hàm tương thích với UNIX sử dụng một trị số gọi là Handle như là một mã để truy nhập đến một File xác định. Mã này được truyền cho chương trình gọi ngay sau khi File được mở. Tuy nhiên, DOS sẽ cất 1 lượng nhất định các thông tin về File và lưu trong 1 cơ sở dữ liệu đặt ở vùng nhớ mà DOS chiếm, chứ không phải trong vùng nhớ dành cho chương trình.

VI. HỆ THỐNG VÀO RA.

1. Thiết bị nhập dữ liệu.

a. Bàn phím.

Bàn phím máy tính là thiết bị mà người sử dụng trực tiếp nhập (input) thông tin vào RAM dưới dạng mã ascii.

Có nhiều kiểu bàn phím khác nhau, mỗi kiểu có thể sắp đặt vị trí, chữ dấu khác nhau... nhưng chung quy chúng đều phải chứa đầy đủ các bộ phận và các phím sử dụng.

+ Hiện nay trên thị trường có hai loại bàn phím:

- Bàn phím tiêu chuẩn (Standar keyboard): IBM AT đầu tiên đưa ra vào tháng 8 năm 1984 với 84 phím.

- Bàn phím mở rộng (Extended keyboard): Đã được IBM đưa vào tháng 4 năm 1986, cùng với sự đưa vào loại IBM AT 8 MHz với 101 phím.

+ Các đèn chỉ thị bàn phím: Các đèn chỉ thị ở góc bên phải bàn phím là: NumLock, CapsLock, ScrollLock. Khi ấn các phím đèn chỉ thị trên thì chúng tương ứng sẽ sáng.

+ Các bộ phận chính của bàn phím:

-Vùng phím đánh máy: Đây là phần chính của bàn phím, được dùng để ghi nhập phần lớn dữ liệu.

Các phím ký tự: Gồm các phím chữ: a – z , A – Z, các phím số: 0 – 9 và các phím ký tự đặc biệt: +, / , ?,...

Các phím điều khiển: Phím Enter: Được ấn sau khi đưa vào một lệnh, khi đó máy mới bắt đầu thực hiện lệnh đó. Trong soạn thảo phím này còn được dùng để đưa con trỏ về đầu hàng dưới.

Phím BackSpace: Di chuyển con trỏ sang trái một vị trí (so với vị trí hiện tại) đồng thời xoá ký tự tại vị trí đó. Phím này dùng để sửa một lỗi gõ sai trên dòng hiện thời.

Phím Cách (Space Bar): Dịch chuyển con trỏ sang bên phải một ký tự. Phím này dùng để đưa một ký tự trống vào dòng lệnh.

Phím (Tab): Đẩy con trỏ nhảy qua một số ô sang phải, khoảng cách giữa các ô chính là khoảng cách giữa hai Tab.

Phím (Shift): Có hai phím nằm ở bên trái và bên phải của bàn phím cùng chức năng như nhau. Phím này được dùng để sử dụng phần phía trên ký hiệu ghi trên phím hoặc để đưa vào chữ hoa (khi đèn CapsLock không sáng), hay đưa vào chữ thường (khi đèn CapsLock sáng).

Phím (CapsLock): Dùng để đưa vào chữ in hoa khi đèn CapsLock sáng. Các phím ký tự đặc biệt không chịu ảnh hưởng của phím này.

Phím Ctrl: Phím này luôn đi kèm với một hoặc hai phím khác để thực hiện một chức năng đặc biệt hoặc một lệnh.

Phím Alt: Phím này được sử dụng theo nhiều phương cách khác nhau, theo từng phần mềm ứng dụng. Nó cũng được sử dụng với một hoặc hai phím khác.

-Khung phím số:

Khi đèn chỉ thị NumLock sáng các số được xếp như trong máy tính điện tử bỏ túi.

Khi đèn chỉ thị NumLock tắt các số ở chế độ “điều khiển con trỏ”.

-Bộ phím điều khiển con trỏ:

Vị trí: Nằm giữa “vùng phím đánh máy” và “khung phím số”.

Dùng để điều khiển con trỏ bằng các phím mũi tên tương ứng (nhưng không xoá đi ký tự tại vị trí con trỏ).

Phím INSERT: Dùng để chèn một hay nhiều ký tự trên cùng một hàng, khi đó các phím đằng sau sẽ bị đẩy lùi sang bên phải.

Phím DEL: Dùng để xoá ký tự ngay vị trí con trỏ, khi đó tất cả các ký tự ở bên phải con trỏ sẽ di chuyển sang trái một ký tự.

Phím HOME: Di chuyển con trỏ về đầu dòng.

Phím END: Di chuyển con trỏ về cuối dòng.

Phím PgUp: Dùng để di chuyển con trỏ lên trên một trang màn hình.

Phím PgDn: Dùng để di chuyển con trỏ xuống một trang màn hình kế tiếp.

-Nhóm các phím chức năng:

Phím F1: Trong DOS , phím này dùng để sao một ký tự từ trong vùng đệm ra dòng lệnh hiện thời.

Phím F2: Trong DOS , phím này dùng để sao tất cả các ký tự từ vị trí con trỏ tới trước ký tự chỉ ra trong vùng đệm ra dòng lệnh hiện thời.

Phím F3: Trong DOS , phím này dùng để sao toàn bộ các ký tự từ trong vùng đệm ra dòng lệnh hiện thời.

Phím F4: Trong DOS , phím này dùng để bỏ qua các ký tự từ vị trí con trỏ tới trước ký tự chỉ ra trong vùng đệm.

Phím F5: Trong DOS , phím này tạo ra vùng mới trong vùng đệm.

Phím F6: Trong DOS , phím này có tác dụng ghi mã ASCII của ký tự kết thúc file EOF trên vùng đệm.

Phím F11, F12: Có thể cài đặt được bằng các lệnh trong DOS.

-Các phím có chức năng đặc biệt:

Phím PrintScreen(hay SysRq): Dùng để điều khiển máy in, in thông tin trên màn hình.

Phím ESC: Trong DOS nó dùng để bãi bỏ việc đưa vào một lệnh mà không làm thay đổi nội dung thông vùng đệm.

Phím Pause (hay Break): Dùng để dừng trôi màn hình hiển thị.

b. "Chuột" (Mouse).

Trên hệ điều hành DOS chuột được xử dụng một cách rất hạn chế.

Chuột là một thiết bị hình dáng như con chuột, với kích cỡ như lòng bàn tay, được nối với máy tính bằng dây điện (“đuôi” chuột) và chuột di chuyển trên mặt bàn hay bàn di chuột riêng(Mouse pad) để làm chuyển động con trỏ trên màn hình.

Chuột được dùng để định vị con trỏ, cử động của chuột tạo thành một loạt các xung và các xung này được gửi đến PC. Ngoài chuột ra còn có một số thiết bị điều khiển con trỏ khác như chuột quang học, phiến nhân, cần điều khiển – Joystick...

d. Một số thiết bị đọc.

- Thiết bị đánh dấu quang học: Dùng ánh sáng phản xạ để xác định vị trí đã được đánh dấu.

- Thiết bị đọc mã vạch: Dùng ánh sáng để kiểm tra mã sản phẩm(UPC).

- Thiết bị đọc chữ in từ tính: Để đọc các con số theo mẫu được in bằng một loại mực đặc biệt có từ tính dùng trong việc kiểm tra.

- Cây đũa thần: Là thiết bị dùng tia sáng để đọc các kí tự được viết bằng một thiết bị đánh dấu đặc biệt trên vùng dành riêng.

- Cây viết máy tính: Để nhập dữ liệu người ta viết trực tiếp lên màn hình phẳng của máy tính.

e. Các thiết bị số hoá.

- Máy quét ảnh: Là thiết bị có thể biến những hình ảnh thành những thông tin số hoá đại diện cho nó.

- Máy ảnh số: Dùng để chụp ảnh bên ngoài và lưu hình ảnh dưới dạng thông tin số.

- Máy quay phim số: Nó lưu hình ảnh động và cả âm thanh nữa. ứng dụng trong việc truyền thông đa phương tiện hay hội đàm qua màn ảnh.

- Thiết bị âm thanh số hoá: Với các thiết bị khác nhau như loa, micro, các nhạc cụ điện tử... âm thanh được đưa vào dưới nhiều dạng khác nhau và đều được chuyển thành dạng số.

- Thiết bị cảm ứng: Được dùng để đưa dữ liệu bên ngoài vào máy tính dưới dạng số hoá như nhiệt kế điện tử, máy dự báo thời tiết...

2. Thiết bị đầu ra

a. Màn hình máy tính (Monitor)

Là thiết bị sẽ cho ta thấy những ký tự mà ta gõ trên bàn phím hoặc các thông điệp từ máy tính. Những thế hệ màn hình mới có thể thể hiện chi tiết các hình ảnh cũng như chữ, số và các ký hiệu với đủ loại màu sắc khác nhau, thường gọi là màn hình màu, tên gọi như vậy để phân biệt loại màn hình đơn sắc dùng cho các hệ máy cũ (loại máy XT).

Có hai chữ Monitor và display mà người ta hay dùng lẫn lộn, dù đó là hai khái niệm khác nhau. Display- Màn hiển thị: là thiết bị hiển thị tạo ra hình ảnh từ tín hiệu video, như ống phóng điện tử hay màn hình tinh thể lỏng hay bất kỳ thiết bị hiển hình nào khác. Còn monitor là toàn bộ các mạch phụ trợ và cả màn hiển thị, tất cả lắp trong vỏ máy, người ta thường gọi monitor là màn hình.

Màn hình có hai loại chính là: màn hình kiểu thiết kế giống như tivi dùng các bóng đèn tia điện tử cathode CRT (cathode ray cathode) và màn hình tinh thể lỏng LCD (liquid crystal display). Tuy rằng LCD là loại màn hình phẳng nhỏ gọn hơn CRT nhiều, nhưng giá cả của LCD lại quá cao so với CRT. Các màn hình CRT ngày nay cho chất lượng hình ảnh tốt hơn màn hình LCD. Thông thường người ta chỉ dùng LCD cho các máy tính dạng xách tay. Màn hình CRT này về cơ bản gồm một bóng đèn hình lớn chứa ba ống phóng điện tử cho ba màu đỏ, xanh lá cây và xanh dương. Ba màu cơ bản này sẽ tạo ra được mọi màu khác cần hiển thị. Một điểm ảnh sẽ do 3 yếu tố RGB hợp thành tạo ra màu sắc cần thiết.

Màn hình được nối kết với máy tính thông qua bộ điều hợp hiển thị – video adapter hay display adapter. Nó còn có tên gọi là các màn hình – display card, video card. Bộ điều hợp hiển thị là một bảng mạch điện tử được cắm trong máy tính ở khe cắm mở rộng. Hình ảnh là thông tin được lưu ở bộ nhớ màn hình –VRAM. Khả năng của bộ điều hợp hiển thị sẽ quyết định tốc độ làm tươi hình ảnh, tốc độ hiện hình, độ phân giải, mức độ màu có thể hiển thị. Bộ điều hợp hiển thị được phân loại theo độ rộng BUS dữ liệu của nó:

Bộ điều hợp hiển thị 32 bit: có đường dẫn dữ liệu nội bộ 32 bit – có xử lý 4 byte dữ liệu cùng lúc. Bộ điều hợp hiển thị: có đường dẫn dữ liệu nội bộ 64 bit – có thể xử lý 8 byte dữ liệu cùng lúc. Bộ điều hợp hiển thị 128 bit: có đường dẫn dữ liệu nội bộ 128 bit – có thể xử lý 16

byte dữ liệu cùng lúc. Đường dẫn dữ liệu càng rộng thì khả năng của bộ điều hợp càng cao. Do đó, loại 32 bit xem ra đã lỗi thời, mức chuẩn hiện nay là loại 64 bit để có thể hiển thị phân giải cao, lên đến 1280 x 1024 dpi. Còn loại 128 bit có tốc độ cao thích hợp với yêu cầu công tác nhiều hình vẽ như thiết kế đồ họa chẳng hạn.

Kích thước của màn hình cũng gần như một cái tivi. Thông số dùng để phân loại màn hình máy tính và tivi được quy định giống nhau – là độ dài đo được của đường chéo màn hiển thị. Một máy tính để bàn thông thường có màn hình từ 14 đến 17 inch. Hình ảnh hiện trên màn hiển thị là sự kết hợp của nhiều chấm nhỏ – gọi là điểm ảnh – pixel.

Độ phân giải của màn hiển thị thông thường là 72 điểm trong một inch cho mỗi chiều ngang và dọc. (Đơn vị tính độ phân giải viết tắt là dpi - điểm trong một inch: dots per inch) Độ phân giải càng cao, các điểm ảnh càng sát lại với nhau, hình ảnh càng mịn hơn và đẹp hơn.

Còn một cách nói khác về kích thước màn hình, thay vì nói về độ dài đường chéo thực sự của màn hiển thị, người ta nói về mức độ phân giải có thể có của màn hiển thị. Nếu nói màn hình 800 x 600 tức là chiều ngang gồm 800 điểm, chiều dọc gồm 600 điểm.

Yếu tố khác nói về khả năng card màn hình là độ sâu màu có thể hiển thị – color depth. Ví dụ như, màn hình đơn sắc thì thể hiện 2 bit cho mỗi điểm, mỗi bit có thể hiển thị hai màu hoặc màu này hoặc màu kia (các màn hình đơn sắc ban đầu thường có màu xám, xanh hay nâu). Nếu mỗi điểm có 8 bit màu thì có khả năng thể hiện 256 màu -28 - đây là khả năng thông thường mà hầu hết tất cả các máy tính hiện nay đều thể hiện được. Loại cao cấp hơn có thể chấp nhận 24 bit màu, tức là khoảng hơn 16 triệu màu (16.777.216 màu – 2²⁴), hiện nay trên máy vi tính đã có thể thể hiện 32 bit màu – chấp nhận 4.294.967.296 màu! Số màu có thể hiển thị càng nhiều thì hình ảnh càng trung thực sắc nét và sống động – và chắc chắn là đẹp hơn hình ảnh trên tivi nhiều lần.

o Các loại màn hình LCD

Những loại màn hình LCD bạn thấy hiện nay, về cơ bản có thể được hình dung như hệ thống van điều tiết lượng ánh sáng phát ra từ một nguồn sáng phía sau, bản thân LCD không tự phát sáng như màn hình CRT. Trong màn LCD, ánh sáng đi qua một vài lớp cực mỏng

được cấu tạo từ vật liệu tinh thể lỏng có tính phân cực. Chất tinh thể lỏng nằm giữa hai mặt kính mỏng tạo nên màn hình. Hình ảnh hiển thị trên màn hình được xác định bởi một hệ thống lưới (ma trận) bao gồm nhiều điện cực, điều tiết lượng ánh sáng đi qua mỗi điểm của lưới (pixel).

Hiện nay, màn hình LCD được phân ra làm hai loại chính: thụ động (passive, bao gồm cả dual-scan) và tích cực (active- TFT). Sự khác biệt cơ bản giữa hai loại là cách thức điều khiển mỗi điểm riêng biệt. Trong màn hình ma trận thụ động, mỗi hàng điểm ngang do một transistor điều khiển. Vì vậy, mỗi điểm được điều khiển một cách thụ động. Trong khi đó, đối với màn hình LCD tích cực, mỗi điểm có một transistor riêng để điều khiển, và đối với màn hình màu thì có đến 3 transistor cho mỗi điểm (mỗi transistor cho một màu cơ bản). Màn hình thực chất là một tập hợp (array) hình chữ nhật bao gồm nhiều transistor ở dạng lớp mỏng(thin film). Vì vậy còn có tên gọi là TFT (Thin film transistor).

Màn hình LCD tích cực cho hình ảnh nhanh và đẹp hơn nhiều so với màn hình thụ động bởi khả năng điều khiển lượng ánh sáng đi qua mỗi điểm tốt hơn. Bạn có thể nhận sự khác biệt ở đường nét, độ tương phản, tính trung thực màu cũng như khả năng phản ứng nhanh của màn hình đóng vai trò rất quan trọng trong hiển thị video chuyển động thực. Trong khi các loại màn hình ma trận tích cực mới nhất có khả năng tái hiện video với tốc độ 20 – 30 khung hình trong một giây thì màn hình ma trận thụ động, kể cả loại dual-scan chỉ đạt tới mức 5 khung hình một giây. Bởi vậy, nếu muốn chơi video MPEG trên máy notebook, bạn không nên chọn màn hình thụ động. Tuy nhiên, do công nghệ phức tạp trong quy trình sản xuất, cũng như tỷ lệ sản phẩm đạt chất

lượng không cao (chỉ cần 3 trong số hàng trăm nghìn transistor bị hỏng, màn hình coi như bị loại bỏ), màn hình ma trận tích cực đắt hơn rất nhiều so với màn hình thụ động.

Công nghệ Dual - scan là một cố gắng nhằm tăng cường hiệu năng của màn hình thụ động mà không làm tăng giá thành. Về cơ bản, theo công nghệ này, màn hình được chia thành hai nửa, với hai transistor cho mỗi hàng. Như vậy, thời gian phản ứng, cũng như độ tương phản và độ

sáng có tăng lên một ít. Tuy nhiên, so với màn hình tích cực, màn hình thụ động vẫn còn kém một khoảng xa.

Hiện tại, các loại màn hình dual - scan hay TFT có độ phân giải 640 x 480 hay 800 x 600. Màn hình 640 x 480 thường có kích thước 9,5 hay 10,4 inch (theo đường chéo). Loại 800 x 600 có kích thước từ 10 đến 12 inch. Giá của màn hình 800 x 600 đắt hơn loại 640 x 480 không quá 50 USD.

b. Máy in.

Máy in được gắn với máy tính sẽ là thiết bị xuất có giá trị lưu trữ bởi các bản in giấy. Máy in có nhiều loại và được chia thành 2 nhóm chính: Máy in gõ hay máy in không gõ (impact printer ở nonimpact printer)

- Máy in gõ: Là các máy in theo dòng hay theo ma trận điểm. Các máy in kim là loại này, có đặc điểm là tốc độ in chậm, ồn ào, độ phân giải thấp cho chất lượng in ấn trung bình. Máy in này dùng một đầu kim chạy suốt chiều ngang giấy để ấn các kim xuống giấy (Qua lớp băng mực) theo tín hiệu điều khiển, nhiều lần như vậy tạo nên bản in. Số đầu kim của máy càng cao thì tốc độ phân giải đạt được càng cao, các loại máy in 9,18 kim hầu như không còn được sử dụng nữa, thông dụng nhất của máy in loại này có lẽ là có thể in trên khổ giấy lớn mà giá máy rẻ và có thể nhân thành nhiều bản bằng giấy than do sự gõ truyền lực.

VII. Hệ thống dịch

Hệ điều hành ms-dos là hệ điều hành phiên bản đầu tiên của máy tính, hệ điều hành này sử dụng chế độ giao diện lệnh (command mode). ở đây người sử dụng (user) ra lệnh cho máy tính qua lệnh được nhập vào từ bàn phím với vậy mà lệnh là cửa ngõ duy nhất giúp cho người can thiệp vào hệ thống máy tính. lệnh trong ms-dos gồm hai loại là lệnh nội trú (internal command) và lệnh ngoại trú (external command) trong đó lệnh nội trú là lệnh được nạp thường trú vào bộ nhớ ta có thể sử dụng trực tiếp lệnh này bằng cách chỉ cần gõ lệnh, còn đối với lệnh ngoại trú thì mỗi khi chạy ta phải nạp vào bộ nhớ từ nguồn chứa tệp lệnh đó. về thực tế 1 lệnh trong dos gồm 3 phần là : tên lệnh ; điều kiện và tham số, lệnh trong dos là một chuỗi ký tự đọc được và giữa các phần được ngăn cách nhau bởi dấu cách (ký tự trắng) và kết thúc bằng enter, mỗi lần enter đồng nghĩa với một lệnh sẽ được tải lên system bus để vào hàng đợi lệnh chờ CPU xử lý, hệ điều hành dos cũn được gọi là hệ điều hành khai thác đĩa do đủ khả năng quản lý và sử dụng tài nguyên ở đây là rất chặt chẽ riêng đối với việc quản lý chuỗi ký tự nhập vào thì dos sử dụng hàm nằm trong ngắt 21h, trong dos cũng phải có 1 thư viện các lệnh và tương ứng với nó là các chương trình

con để thi hành các lệnh đó .ngoài ra dos cũn xây dựng các trình biên dịch để cho người sử dụng can thiệp trực tiếp vào hệ thống thụng qua các chương trình được viết dưới dạng hợp ngữ , nhưng ở đây ta chỉ xét đến các lệnh cơ bản của dos đây là các lệnh nờu bật lờn bản chất khai thỏc đĩa của dos.

Cỳ phỏp tổng quỏt lệnh là:

Lệnh nội trú:

Lệnh đối tượng lệnh[[tham số] [...]] <enter>

Lệnh ngoại trú:

[địa chỉ] Lệnh đối tượng lệnh[[tham số] [...]] <enter

(trong đó địa chỉ lệnh là ổ đĩa:\thư mục chứa tập tin chương trình của lệnh ,tham số là các thi hành ,các tùy chọn theo đó lệnh thi hành theo yêu của các bạn theo thực tế.)

Chúng ta có thể tìm hiểu chúng bằng câu lệnh:

LỆNH/?<enter>

Hay

[địa chỉ] LÊNH /?<enter>

một số lệnh nội trú hay dùng gồm:

DATE :sửa đổi ngày hệ thống.

TIME :sửa đổi giờ của hệ thống.

CLS :xoá màn hình. .

VOL:hỏi tên đĩa.

PROMPT:thay đổi dấu đợi lệnh.

PATH:khai báo đường dẫn cho DOS.

SET:đặt biến môi trường cho DOS.

COPY:sao chép tập tin .

TYPE:đọc ra màn hình hay máy in nội dung của tập tin văn bản...

DIR :xen danh sách tập tin trên đĩa .

MKDIR:tạo thư mục mới .

RMDIR:xoá thư mục đã có .

CHDIR:chuyển sang thư mục .

Và các lệnh ngoại chú hay dùng:

LABEL:thay đổi nhãn đĩa.

ATTRIB:đặt hoặc chỉ ra thuộc tính tập tin.

CHKDSK:kiểm tra đĩa.

DISKCOPY:sao chép nguyên đĩa .

XCOPY :chép tập tin và cả thư mục.

FORMAT:định dạng đĩa.

TREE:xem sơ đồ danh sách tập tin và thư mục theo hình cây.

MORE:trìn bày thông tin theo tong trang.

MIRROR:tạo tập tin lưu trữ vùng hệ thống của đĩa.

UNDELETE:phục hồi tập tin vừa xáo.

UNFORMAT:phục hồi đĩa bị định dạng.

SORT:sắp thứ tự thông tin theo chữ hay số.

Ví dụ

a:\format c: /q/s

c:\dir /u/p.

c:\md folder

c:\cd folder

c:\rd folder

c:\copy con tentep.ex

c:\type type tentep.ex }

Ngoài ra cũn còc lệnh như xem cõy (tree),lễnhỏ thư mục bằng deltree DOS sẽ hiển thị thông báo lỗi nếu như bạn nhập vào một lệnh sai,ngày nay người ta trang bị thòm còc tiện óch như NC cú giao diện panel (cửa số) do đú việc thực hiện còc lệnh trong dos cú phần đơn giản hơn tuy nhiên chỉ khỏc về mặt giao diện cũn về còch thức điều hành thờ vãn khụng thay đỏi.

đánh giá công việc và kết luận

Qua quá trình học tập và nghiên cứu tiểu luận “Hệ điều hành MD – DOS”. Đây là một đề tài không khó so với các đề tài khác, nhưng với chút hiểu biết trong quá trình học tập và phạm vi đề tài của chúng em chưa được rộng lắm. Đề tài của chúng em còn hạn hẹp, tuy nhiên cũng đã làm rõ được một số vấn đề cơ bản của đề tài:

- Quản lý tiến trình.
- Quản lý bộ nhớ.
- Quản lý bộ nhớ phụ.
- Quản lý hệ thống vào ra.
- Hệ thống bảo vệ.
- Hệ thống giải thích lệnh.

Nhưng do thời gian còn hạn hẹp và còn thiếu một số kiến thức thực tiễn, cho nên bài làm của chúng em chắc chắn sẽ còn nhiều thiếu sót, không thể thiếu được trong quá trình làm bài.

Đối với chúng em là những sinh viên chuyên ngành công nghệ thông tin, tham vọng của chúng em là mang những gì đã học hỏi được của thầy cô khi ngồi trên ghế nhà trường. Vì vậy chúng em rất mong sự góp ý, phê bình của thầy để chúng em hiểu rõ về đề tài.

