Giáo trình tổng hợp những dữ liệu về file FAT1 và FAT2 trong hệ điều hành

Khi cần ghi nội dung của một file vào đĩa hoặc khi cần đọc nội dung của một file trên đĩa hệ điều hành phải dựa vào bảng FAT, nếu bảng FAT bị hỏng thì hệ điều hành không thể ghi/đọc các file trên đĩa. Do đó, hệ điều hành DOS tạo ra hai bảng FAT hoàn toàn giống nhau là FAT1 và FAT2, DOS sử dụng FAT1 và dự phòng FAT2, nếu FAT1 bị hỏng thì DOS sẽ sử dụng FAT2 để khôi phục lại FAT1. Điều không đúng với hệ thống file FAT32, FAT32 vẫn tạo ra 2 FAT như của DOS, nhưng nếu FAT1 bị hỏng thì hệ điều hành sẽ chuyển sang sử dụng FAT2, sau đó mới khôi phục FAT1, và ngược lại.

Hệ điều hành DOS tổ chức cấp phát động các cluster cho các file trên đĩa, sau mỗi thao tác cấp phát/ thu hồi cluster thì hệ điều hành phải cập nhật lại nội dung cho cả FAT1 và FAT2. Có thể hệ điều hành chỉ thực hiện cấp phát động cluster cho các file dữ liệu (có kích thước thay đổi), còn đối với các file chương trình, file thư viện, file liên kết động, … (có kích thước không thay đổi) thì hệ điều hành sẽ thực hiện cấp tĩnh cluster cho nó.

Bảng FAT bao gồm nhiều phần tử (điểm nhập/ mục vào), các phần tử được đánh địa chỉ bắt đầu từ 0 để phân biệt, địa chỉ cluster cũng có thể gọi là số hiệu của cluster. Giá trị dữ liệu tại một phần tử trong bảng FAT cho biết trạng thái của một cluster tương ứng trên vùng dữ liệu. Ví dụ, phần tử thứ 7 trong bảng FAT chứa giá trị 000h, giá trị này cho biết cluster thứ 7 trên vùng dữ liệu còn trống, có thể dùng để cấp phát cho một file. Phần tử thứ 5 trong bảng FAT chứa giá trị FF7h, giá trị này cho biết cluster thứ 5 trên vùng dữ liệu bị bad, không thể cấp phát được, …

Hệ điều hành DOS có thể định dạng hệ thống file theo một trong 2 loại FAT là FAT12 và FAT16. Mỗi phần tử trong FAT12 rộng 12 bít(1.5 byte), mỗi phần tử trong FAT16 rộng 16 bít(2 byte). Các đĩa hiện nay thường được DOS định dạng theo hệ thống file với FAT16. Sau đây là danh sách các giá trị dữ liệu được chứa tại các phần tử trong bảng FAT (số trong ngoặc dùng trong FAT16) và ý nghĩa của nó.

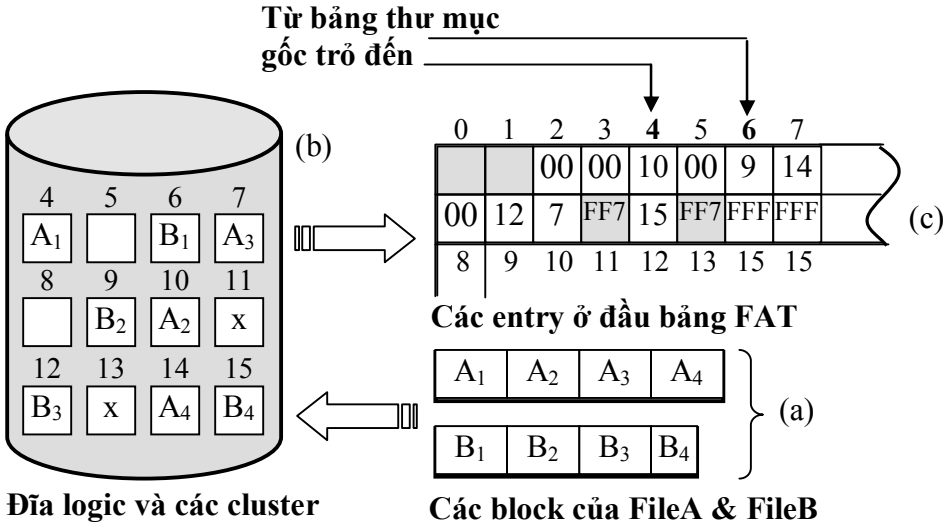
- *000h (0000h):* cluster tương ứng còn để trống.
- *FF7h (FFF7h):* cluster tương ứng bị bad. Trong quá trình định dạng đĩa hệ điều hành đánh dấu loại bỏ các cluster bị bad bằng cách ghi giá trị này voà phần tử tương ứng trong bảng FAT.

- *FF0h (FFF0h) - FF6h (FFF6h):* cluster tương ứng dành riêng cho hệ điều hành.
- *FF8h (FFF8h) - FFFh (FFFFh):* cluster tương ứng là cluster cuối cùng trong dãy các cluster chứa nội dung của một file.
- *002h (0002h) – FFEh (FFFEh):* đây là số hiệu của cluster trong bảng FAT, nó cho biết cluster tiếp theo trong dãy các cluster chứa nội dung của một file.

Trong bảng FAT, hai phần tử đầu tiên (00 và 01) không dùng cho việc theo dõi trạng thái cluster và ghi nhận bảng đồ cấp phát file, mà nó được sử dụng để chứa một giá trị nhận biết khuôn dạng đĩa, được gọi là byte định danh (byte ID) của đĩa, đây là byte đầu tiên của bảng FAT. Đối với đĩa cứng thì byte ID = F8h.

Như vậy để đọc được nội dung của một file trên đĩa thì trước hết hệ điều hành phải tìm được dãy các cluster chứa nội dung của một file. Nhưng bảng FAT chỉ cho biết số hiệu các cluster từ cluster thứ hai đến cluster cuối cùng trong dãy nói trên. Cluster đầu tiên trong dãy các cluster chứa nội dung của một file trên đĩa được tìm thấy trong bảng thư mục gốc.

Để thấy được cách mà hệ điều hành DOS dùng bảng FAT để quản lý việc lưu trữ các file trên đĩa cũng như theo dõi trạng thái các cluster trên vùng dữ liệu, ta xem hình minh hoạ sau đây.



Hình 4.10: Các file FileA và FileB (a) được lưu trên các cluster của đĩa logic (b) và sơ đồ định vị của nó trên bảng FAT (c).

Hình (a) ở trên cho thấy: có hai file, FileA và FileB, FileA có kích thước vừa đủ 4 cluster và được chia thành 4 block, FileB có kích thước nhỏ hơn 4

cluster cũng được chia thành 4 block, trong đó block B_4 mặc dù chưa đủ một cluster nhưng vẫn được chứa vào một cluster. Tức là, hệ điều hành cũng phải dùng đủ 8 cluster để lưu trữ nội dung của hai file FileA va FileB vào đĩa (hình b).

Đoạn FAT trong hình (c) ở trên cho biết các thông tin sau đây:

- Các cluster bị bad, không thể sử dụng: cluster 11 và cluster 13.
- Các cluster còn trống, chưa cấp phát: cluster 2, cluster 3, cluster 5, cluster 8.
- FileA được lưu tại các cluster: 4, 10, 7, **14** (chứa block cuối cùng)
- FileB được lưu tại các cluster: 6, 9, 12, **15** (chứa block cuối cùng)

Như vậy bảng thư mục gốc cho biết cluster đầu tiên chứa FileA là cluster 4, phần tử thứ 4 trong bảng FAT chứa giá trị 10, điều này chứng tỏ cluster 10 là cluster tiếp theo chứa nội dụng FileA, phần tử thứ 10 trong bảng FAT chứa giá trị 7, điều này chứng tỏ cluster 7 là cluster tiếp theo chứa nội dụng FileA, phần tử thứ 7 trong bảng FAT chứa giá trị FFFh, điều này chứng tỏ cluster 7 là cluster chứa block cuối cùng của FileA.

Các cluster chứa nội dung của một file có thể không liên tiếp nhau, nhưng nó thường nằm rải rác trong một phạm vi hẹp nào đó trên đĩa. Điều này giúp hệ điều hành đọc file được nhanh hơn nhờ tiết kiệm được thời gian duyệt và đọc qua các byte từ đầu đến cuối bảng FAT để dò tìm dãy các cluster chứa nội dung của file. Mặt khác, việc phân bố tập trung các cluster của một file rất phù hợp với các thuật toán đọc đĩa của hệ điều hành. Đối với các file dữ liệu, sau một thời gian kích thước của nó có thể tăng lên, hệ điều hành phải cấp phát thêm các cluster cho nó, các cluster mới này có thể nằm tại các vị trí tách xa các cluster trước đó, dẫn đến các cluster chứa nội dung của một file phân bố rải rác khắp bề mặt đĩa, điều này sẽ làm chậm tốc độ đọc file của hệ điều hành. Các file dữ liệu bị mở, thay đổi, ghi và đóng lại nhiều lần cũng có thể dẫn đến hiện tượng trên. Trên đĩa có thể xuất hiện hiện tượng có nhiều file bị phân bố rải rác khắc bề mặt đĩa, hiện tượng này được gọi là hiện tượng đĩa bị phân mảnh (fragmentary). Các đĩa bị phân mảnh sẽ làm cho tốc độ đọc file trên nó chậm đi rất nhiều. Trong trường hợp này người sử dụng phải thực hiện việc sắp xếp lại các cluster trên đĩa, để các cluster chứa nội dung của một file của tất cả các file trên đĩa được phân bố tập trung hơn, thao tác này được gọi là chống phân mảnh cho đĩa. Hệ điều hành DOS cung cấp nhiều công cụ để người sử dụng thực hiện việc chống phân mảnh cho đĩa cả ở mức ứng dụng và mức lập trình.

Để đọc nội dung của một file trên đĩa dựa vào bảng thư mục gốc và bảng FAT, hệ điều hành thực hiện theo các bước sau đây:

Tìm phần tử trong bảng thư mục gốc chứa thông tin của file cần đọc.

Tại phần tử này, xác định số hiệu của cluster đầu tiên trong dãy các cluster chứa nội dung của file **(giả sử cluster 4)**, giá trị này được xem như con trỏ trỏ tới bảng FAT để bắt đầu dò tìm các cluster từ thứ 2 đến cuối cùng trong dãy các cluster chứa nội dung của file cần đọc. Sau đó đọc block dữ liệu đầu tiên của file tại **cluster 4** trên vùng data của đĩa.

Xác định byte tương ứng với **phần tử 4** trong bảng FAT. Đọc giá trị dữ liệu tại **phần tử 4** này, giả sử **giá trị đọc được là 10**. Sau đó đọc block dữ liệu tiếp theo của file tại **cluster 10** trên vùng data của đĩa.

Xác định byte tương ứng với **phần tử 4** trong bảng FAT. Đọc giá trị dữ liệu tại **phần tử 4** này, giả sử **giá trị đọc được là 17**. Sau đó đọc block dữ liệu tiếp theo của file tại **cluster 17** trên vùng data của đĩa.

Xác định byte tương ứng với **phần tử 17** trong bảng FAT, sau đó thực hiện hoàn toàn tương tự như bước 4 cho đến khi đọc được giá trị **FFFh** (với FAT12) hoặc **FFFFh** (với FAT16) tại một phần tử nào đó **(giả sử phần tử 43)** trong bảng FAT thì đọc block dữ liệu cuối cùng của file tại **cluster 43** trên vùng data của đĩa, sau đó dừng lại. Tới đây kết thúc quá trình đọc file.

Chúng ta sẽ hiểu rõ hơn các bước 1 và 2 ở phần mô tả về bảng thư mục gốc trong mục này. Các bước trên chỉ đúng cho việc đọc các file mà thông tin của nó được lưu trữ trên ở các phần tử trong bảng thư mục gốc (được lưu trữ ở thư mục gốc) của đĩa.

Thao tác đọc file của DOS như trên là kém hiệu quả, vì ngoài việc đọc nội dung của file tại các cluster trên vùng data của đĩa hệ điều hành còn phải đọc và phân tích bảng FAT để dò tìm ra dãy các cluster chứa nội dung của một file. Hệ thống file NTFS trong windowsNT/2000 khắc phục điều này bằng cách lưu danh sách các cluster chứa nội dung của một file vào một vị trí cố định nào đó, nên khi đọc file hệ điều hành chỉ cần đọc nội dung của các cluster trên đĩa theo danh sách ở trên, mà không phải tốn thời gian cho việc dò tìm dãy các cluster chứa nội dung của file của hệ thống file FAT trong DOS.

Ngoài ra, nếu DOS có một cơ chế nào đó ghi lại được danh sách các cluster còn trống trên đĩa, thì tốc độ ghi file của hệ điều hành sẽ tăng lên vì hệ điều hành không tốn thời gian cho việc đọc bảng FAT để xác định cluster còn trống. Các hệ thống file của các hệ điều hành sau này như windows98, windowsNT/2000 đã thực hiện được điều này.

Độ rộng của một phần tử trong bảng FAT (12 bít hay 16 bit), quyết định dung lượng đĩa tối đa mà hệ điều hành có thể quản lý được. Nếu hệ điều hành sử dụng FAT12 thì mỗi phần tử trong FAT12 có thể chứa một giá trị

lên đến 2^{12}, đa số trong số này là số hiệu các cluster trên vùng data của đĩa, điều này có nghĩa là trên vùng data của đĩa có tối đa là 2^{12} cluster. Từ đây ta có thể tính được dung lượng đĩa tối đa (byte) mà hệ thống file FAT12 có thể quản lý được là: 2^{12} cluster * 4 sector/1 cluster * 512 byte/1 sector (a). Tương tự, dung lượng đĩa tối đa (byte) mà hệ thống file FAT16 có thể quản lý được là: 2^{16} cluster * 4 sector/1 cluster * 512 byte/1 sector (b). Rõ ràng với hệ thống file FAT12 thì DOS sẽ quản lý được một không gian đĩa lớn hơn so với FAT12 (theo a và b).

Windows98 sử dụng hệ thống file FAT32 và nó cho phép có tới 6 sector trên một cluster, nên nó có thể quản lý được một không gian đĩa lớn hơn nhiều lần (2^{32} cluster * 6 sector/1 cluster * 512 byte/1 sector) so với DOS. Nếu một file có kích thước là 50 sector, thì trong DOS file được chia thành 13 block (có 4 sector trong 1 block ≡ cluster) còn trong windows98 file được chia thành 9 block (có 6 sector trong 1 block ≡ cluster). Tức là, trong DOS file này được chứa ở 13 cluster trên đĩa (dãy các cluster chứa file gồm 13 phần tử) còn trong windows98 fie này được chứa trong 9 cluster (dãy các cluster chứa file gồm 9 phần tử). Điều này cho thấy file này sẽ được đọc nhanh hơn trong windows98, vì chỉ cần đọc 9 lần thay vì phải đọc 13 lần như trong DOS. Chưa kể, để đọc file các hệ điều hành phải phân tích bảng FAT để dò ra dãy các cluster chứa nội dung của file. Như vậy, hệ thống file của windows98 quản lý được một không gian đĩa lớn hơn và có tốc độ đọc file nhanh hơn, so với hệ thống file của DOS.

Để ghi một file vào đĩa, thì trong nhiều thao tác phải thực hiện hệ điều hành phải thực hiện việc đọc nội dung của các phần tử trong bảng FAT để tìm phần tử chứa giá trị 0, để ghi một block file vào cluster tương ứng trên vùng data. Trong khi đọc giá trị của các phần tử trong bảng FAT hệ điều hành có thể đọc được giá trị FF7h hoặc FFF7h, dấu hiệu của bad cluster, trong trường hợp này hệ điều hành sẽ không ghi file vào cluster tương ứng với phần tử này, và hệ điều hành sẽ tìm đọc một phần tử khác. Như vậy các bad cluster trên đĩa sẽ làm chậm tốc độ ghi file của hệ điều hành. Đây là một trong các hạn chế của các hệ thống file FAT. Các hệ thống file khác, NTFS chẳng hạn, khắc phục điều này bằng cách tạo ra một danh sách riêng để theo dõi các cluster bị bad và khi tìm cluster trống để ghi file hệ điều hành sẽ không đọc các phần tử trong bảng FAT tương ứng với các cluster này.

Việc đọc nội dung của một phần tử trong FAT16 chỉ đơn giản là đọc nội dung của 2 byte (1 word), trong khi đó việc đọc nội dung của một phần tử trong FAT12 sẽ phức tạp hơn vì 1.5 byte không phải là kiểu dữ liệu chuẩn của ngôn ngữ máy và DOS. Do đó, DOS phải gộp 2 phần tử liên tiếp để có được 3 byte sau đó đọc hết 3 byte và phân tích để có được nội dung của 2

filesize này chứa giá trị 0. Kích thước của thư mục con chỉ được biết nhờ lần theo chuỗi số hiệu cluster tương ứng trong bảng FAT.

Khi người sử dụng xóa một file trên đĩa hệ điều hành không xóa nội dung của file tại các cluster trên vùng data, không xóa dãy các cluster chứa file trong bảng FAT, thậm chí không xóa cluster đầu tiên trong dãy các cluster chứa file tại phần tử tương ứng với file trong bảng thư mục gốc mà hệ điều hành chỉ thay kí tự đầu tiên của tên file tại phần tử trong bảng thư mục gốc bằng giá trị E5h. Do đó, sau khi đã xóa một file thì hệ điều hành có thể khôi phục lại được file này, bằng các thay kí tự mã E5h ở byte đầu tiên của tên file bằng một kí tự khác. Trường hợp không khôi phục được là do sau một thời gian, hệ điều hành đã sử dụng phần tử trong thư mục gốc, các phần tử trong bảng FAT và các cluster trên vùng data của file đã bị xóa, cấp phát cho các file mới sau này. Khi duyệt bảng thư mục gốc gặp các phần tử có byte đầu bàng E5h hệ điều hành biết đây là phần tử của file đã bị xóa nên không in ra màn hình. Điều vừa trình bày trên đây hoàn toàn đúng với trường hợp của các thư mục con trên đĩa.

Để ghi một file vào thư mục gốc của đĩa (thông tin của file chứa ở một phần tử trong bảng thư mục gốc) hệ điều hành DOS thực hiện các bước sau đây:

1. Tìm một phần tử trong bảng thư mục gốc chưa sử dụng, đó là phần tử mà byte đầu tiên của nó chứa gia trị 00. **Giả sử tìm được phần tử thứ 105**.
2. Ghi tên file, phần mở rộng, thuộc tính của file, ngày giờ tạo file vào các trường tương ứng tại **phần tử 105** trong bảng thư mục gốc.
3. Tìm một entry trong bảng FAT chứa giá trị 000h, giả sử tìm được **entry 207**, điều này có nghĩa **cluster 207** trên vùng data còn trống.
4. Ghi số hiệu của entry này, **entry 207**, vào **trường start cluster** tại offset 1Ah của **phần tử 107** trong bảng thư mục gốc.
5. Ghi block đầu tiên của file vào **cluster 107** trên vùng data. Nếu nội dung của tập tin chứa vừa đủ trong 1 cluster, thì DOS sẽ thực hiện bước cuối cùng (bước 9), ngược lại DOS tiếp tục thực hiện bước 6.
6. Tiếp tục tìm một entry trong bảng FAT chứa giá trị 000h, giả sử tìm được **entry 215**, điều này có nghĩa **cluster 215** trên vùng data còn trống.
7. Ghi **giá trị 215** vào **entry 207** trong bảng FAT và ghi block thứ hai của file vào **cluster 215** trên vùng data.
8. Lặp lại các bước 6 và 7 cho đến khi ghi hết các block của file vào các cluster trên vùng data. Giả sử block cuối cùng của file được ghi vào **cluster 302** trên vùng data, tức là entry cuối cùng được tìm thấy (chứa giá trị 00h) là **entry 302**.

9. Bước cuối cùng: ghi **giá trị FFFh** vào **entry 107** hoặc vào **entry 302**.
10. Tính kích thước của tập tin và ghi vào **trường filesize** của **phần tử 105** trong bảng thư mục gốc.

➢ Thư mục con (subdirectory): **Như đã biết, bảng thư mục gốc của DOS định vị tại một vị trí cố định trên đĩa logic, sau 2 bảng FAT, số lượng phần tử trong bảng thư mục gốc là cố định, không thể mở rộng được, và được DOS quy định trong quá trình định dạng đĩa. Đó là những hạn chế về cấu trúc và tổ chức bảng thư mục gốc của DOS. Cụ thể là người sử dụng không thể chứa quá nhiều file, thư mục trên thư mục gốc và bảng thư mục gốc dễ bị virus tấn công. Hệ điều hành DOS đưa ra khái niệm thư mục con để khắc phục một phần những hạn chế trên.**

Đối với người sử dụng, thư mục con là những thư mục nằm trên thư mục gốc của đĩa, trên đĩa chỉ có một thư mục gốc nhưng có thể có nhiều thư mục con, trong thư mục con có thể chứa nhiều (không giới hạn) file và thư mục con khác, gắn liền với thư mục con là thư mục cha của nó, thư mục cha có thể là thư mục gốc hoặc một thư mục con khác. Nhưng đối với DOS, thư mục con là một file đặc biệt, file này có thuộc tính thư mục con, byte thuộc tính có giá trị 00010000 (16), và có trường filesize = 0.

Về mặt hệ thống, thư mục con có các điểm khác sau đây so với thư mục gốc:

- Hệ điều hành lưu trữ nó giống như lưu trữ các file khác trên đĩa. Tức là, muốn đọc được thư mục con hệ điều hành phải lần theo dấu vết của nó trong bảng FAT.
- Bảng thư mục của nó có số lượng phần tử không giới hạn, có thể tăng lên hay giảm xuống tùy thuộc vào số lượng file và thư mục chứa trong nó. Nhờ vậy mà người sử dụng có thể chứa nhiều file thư mục trong một thư mục con trên đĩa. Số lượng phần tử trong bảng thư mục của các thư mục con chỉ bị giới hạn bởi dung lượng đĩa và kích thước các file thư mục chứa trong nó.
- Bảng thư mục của nó có thể định vị tại một vị trí bất kỳ trên vùng data của đĩa. Có thể xem đây là một hạn chế của thư mục con, vì nếu tạo quá nhiều thư mục con trên đĩa thì bảng thư mục của các thư muc con sẽ chiếm hết nhiều không gian đĩa trên vùng data. Do đó hệ điều hành không khuyến khích tạo quá nhiều thư mục con trên các đĩa mềm. Virus khó có thể tấn công bảng thư mục của thư mục con vì nó không cố định.
- Muốn biết được kích thước của thư mục con hệ điều hành phải tính toán từ kích thước của tất cả các file trong thư mục con, hệ điều hành dựa vào bảng thư mục của thư mục con và bảng FAT để thực hiện việc tính toán này.

Cấu trúc của một phần tử trong bảng thư mục của thư mục con tương tự cấu trúc của một phần tử trong bảng thư mục gốc. Đa số các phần tử trong bảng thư