TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN MÔN HỌC: NGUYÊN LÝ HỆ ĐIỀU HÀNH

ĐỀ TÀI: NGHIÊN CỨU TÌM HIỂU VỀ QUẢN LÝ BỘ NHỚ NGOÀI TRONG HĐH WINDOWS.

Giáo viên:	Ths.Nguyễn Bá Nghiễn
Nhóm:	02
Lớp:	20212IT6025004. Khóa: 15

Hà Nội, 2022

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN MÔN HỌC: NGUYÊN LÝ HỆ ĐIỀU HÀNH

ĐỀ TÀI: NGHIÊN CỨU TÌM HIỂU VỀ QUẢN LÝ BỘ NHỚ NGOÀI TRONG HĐH WINDOWS.

Giáo viên:	Ths.Nguyễn Bá Nghiễn
Nhóm:	02
Sinh viên thực hiện:	Nguyễn Hải Dương - 2020602562
	Trần Công Huân - 2020602941
	Trần Đăng Khoa - 2020603025
	Nguyễn Văn Khơi - 2020602139
	Nguyễn Hồng Phong - 2020603340
Lớp:	20212IT6025004. Khóa: 15

Hà Nội, 2022

Mục lục

LỜI NÓI ĐẦU

Tất cả các ứng dụng trên máy tính đều cần lưu trữ và đọc lại thông tin mà nó nhận vào và xử lý. Trong khi một tiến trình đang chạy, nó có thể lưu trữ một lượng giới hạn thông tin trong phạm vị không gian địa chỉ sở hữu của nó.

Tuy nhiên khả năng lưu trữ này bị giới hạn bởi kích thước không gian địa chỉ ảo của hệ thống. Đối với một vài ứng dụng thì không gian này là vừa đủ, nhưng với một số ứng dụng khác thì nó quá nhỏ. Mặt khác nếu lưu giữ thông tin trong không gian địa chỉ của tiến trình thì thông tin này sẽ bị mất khi tiến trình kết thúc. Vấn đề thứ ba là phải đáp ứng việc truy cập thông tin đồng thời giữa các tiến trình trong môi trường hệ điều hành đa nhiệm. Những vấn đề này chúng ta có thể dễ dàng tìm hiểu thông qua quản lý tiến trình và quản lý bộ nhớ trong của máy tính. Để giải quyết những vấn đề trên hệ điều hành buộc phải thiết kế một hệ thông lưu trữ thông tin sao cho: Thứ nhất là phải lưu trữ được một khối lượng lớn thông tin. Thứ hai là thông tin phải được bảo toàn khi mà tiến trình sử dụng nó kết thúc. Và cuối cùng là có thể có nhiều tiến trình truy xuất thông tin đồng thời.

Trên hệ điều hành Windows, giải pháp cho tất cả vấn đề trên là lưu trữ thông tin trên đĩa cứng và các thiết bị media khác trong các đơn vị dữ liệu được gọi là các file (tập tin). Các tiến trình có thể đọc thông tin của tập tin và rồi ghi mới thông tin vào tập tin đó nếu cần thiết. Thông tin được lưu trữ trong tập tin phải không bị tác động bởi việc tạo và kết thúc tiến trình.

Các tập tin được quản lý bởi hệ điều hành. Thành phần hệ điều hành tham gia trực tiếp vào quá trình quản lý các tập tin trên đĩa được gọi là hệ thống file. Hệ điều hành phải xây dựng cấu trúc và tổ chức hoạt động của hệ thống tập tin. Một trong những nhiệm vụ quan trọng của hệ thống tập tin là theo dõi việc lưu trữ tập tin trên đĩa, theo dõi và điều hành việc truy cập vào tập tin của các tiến trình, bảo vệ tập tin và nội dung của nó,… Cấu trúc, tổ chức, hoạt động và những nhiệm vụ của bộ nhớ ngoài trên Hệ điều hành Windows sẽ được trình bày trong các phần dưới đây.

CHƯƠNG 1: TỔNG QUAN VỀ BỘ NHỚ NGOÀI

Bộ nhớ trong (RAM) không thể dung để lưu giữ dữ liệu vì mọi dữ liệu sẽ bị mất đi khi ngưng cung cấp nguồn điện cho bộ nhớ trong. Thay vào đó người ta sử dụng bộ nhớ ngoài, mà chủ yếu là đĩa từ để thay thế.

Ổ đĩa cứng, hay còn gọi là ổ cứng (HDD – Hard Disk Drive) là thiết bị dùng để lưu trữ dữ liệu trên bề mặt các tấm đĩa hình tròn phủ vật liệu từ tính. Ổ đĩa cứng là loại bộ nhớ "không thay đổi" (non-volatie), có nghĩa là chúng không bị mất dữ liệu khi ngừng cung cấp nguồn điện cho chúng.

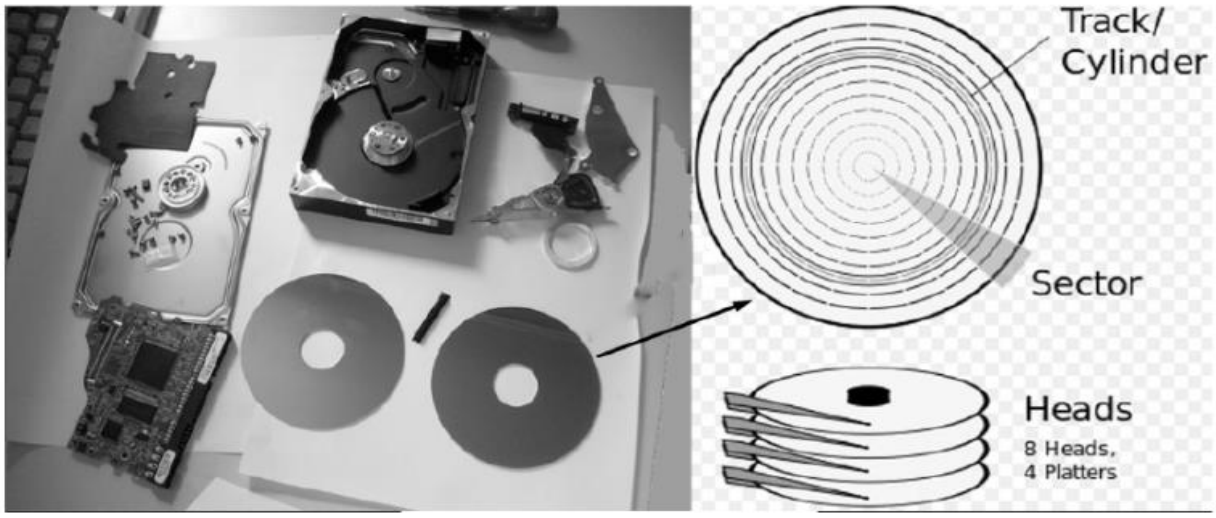
Ổ đĩa cứng là một thiết bị rất quan trọng trong hệ thống bới chúng chứa dữ liệu thành quả của một quá trình làm việc của những người sử dụng máy tính. Những sự hư hỏng của các thiết bị khác trong hệ thống máy tính có thể sửa chữa hoặc thay thế được, nhưng dữ liệu bị mất do yếu tố hư hỏng phần cứng của ổ đĩa cứng thường rất khó lấy lại được.

1.1. Cấu trúc vật lý

1.2. Đĩa từ (Platter)

Đĩa thường cấu tạo bằng nhôm hoặc thuỷ tinh, trên bề mặt được phủ một lớp vật liệu từ tính là nơi chứa dữ liệu. Tuỳ theo hãng sản xuất mà các đĩa này được sử dụng một hoặc cả hai mặt trên và dưới. Số lượng đĩa có thể nhiều hơn một, phụ thuộc vào dung lượng và công nghệ của mỗi hãng sản xuất khác nhau.

Mỗi đĩa từ có thể sử dụng hai mặt, đĩa cứng có thể có nhiều đĩa từ, chúng gắn song song, quay đồng trục, cùng tốc độ với nhau khi hoạt động.



Hình 1.1. Cấu tạo đĩa từ

1.2.1.Các rãnh từ (Track)

Trên một mặt làm việc của đĩa từ chia ra nhiều vòng tròn đồng tâm thành các track. Track có thể được hiểu đơn giản giống các rãnh ghi dữ liệu giống như các đĩa nhựa (ghi âm nhạc trước đây) nhưng sự cách biệt của các rãnh ghi này không có các gờ phân biệt và chúng là các vòng tròn đồng tâm chứ không nối tiếp nhau thành dạng xoắn trôn ốc như đĩa nhựa. Track trên ổ đĩa cứng không cố định từ khi sản xuất, chúng có thể thay đổi vị trí khi định dạng cấp thấp ổ đĩa (low format).

1.2.2.Sector

Trên track chia thành những phần nhỏ bằng các đoạn hướng tâm thành các sector. Các sector là phần nhỏ cuối cùng được chia ra để chứa dữ liệu.Theo chuẩn thông thường thì một sector chưa dung lượng 512 bytes.
Số sector trên các track là khác nhau từ phần rìa đĩa vào đến vùng tâm đĩa, các ổ đĩa cứng đều chia ra hơn 10 vùng mà trong mỗi vùng có số sector/track bằng nhau.

1.2.3.Cylinder

Tập hợp các track cùng bán kính (cùng số hiệu trên) ở các mặt đĩa khác nhau thành các cylinder. Nói một cách chính xác hơn thì: khi đầu đọc/ghi đầu tiên làm việc tại một track nào thì tập hợp toàn bộ các track trên các bề mặt đĩa còn lại mà các đầu đọc còn lại đang làm việc tại đó gọi là cylinder.

Trên một ổ đĩa cứng có nhiều cylinder bởi có nhiều track trên mỗi mặt đĩa từ.

1.2.4.Đầu đọc/ ghi (Read Write Heads)

Đầu đọc đơn giản được cấu tạo gồm lõi ferit (trước đây là lõi sắt) và cuộn dây (giống như nam châm điện). Gần đây các công nghệ mới hơn giúp cho ổ đĩa cứng hoạt động với mật độ xít chặt hơn như: chuyển các hạt từ sắp xếp theo phương vuông góc với bề mặt đĩa nên các đầu đọc được thiết kế nhỏ gọn và phát triển theo các ứng dụng công nghệ mới.

Đầu đọc trong đĩa cứng có công dụng đọc dữ liệu dưới dạng từ hoá trên bề mặt đĩa từ hoặc từ hoá lên các mặt đĩa khi ghi dữ liệu.

Số đầu đọc ghi luôn bằng số mặt hoạt động được của các đĩa cứng, có nghĩa chúng nhỏ hơn hoặc bằng hai lần số đĩa (nhỏ hơn trong trường hợp ví dụ hai đĩa nhưng chỉ sử dụng 3 mặt).

1.2.5.Cần di chuyển đầu đọc/ghi

Cần di chuyển đầu đọc/ghi là các thiết bị mà đầu đọc/ghi gắn vào nó. Cần có nhiệm vụ di chuyển theo phương song song với các đĩa từ ở một khoảng cách nhất định, dịch chuyển và định vị chính xác đầu đọc tại các vị trí từ mép đĩa đến vùng phía trong của đĩa (phía trục quay).

Các cần di chuyển đầu đọc được di chuyển đồng thời với nhau do chúng được gắn chung trên một trục quay (đồng trục), có nghĩa rằng khi việc đọc/ghi dữ liệu trên bề mặt (trên và dưới nếu là loại hai mặt) ở một vị trí nào thì chúng cũng hoạt động cùng vị trí tương ứng ở các bề mặt đĩa còn lại

CHƯƠNG 2: QUẢN LÝ BỘ NHỚ NGOÀI TRÊN WINDOWS

Vì sao phải quản lý bộ nhớ ngoài ?

■ Khi cần lưu trữ các chương trình hoặc dữ liệu, các hệ thống máy tính cần sử dụng bộ nhớ ngoài(đĩa từ , băng từ…) .

■ Nhiêm vụ chính của hệ điều hành phải đảm bảo được các chức năng sau :

- ✓ Quản lý không gian nhớ tự do trên bộ nhớ ngoài (*Free space mange*).
- ✓ Cấp phát không gian nhớ tự do(Allocation methods).
- ✓ Cung cấp các khả năng định vị bộ nhớ ngoài.
- ✓ Lập lịch cho bộ nhớ ngoài (Disk scheduling).

2.1. Các dạng lưu trữ dữ liệu trên hệ điều hành Windows.

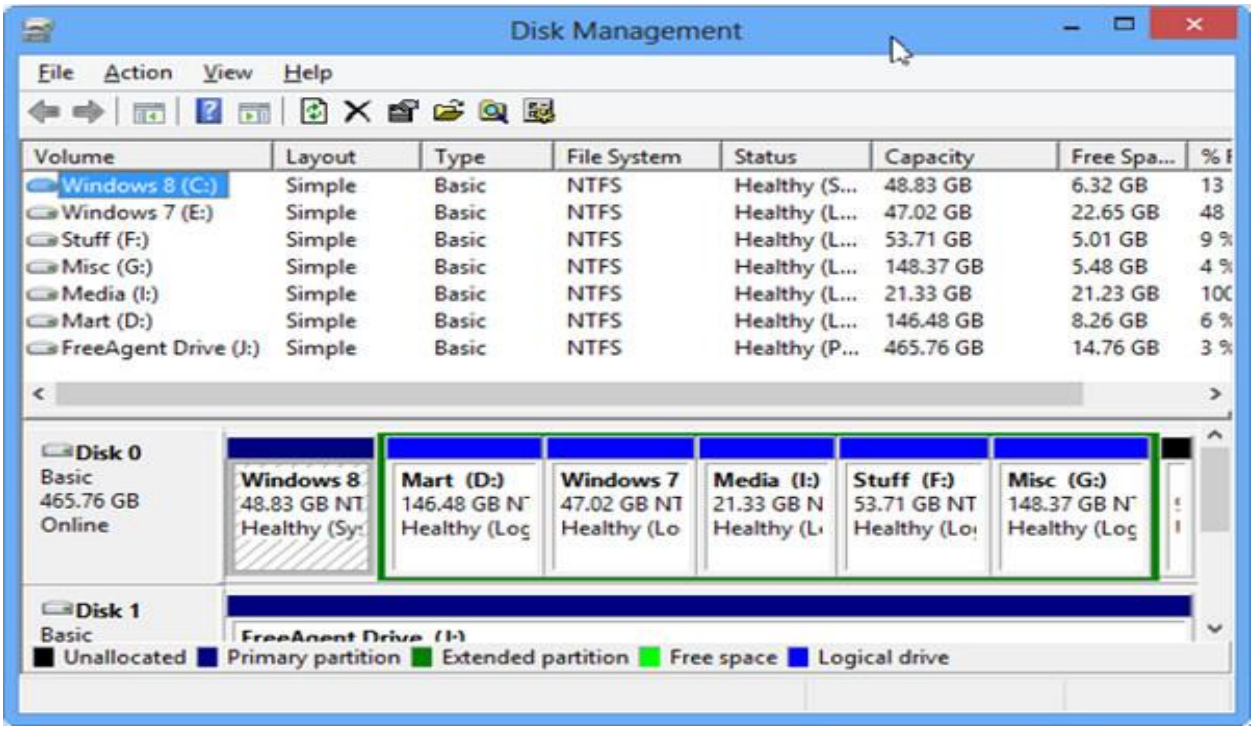
Hệ điều hành Windows hỗ trợ hai loại đĩa lưu trữ chủ yếu là ***Basic*** (cơ bản) và ***Dynamic*** (động).

2.1.1. Lưu trữ cơ bản (Basic Storage).

Gồm có các phân vùng cơ bản (*Partition Primary*), hay còn gọi là phân vùng chính, và phân vùng mở rộng (*Extended Partition*). Phân vùng tạo ra đầu tiên trên đĩa được gọi là phân vùng chính và toàn bộ không gian cấp cho phân vùng sẽ được sử dụng trọn vẹn. Mỗi ổ đĩa vật lý có thể tạo tối đa bốn phân vùng chính hoặc ba phân vùng chính và một phân vùng mở rộng. Với phân vùng mở rộng, ta có thể tạo ra tùy ý số phân vùng logic khác.

Trên ổ cứng có 1 vùng nhỏ dùng để ghi bảng phân vùng ổ đĩa (*Disk Partition Table*). Đây là nơi hệ điều hành sẽ đọc để theo dõi cách thức phân chia đang tồn tại trên ổ đĩa. Bảng phân vùng ổ đĩa có độ lớn 64 byte chia làm 4 mục, các thông tin về mỗi phân vùng chính được ghi trên một mục chiếm 16 Byte, ổ cứng vật lý chỉ có thể chia làm 4 phân vùng cũng là vì lý do đó. Tại một thời điểm chỉ có một phân vùng được nhận quyền khởi động, đó là phân vùng chứa hệ điều hành dùng để khởi động máy.

Để vượt qua giới hạn chỉ chia được 4 phân vùng cơ bản trên một ổ vật lý, người ta chia ổ cứng vật lý thành ba phân vùng cơ bản (*Partition Primary*) và một phân vùng mở rộng (*Extended Partition*). Sau đó lại chia phân vùng mở rộng này thành nhiều ổ đĩa logic (*Logical Drive*) như mô tả ở hình vẽ dưới đây:



Hình 2.1. Quản lý đĩa cứng trên Windows

2.1.2 Lưu trữ động (Dynamic Storage).

Đĩa lưu trữ động được chia thành các phân vùng động. Phân vùng động không chứa phân vùng hoặc ổ đĩa logic, và chỉ có thể truy cập được trên hệ điều hành Windows Server 2003 và Windows 2000. Windows Server 2003/ Windows 2000 hỗ trợ năm loại phân vùng động là ***spanned, simple, striped, mirrored*** và ***RAID-5.***

Ưu điểm của công nghệ lưu trữ động so với công nghệ lưu trữ căn bản là:

- Cho phép ghép nhiều ổ đĩa vật lý để tạo thành các ổ đĩa logic (Volume).
- Cho phép ghép nhiều vùng trống không liên tục trên nhiều đĩa cứng vật lý để tạo ổ đĩa logic.

■ Có thể tạo ra các ổ đĩa logic có khả năng dung lỗi cao và tăng tốc độ truy xuất…

2.1.2.1. Spanned Volume

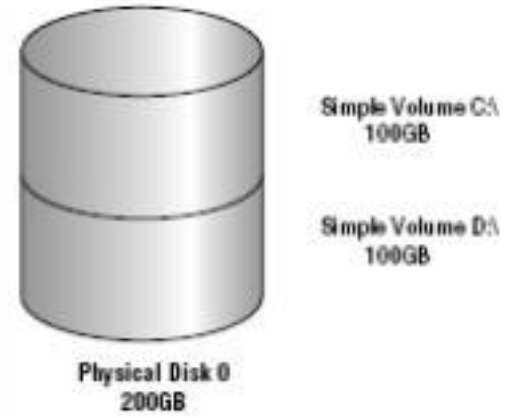
Bao gồm một hoặc nhiều đĩa lưu trữ động (tối đa là 32 đĩa). Sử dụng khi người dùng muốn tăng kích cỡ của phân vùng. Dữ liệu được ghi lên phân vùng theo thứ tự và hết đĩa này đến đĩa khác. Thông thường người quản trị sử dụng phân vùng *spanned* khi ổ đĩa đang sử dụng trong phân vùng sắp bị đầy và muốn tăng kích thước của phân vùng bằng cách bổ sung thêm một đĩa khác.

5

Do dữ liệu được ghi tuần tự nên phân vùng loại này không tăng hiệu năng sử dụng. Nhược điểm chính của phân vùng spanned là nếu một đĩa bị hỏng thì toàn bộ dữ liệu trên phân vùng sẽ không thể truy xuất được.

2.1.2.2. Simple Volume

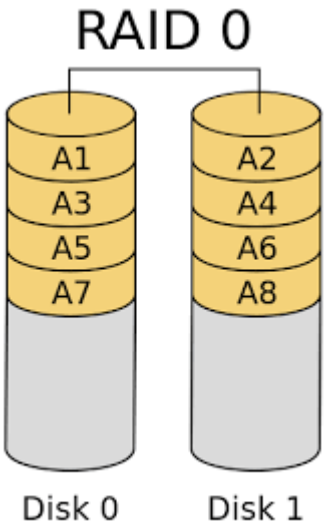
Phân vùng simple chứa không gian lấy từ một đĩa lưu trữ động duy nhất. Không gian đĩa này có thể liên tục hoặc không liên tục trên cùng một đĩa vật lý



Hình 2.2. Một đĩa vật lý được chia thành hai phân vùng đơn giản.

2.1.2.3. Striped Volume (RAID-0)

Lưu trữ dữ liệu lên các dãy (*strip*) bằng nhau trên một hoặc nhiều đĩa vật lý (tối đa là 32). Do dữ liệu được ghi tuần tự lên từng dãy nên người dùng có thể thi hành nhiều tác vụ I/O đồng thời, làm tăng tốc độ truy xuất dữ liệu. Thông thường, người quản trị mạng sử dụng phân vùng *striped* để kết hợp dung lượng của nhiều ổ đĩa vật lý thành một đĩa logic, đồng thời tăng tốc độ truy xuất.



Hình 2.3. Mô hình Striped Volume (Physic Disk: đĩa vật lý)

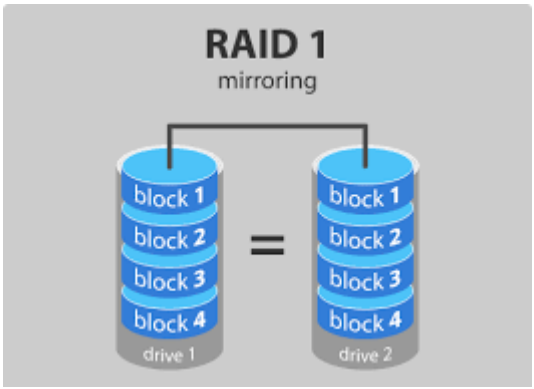
Nhược điểm chính của phân vùng striped là nếu một ổ đĩa bị hỏng thì dữ liệu trên toàn bộ phân vùng sẽ mất giá trị.

2.1.2.4. Mirror Volume (RAID-1)

Là hai bản sao của một phân đơn giản. Người dùng sử dụng dùng một ổ đĩa chính và một ổ đĩa phụ. Dữ liệu khi ghi lên đĩa chính đồng thời cũng sẽ được ghi lên đĩa phụ. Phân vùng dạng này cung cấp khả năng dung lỗi tốt. Nếu một đĩa bị

hỏng thì ổ đĩa kia vẫn làm việc bình thường và không làm gián đoạn quá trình truy xuất dữ liệu.

Nhược điểm của phương pháp này là bộ điều khiển đĩa phải ghi lần lượt lên hai đĩa, điều đó làm giảm hiệu năng hệ thống.



Hình 2.4. Mô hình phân vùng mirrored

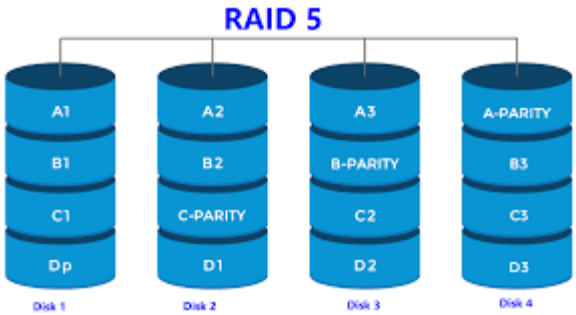
Để tăng tốc độ ghi đồng thời cũng tăng khả năng dung lỗi, người dùng có thể sử dụng một biến thể của phân vùng *mirrored* là duplexing. Theo cách này người dùng phải sử dụng một bộ điều khiển đĩa khác cho ổ đĩa thứ hai.

Nhược điểm chính của phương pháp này là chi phí cao. Để có một phân vùng 4GB cần phải tốn đến 8GB cho hai ổ đĩa.

2.1.2.5. RAID-5 Volume.

RAID là viết tắt của Redundant Arrays of Independent Disks (Các dãy đĩa dư thừa độc lập) thường chỉ được ứng dụng cho các máy chủ. Nguyên lý của RAID là đổi dung lượng lấy tốc độ hoặc sự an toàn dữ liệu. Để thiết lập được một hệ thống RAID ta cần phải có ít nhất 2 đĩa cứng trở lên.

Tương tự như phân vùng *striped* nhưng RAID-5 lại dùng thêm một dãy (*strip*) ghi thông tin kiểm lỗi parity. Nếu một đĩa của volume bị hỏng thì thông tin parity ghi trên đĩa khác sẽ giúp phục hồi lại dữ liệu trên đĩa hỏng. Phân vùng RAID-5 sử dụng ít nhất ba ổ đĩa và tối đa là 32 ổ đĩa.



Hình 2.5. Mô hình RAID-5. Các dãy đĩa độc lập

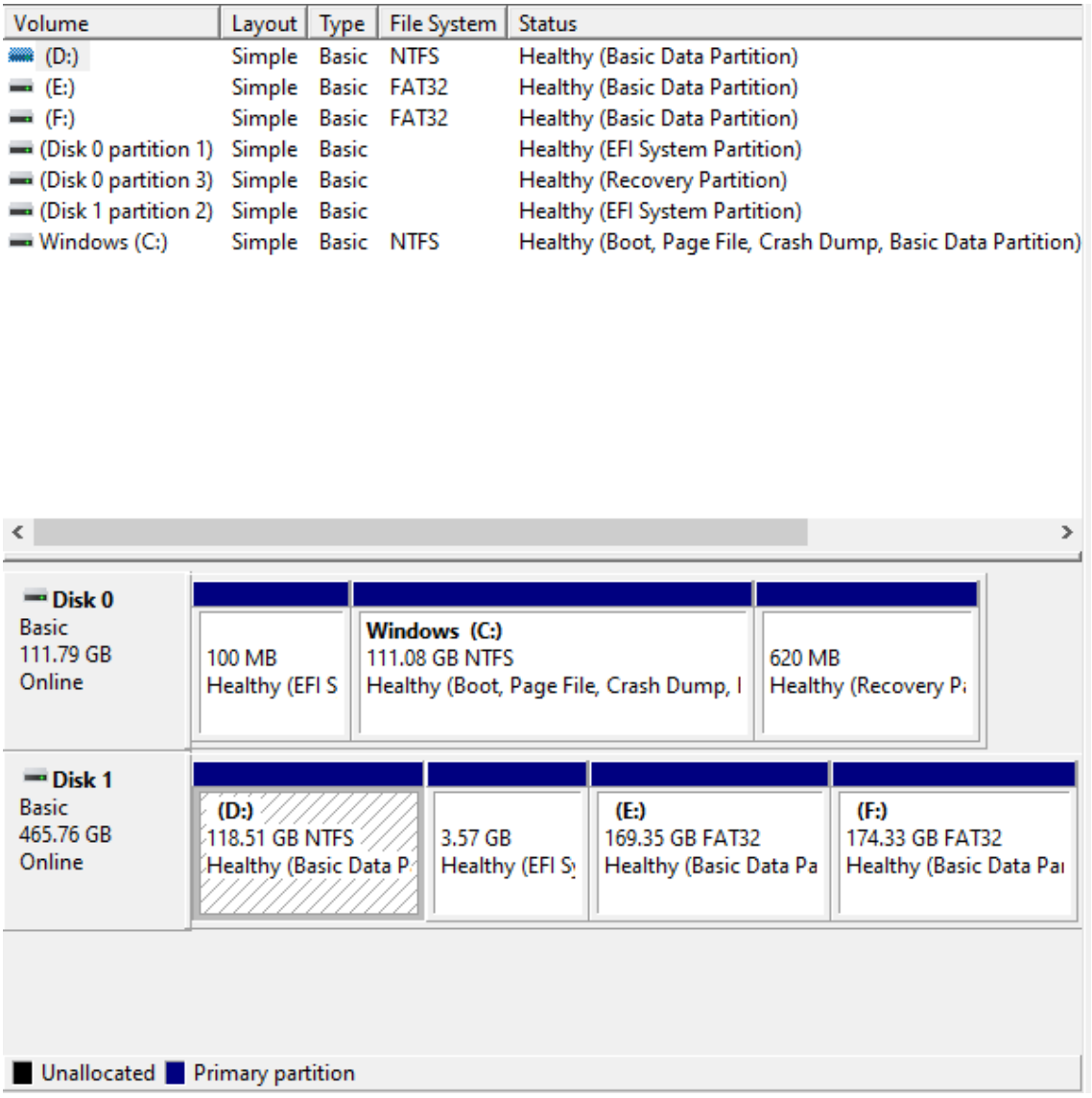
Ưu điểm chính của kỹ thuật này là khả năng dung lỗi cao và tốc độ truy xuất cao bởi sử dụng nhiều kênh vào/ra.

2.2. Chương trình quản lý bộ nhớ ngoài Disk Manager.

Disk Manager là một tiện ích giao diện đồ hoạ được Microsoft tích hợp sẵn phục vụ việc quản lý đĩa và phân vùng trong môi trường của hệ điều hành Windows. Để có thể sử dụng được hết các chức năng của chương trình, người dùng phải đăng nhập vào hệ thống bằng tài khoản Administrator. Để khởi động được chương trình Disk Manager trên Windows, ta có thể làm như sau:

- Đối với Windows 2000, Windows Server 2003 và một số phiên bản khác, vào menu Start, chọn Programs, chọn Administrative Tools, và chọn Computer Management.
- Đối với Windows XP, tại giao diện chính của hệ điều hành, ta chỉ cần kích chuột phải vào My Computer, chọn mục Manage.

Sau đó mở rộng mục Storage và chọn DiskManagement. Cửa sổ Disk Management sẽ xuất hiện như sau:



Hình 2.6. Giao diện chính của chương trình Disk Management

Chúng ta cùng tìm hiểu một vài tính năng thông dụng của công cụ này.

2.2.1. Xem thuộc tính của đĩa.

Nhấp phải chuột lên ổ đĩa vật lý muốn biết thông tin và chọn Properties. Hộp thoại Disk Properties xuất hiện với các thẻ (tab).

Thẻ Volumes cho ta biết các thông tin cơ bản của đĩa cứng vật lý:

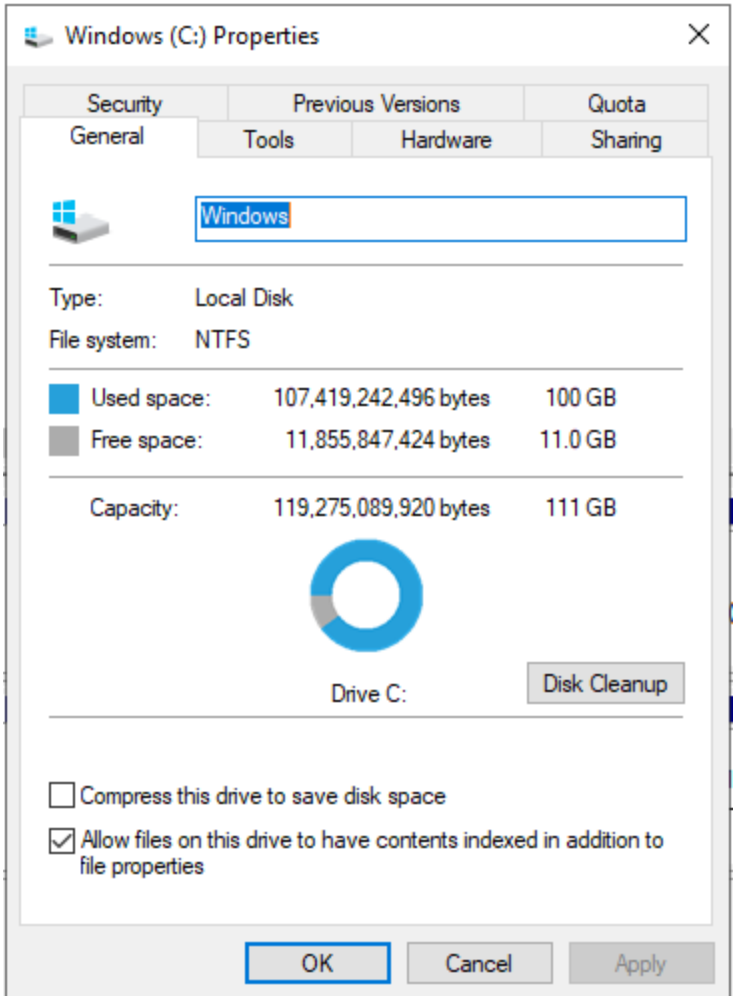
Disk: Số thứ tự của ổ đĩa vật lý

Type: Loại đĩa (basic, dynamic, CD-ROM, DVD, đĩa chuyển dời được, hoặc unknown)

Status: Trạng thái của đĩa (online hoặc offline)
Capacity: Dung lượng
Unllocated space: Lượng không gian chưa cấp phát
Bảng Volumes: Danh sách các phân vùng đã tạo trên đĩa

2.2.2. Xem thuộc tính của phân vùng hoặc đĩa cục bộ.

Trên một ổ đĩa lưu trữ động, người dùng sử dụng các phân vùng. Ngược lại trên một ổ đĩa lưu trữ căn bản, người dùng phải sử dụng các đĩa cục bộ (*local disk*). Phân vùng và đĩa cục bộ đều có chức năng như nhau, do vậy các phần sau dựa vào đĩa cục bộ để minh hoạ. Để xem thuộc tính của một đĩa cục bộ, nhấn chuột phải lên đĩa cục bộ đó và chọn Properties và hộp thoại Local Disk Properties xuất hiện như sau:



Hình 2.7. Hộp thoại Local Disk Properties

Hộp thoại cho ta biết khá chi tiết về thông tin của phân vùng:

■ General: Cung cấp các thông tin như nhãn đĩa, loại, hệ thống tập tin, dung lượng đã sử dụng, dung lượng còn trống và tổng dung lượng. Nút Disk Cleanup dùng để mở chương trình Disk Cleanup dùng để xoá các tập tin không cần thiết, giải phóng không gian đĩa.

■ Tools. Bấm nút Check Now để kích hoạt chương trình Check Disk dùng để kiểm tra lỗi như khi không thể truy xuất đĩa hoặc khởi động lại máy không đúng cách. Nút Backup Now sẽ mở chương trình BackupWizard, ở đây sẽ hướng dẫn các bước thực hiện việc sao lưu các tập tin và thư mục trên đĩa. Nút Defragment Now mở chương trình Disk Defragment dùng để dồn các tập tin trên đĩa thành một khối liên tục, giúp ích cho việc truy xuất đĩa.

■ Hardware. Liệt kê các ổ đĩa vật lý mà Windows nhận diện được. Bên dưới danh sách liệt kê các thuộc tính của ổ đĩa được chọn.

2.3. Quản lý không gian nhớ tự do trong hệ điều hành

2.3.1. Quản lý bộ nhớ bằng phương pháp liệt kê (free list)

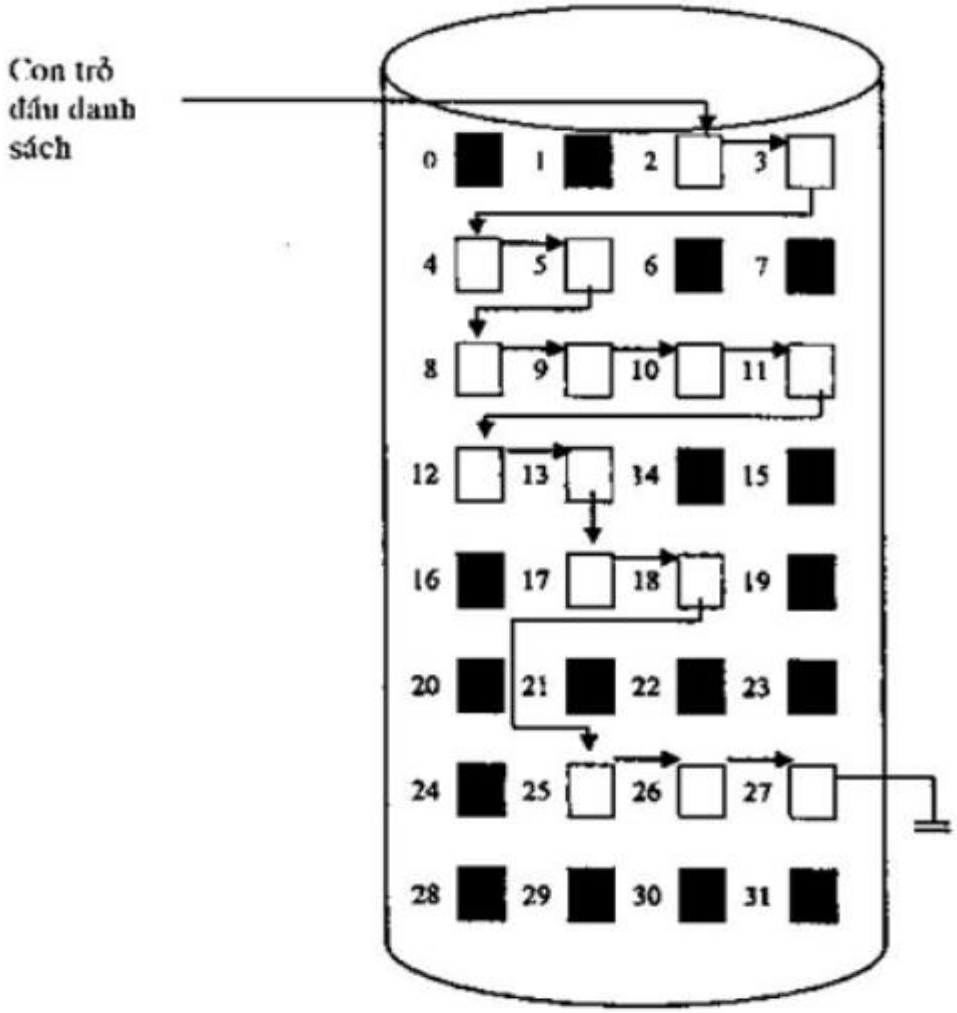
Trong phương pháp này, hệ thống sử dụng 1 danh sách móc nối để liệt kê các khối đĩa tự do. Con trỏ đầu danh sách chỉ tới khối đĩa tự do đầu tiên,mỗi khối có 1 con trỏ để trỏ tới khối kế tiếp.

Ưu điểm của phương pháp này là tiết kiệm không gian nhớ nhưng làm tang thời gian truy nhập dữ liệu.

2.3.2. Quản lý bộ nhớ bằng phương pháp lập nhóm(Grouping)

Trong phương pháp này, hệ thống cho phép nhóm các khối đĩa tự do liên tiếp thành 1 nhóm. Khối đĩa tự do đầu tiên trong nhóm lưu trữ địa chỉ của các khối đĩa tư do trong nhóm.

Khối đĩa cuối cùng trong nhóm lưu trữ địa chỉ của khối đĩa tự do đầu tiên của nhóm tiếp theo.



.

Hình 2.8. Mô tả không gian đĩa từ.

10

ta có bảng quản lý không gian nhớ tự do như sau:

Nhóm	Khối đầu	Khối cuối
I	2(2,3,4,5)	5(8)
II	8(8,9,10,11,12,13)	13(17)
III	17(17,18)	18(25)
IV	25(25,26,27)	27(…)

Hình 2.9. Bảng quản lý không gian nhớ tự do

2.3.3. Phương pháp đếm (Counting)

Phương pháp đếm là sự biến đổi của phương pháp lập nhóm. Trong phương pháp này, hệ thống lập danh sách quản lý địa chỉ của các khối đĩa tự do đầu tiên và số lượng các khối đĩa tự do liên tục kế tiếp các khối đĩa đó.
Ví dụ: theo hình 2.9 ta có danh sách quản lý không gian nhớ như sau:

Danh sách	Số lượng
2	4
8	6
17	2
25	3

2.4. Cấp phát không gian nhớ tự do trong hệ điều hành Windows

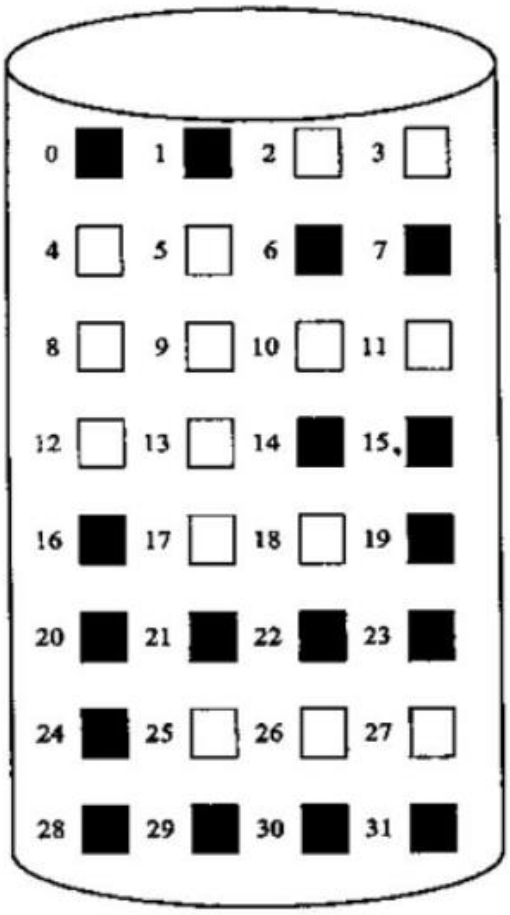
Có 3 phương pháp chính cấp phát không gian nhớ tự do: cấp phát liên tục (Contiguous), liên kết (Linked) và chỉ số (Index).

2.4.1. Cấp phát kề (Contiguous)

Để phân bố không gian nhớ cho một file, hệ thống chọn một đoạn liên tục các khối đĩa tự do để cấp phát cho file đó. Với phương pháp này, để định vị file hệ thống chỉ cần biết địa chỉ của khối đĩa tự do đầu tiên và số lượng block đã dùng.

Ưu điểm của cấp phát liên tục là hỗ trợ phương pháp truy nhập tuần tự và truy nhập trực tiếp, nhưng tồn tại 3 nhược điểm chính:

- Phải chọn được thuật toán tối ưu để tìm các vùng không gian tự do cấp phát cho file (First – Fit, Best – Fit hoặc Worst – Fit)
- Có thể xảy ra trường hợp không đủ số khối đĩa tự do liên tiếp cần thiết để cấp phát cho file (kích thước file lớn hơn vùng các khối đĩa liên tục lớn nhất).
- Trường hợp các khối đĩa tự do nằm tản mạn sẽ không được sử dụng, sẽ gây lãng phí không gian nhớ.



Directory

File	Start	Length
f1	0	2
f2	14	3
f3	19	6
f4	28	4
f5	6	2

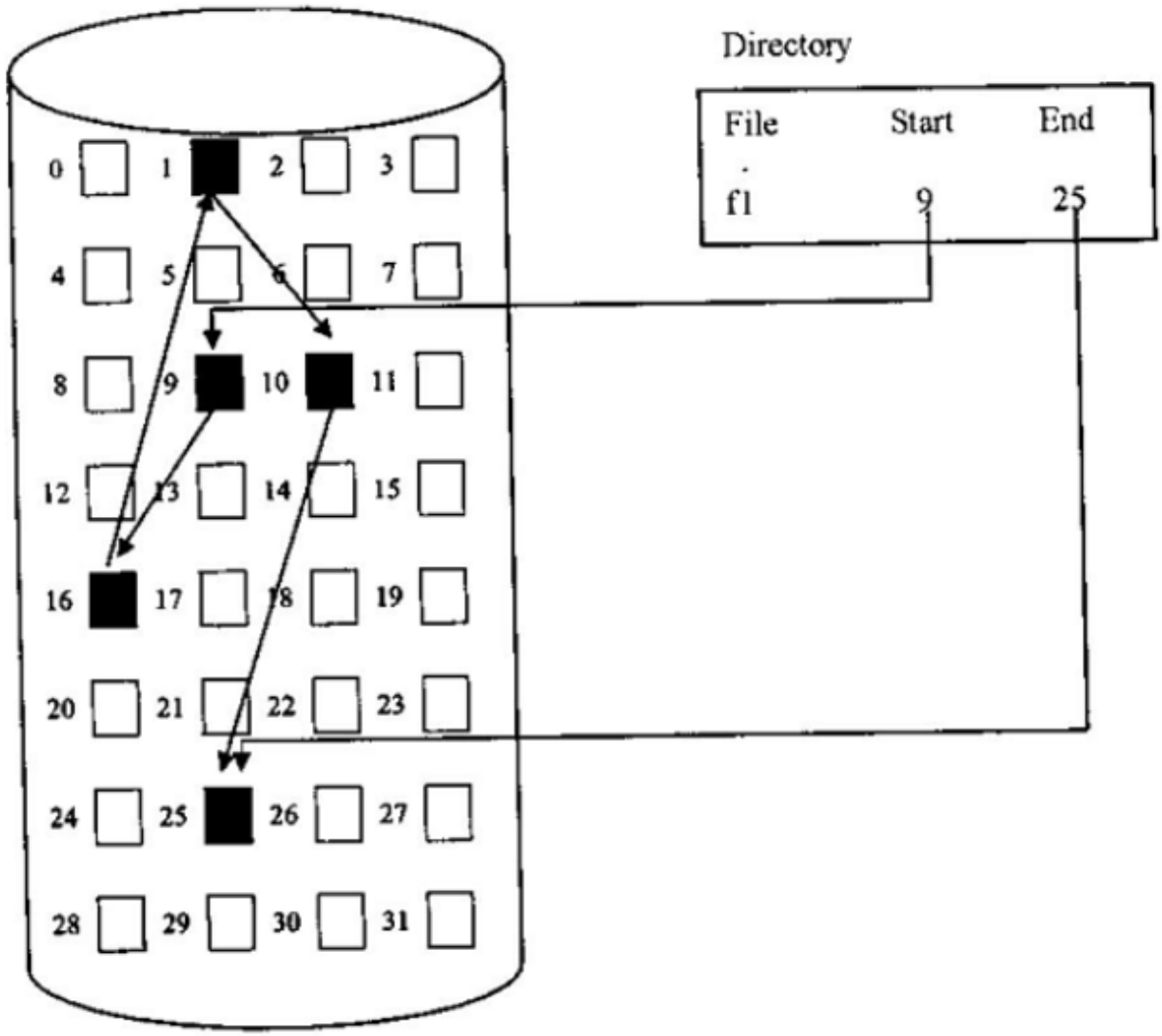
Hình 2.10. Không gian đĩa được cấp phát kề

2.4.2. Cấp phát liên kết (Linked)

Windows dùng phương pháp cấp phát liên kết để cấp phát không gian nhớ tự do.

Trong phương pháp này, mỗi file được định vị trong thư mục thiết bị bằng hai con trỏ, một cái trỏ tới khối đĩa đầu tiên, một cái trỏ tới khối đĩa cuối cùng để cấp phát cho file. Trong mỗi khối đĩa đã cấp phát cũng có một con trỏ để trỏ tới khối đĩa kế tiếp.

Ví dụ: File F1 được cấp phát 5 khối đĩa có số hiệu 9,16,1,11,25; khối đầu là 9, khối cuối là 25.

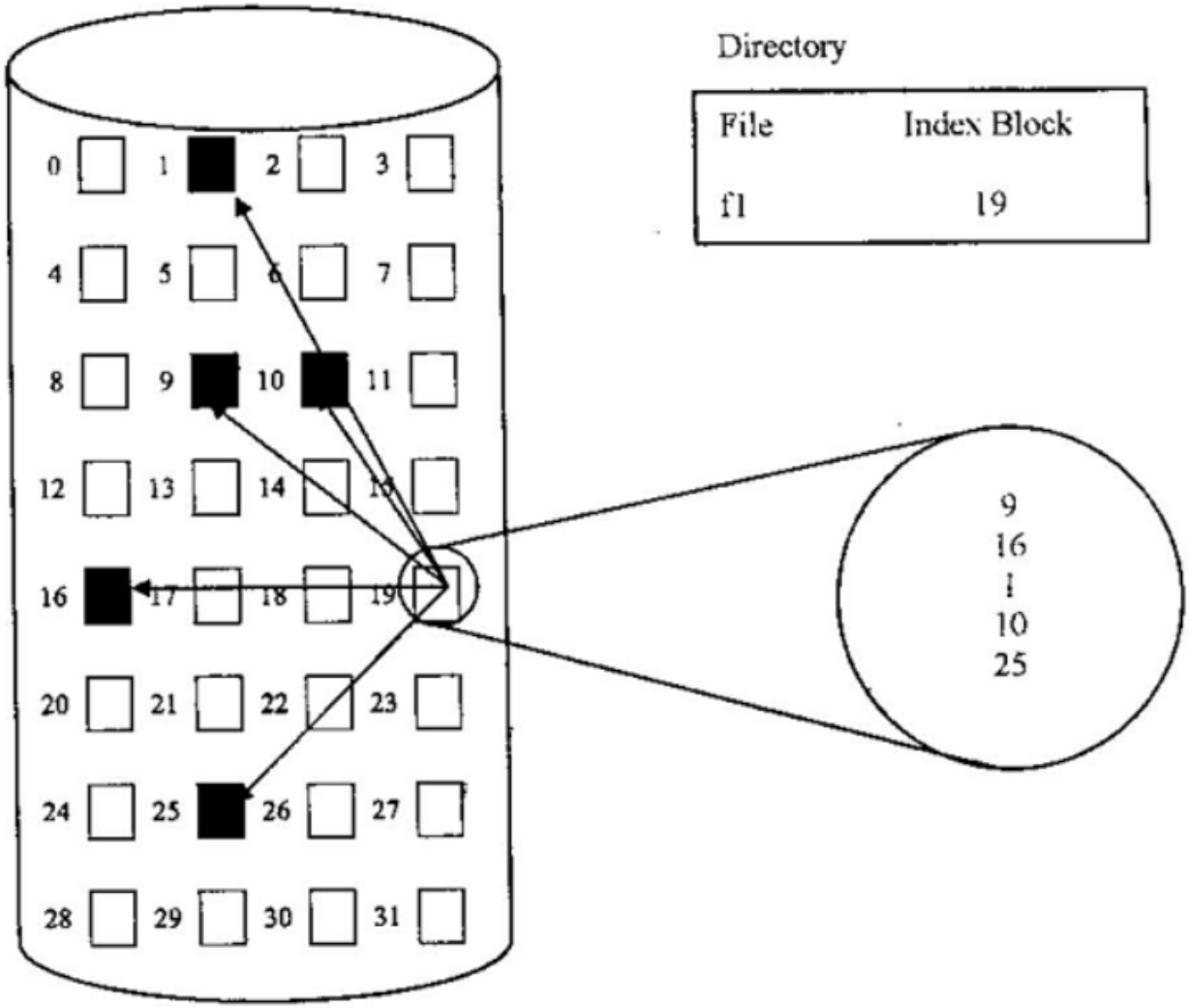


Hình 2.11. Không gian đĩa được cấp phát liên kết.

- Ưu điểm: Sử dụng được các khối đĩa tự do nằm tản mạn.
- Nhược điểm: Chỉ hỗ trợ truy nhập tuần tự không hỗ trợ truy nhập trực tiếp, độ tin cậy không đảm bảo nếu bị mất các con trỏ liên kết.
-Khi Windows cấp phát không gian nhớ theo phương pháp này, vì tận dụng các khối đĩa tự do nằm tảm mạn nên sẽ gây phân mảnh đĩa từ, phải dùng công cụ chống phân mảnh đĩa cứng. Điều này được thể hiện rõ trong hệ thống tập tin FAT của Windows.

2.4.3. Cấp phát theo chỉ số (Index)

Phương pháp này, để cấp phát không gian nhớ cho một file, hệ thống sử dụng một khối đĩa đặc biệt gọi là khối đĩa chỉ số (Index block) cho mỗi file. Trong khối đĩa chỉ số chứa địa chỉ các khối đĩa đã cấp phát cho file, trong thư mục thiết bị địa chỉ của các khối đĩa chỉ số. Khi một khối đĩa được cấp phát cho file thì hệ thống loại bỏ địa chỉ của khối đĩa này khỏi danh sách của cấc khối đĩa tự do và cập nhật vào khối chỉ số của file.



Hình 2.12. Cấp phát không gian đĩa theo chỉ số

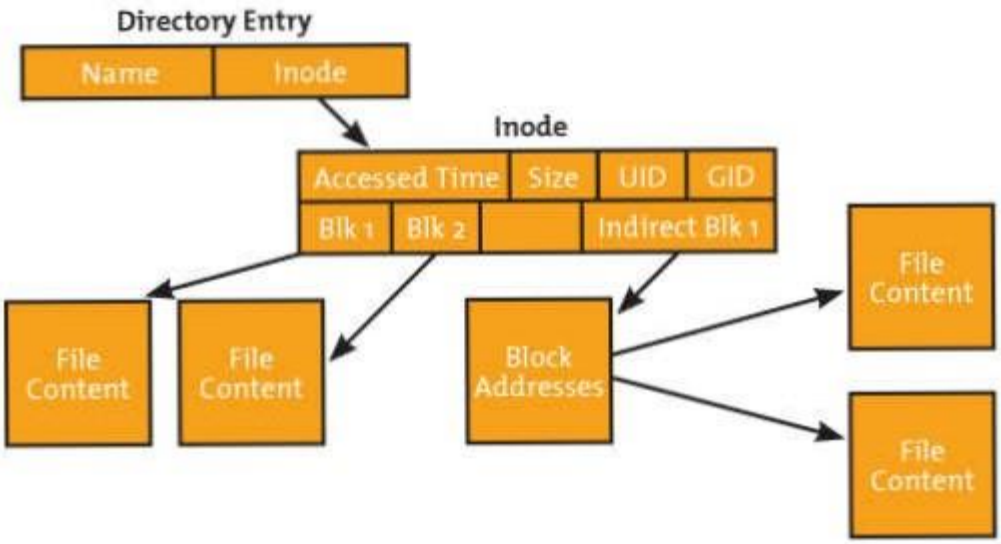
Phương pháp cấp phát theo chỉ số hỗ trợ truy nhập trực tiếp nhưng lãng phí không gian nhớ dành cho khối đĩa chỉ số.

Điểm này sinh ra câu hỏi: Khối chỉ số nên lớn bao nhiêu? Tuy nhiên, nếu khối chỉ số quá nhỏ nó không thể quản lý đủ các con trỏ cho một tập tin lớn, cần có một cơ chế giải quyết vấn đề này:

+ Cơ chế liên kết (linked scheme): một khối chỉ số thường là một đĩa. Do đó, nó có thể được đọc và viết trực tiếp bởi chính nó. Để cho phép đối với các tập tin lớn, chúng ta có thể liên kết nhiều khối chỉ số với nhau. Thí dụ: một khối chỉ số có thể chứa một header nhỏ cho tên tập tin và một tập hợp của các địa chỉ 100 khối đĩa đầu tiên. Địa chỉ tiếp theo (từ cuối cùng trong khối chỉ số) là nil (đối với một tập tin nhỏ) hay một con trỏ tới khối chỉ số khác (cho một tập tin lớn).

+ Chỉ số nhiều cấp (multilevel index): một biến dạng của biểu diễn liên kết là dùng khối chỉ số cấp 1 để chỉ tới khối chỉ số cấp 2. Khối chỉ số cấp 2 chỉ tới các khối tập tin. Để truy xuất một khối, hệ điều hành dùng chỉ số cấp 1 để tìm một khối chỉ số cấp 2 và khối đó tìm khối dữ liệu mong muốn. Tiếp cận này có thể được tiếp tục tới cấp 3 hay cấp 4, tùy thuộc vào kích thước tập tin lớn nhất được mong muốn. Với khối có kích thước 4,096 bytes, chúng ta có thể lưu 1,024 con trỏ 4 bytes trong một khối chỉ số. Chỉ số hai cấp cho phép 1,048,576 khối dữ liệu, cho phép tập tin có kích thước tới 4GB.
+ Cơ chế kết hợp (combined scheme): một biến dạng khác được dùng trong UFS là giữ 15 con trỏ đầu tiên của khối chỉ số trong inode của tập tin. 12 con trỏ đầu tiên của 15 con trỏ này chỉ tới khối trực tiếp (direct blocks), nghĩa là chúng chứa các địa chỉ của khối mà chứa dữ liệu của tập tin. Do đó, dữ liệu đối với các tập tin nhỏ (không lớn hơn 12 khối) không cần một khối chỉ số riêng. Nếu kích thước khối là 4KB, thì 48KB dữ liệu có thể truy xuất trực tiếp. 3 con trỏ tiếp theo chỉ tới các khối gián tiếp (indirect blocks). Con trỏ khối gián tiếp thứ nhất

là đại chỉ của khối gián tiếp đơn (single indirect blocks). Khối gián tiếp đơn là một khối chỉ số không chứa dữ liệu nhưng chứa địa chỉ của các khối dữ liệu. Sau đó, có con trỏ khối gián tiếp đôi (double indirect block) chứa địa chỉ của một khối mà khối này chứa địa chỉ của các khối chứa con trỏ chỉ tới khối dữ liệu thật sự. Con trỏ cuối cùng chứa địa chỉ của khối gián tiếp ba (triple indirect block). Với phương pháp này, số khối có thể cấp phát tới một tập tin vượt quá hạn lượng không gian có thể đánh địa chỉ bởi các con trỏ tập tin 4 bytes hay 4 GB. Nhiều cài đặt UNIX gồm Solaris và AIX của IBM hỗ trợ tới 64 bit con trỏ tập tin. Các con trỏ có kích thước này cho phép các tập tin và hệ thống tập tin có kích thước tới terabytes. Một inode được hiển thị trong hình 2.14:



Hình 2.13. Inode của UNIX

Cơ chế cấp phát lập chỉ số gặp một số vấn đề khó khăn về năng lực như cấp phát liên kết. Đặc biệt các khối chỉ số có thể được lưu trữ (cache) trong bộ nhớ,nhưng các khối dữ liệu có thể trải rông khắp phân khu.

2.5. Lập lịch cho đĩa từ trong hệ điều hành Window

2.5.1. Khái niệm về lập lịch cho đĩa

Thời gian truy nhập đĩa phụ thuộc vào ba yếu tố: thời gian di chuyển đầu từ đọc/ghi đến track hoặc cylinder cần thiết (seek-time), thời gian định vị đầu từ đọc/ghi tại khối đĩa cần truy nhập (latency-time) và thời gian truy nhập dữ liệu (transfer-time). Thời gian định vị đầu từ đọc/ghi và thời gian truy nhập dữ liệu thông thường cố định và phụ thuộc cấu trúc kỹ thuật của ổ đĩa. Do đó để tăng tốc độ truy nhập đĩa, các hệ điều hành thường quan tâm tới thời gian di chuyển đầu từ đọc/ghi.

Lập lịch cho đĩa là xây dựng các thuật toán dịch chuyển đầu từ đọc ghi sao cho thời gian truy nhập đĩa là tối ưu nhất

Thời gian truy nhập đĩa:

- Thời gian di chuyển đầu từ đọc ghi đến strack thích hợp(seek-time)

- Thời gian chờ cho khối cần thiết dưới đầu đọc(latency -time)
- Thời gian vận chuyển dữ liệu giữa đĩa và bộ nhớ chính(transfer-time)

Tất cả mọi công việc đều phụ thuộc vào việc nạp chương trình và nhập xuất tập tin, do đó điều quan trọng là dịch vụ đĩa phải càng nhanh càng tốt. Hệ điều hành có thể tổ chức dịch vụ truy xuất đĩa tốt hơn bằng cách lập lịch yêu cầu truy xuất đĩa.
Tốc độ đĩa bao gồm ba phần. Để truy xuất các khối trên đĩa, trước tiên phải di chuyển đầu đọc đến track hay cylinder thích hợp, thao tác này gọi là seek và thời gian để hoàn tất gọi là seek time. Một khi đã đến đúng track, còn phải chờ cho đến khi khối cần thiết đến dưới đầu đọc. Thời gian chờ này gọi là latency time. Cuối cùng là vận chuyển dữ liệu giữa đĩa và bộ nhớ chính gọi là transfer time. Tổng thời gian cho dịch vụ đĩa chính là tổng của ba khoảng thời gian trên.

Trong đó seek time và latency time là mất nhiều thời gian nhất, do đó để giảm thiểu thời gian truy xuất hệ điều hành đưa ra các thuật toán lập lịch truy xuất.

2.5.2. Nguyên lý làm việc của đĩa từ

2.5.2.1. Giao tiếp với máy tính

Toàn bộ cơ chế đọc/ghi dữ liệu chỉ được thực hiện khi máy tính (hoặc các thiết bị sử dụng ổ đĩa cứng) có yêu cầu truy xuất dữ liệu hoặc cần ghi dữ liệu

vào ổ đĩa cứng. Việc thực hiện giao tiếp với máy tính do bo mạch của ổ đĩa cứng đảm nhiệm.
Ta biết rằng máy tính làm việc khác nhau theo từng phiên làm việc, từng nhiệm vụ mà không theo một kịch bản nào, do đó quá trình đọc và ghi dữ liệu luôn luôn xảy ra, do đó các tập tin luôn bị thay đổi, xáo trộn vị trí. Từ đó dữ liệu trên bề mặt đĩa cứng không được chứa một cách liên tục mà chúng nằm rải rác khắp nơi trên bề mặt vật lý. Một mặt khác máy tính có thể xử lý đa nhiệm (thực hiện nhiều nhiệm vụ trong cùng một thời điểm) nên cần phải truy cập đến các tập tin khác nhau ở các thư mục khác nhau.

Như vậy cơ chế đọc và ghi dữ liệu ở ổ đĩa cứng không đơn thuần thực hiện từ theo tuần tự mà chúng có thể truy cập và ghi dữ liệu ngẫu nhiên tại bất kỳ điểm nào trên bề mặt đĩa từ, đó là đặc điểm khác biệt nổi bật của ổ đĩa cứng so với các hình thức lưu trữ truy cập tuần tự (như băng từ).
Thông qua giao tiếp với máy tính, khi giải quyết một tác vụ, CPU sẽ đòi hỏi dữ liệu (nó sẽ hỏi tuần tự các bộ nhớ khác trước khi đến đĩa cứng mà thứ tự thường là cache L1-> cache L2 ->RAM) và đĩa cứng cần truy cập đến các dữ liệu chứa trên nó. Không đơn thuần như vậy CPU có thể đòi hỏi nhiều hơn một tập tin dữ liệu tại một thời điểm, khi đó sẽ xảy ra các trường hợp:
Ổ đĩa cứng chỉ đáp ứng một yêu cầu truy cập dữ liệu trong một thời điểm, các yêu cầu được đáp ứng tuần tự.

2.5.2.2. Đọc và ghi dữ liệu trên bề mặt đĩa

Sự hoạt động của đĩa cứng cần thực hiện đồng thời hai chuyển động: Chuyển động quay của các đĩa và chuyển động của các đầu đọc.

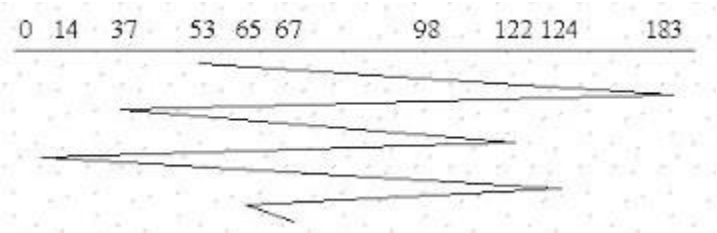
Sự quay của các đĩa từ được thực hiện nhờ các động cơ gắn cùng trục (với tốc độ rất lớn: từ 3600 rpm cho đến 15.000 rpm) chúng thường được quay ổn

định tại một tốc độ nhất định theo mỗi loại ổ đĩa cứng.
Khi đĩa cứng quay đều, cần di chuyển đầu đọc sẽ di chuyển đến các vị trí trên các bề mặt chứa phủ vật liệu từ theo phương bán kính của đĩa. Chuyển động này kết hợp với chuyển động quay của đĩa có thể làm đầu đọc/ghi tới bất kỳ vị trí nào trên bề mặt đĩa.
Tại các vị trí cần đọc ghi, đầu đọc/ghi có các bộ cảm biến với điện trường để đọc dữ liệu (và tương ứng: phát ra một điện trường để xoay hướng các hạt từ khi ghi dữ liệu).
Dữ liệu được ghi/đọc đồng thời trên mọi đĩa. Việc thực hiện phân bổ dữ liệu trên các đĩa được thực hiện nhờ các mạch điều khiển trên bo mạch của ổ đĩa cứng.

2.5.3. Các thuật toán lập lịch cho đĩa

2.5.3.1. First Come First Served (FCFS)

Nguyên lý: Để truy nhập tới 1 file, hệ thống sẽ tổ chức một hàng đợi các yêu cầu phục vụ của các track .Track nào có yêu cầu phục vụ trước thì đầu đọc/ghi sẽ dịch chuyển tới đó.
Ví dụ: Giả sử đĩa cứng có 200 track được đánh dấu từ 0 đến 199 và file F1 được phân bổ lần lượt các track theo thứ tự : 98, 183, 37, 122, 14, 124, 65, 67. Đầu đọc
đang ở vị trí 53.



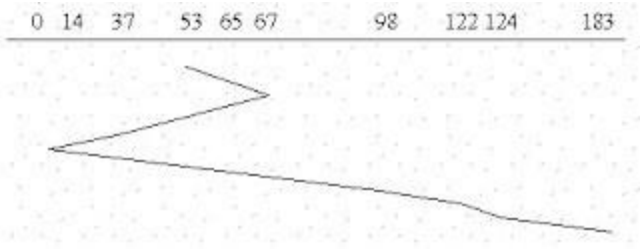
Hình 2.14. Phương pháp FCFS

-Ưu điểm:
Dễ lập trình
Các track cần truy xuất là liên tục
-Nhược điểm:
Số track mà đầu đọc phải di chuyển là nhiều
Hiệu quả của thuật toán phụ thuộc vào thứ tự của các track trong hàng đợi

2.5.3.2. Shortest Remaining Time First (SSTF):

Nguyên lý: Phương pháp này dựa trên quy tắc track nào có thời gian dịch chuyển đầu từ ngắn nhất thì phục vụ trước.

Theo ví dụ trên, sơ đồ dịch chuyển đầu từ đọc/ghi theo thuật toán SSTF được thể hiện như sau:

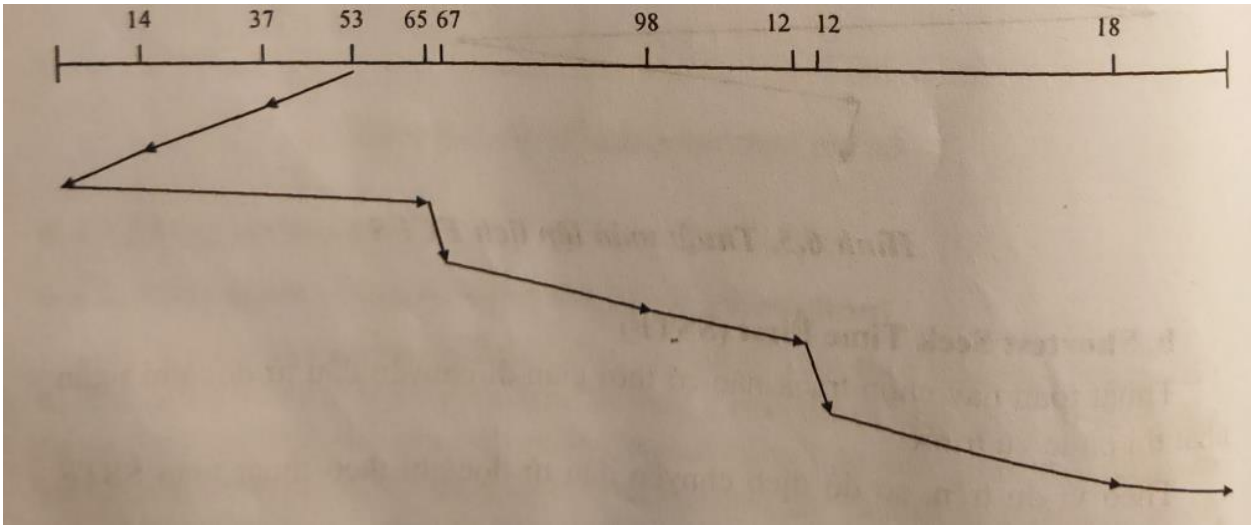


Hình 2.15. Phương pháp SSTF

2.5.3.3. Scan

Trong thuât toán này, đầu từ đọc/ghi sẽ quét về một phía của ô đĩa sau đó quét ngược lại, trong quá trinh quét gặp track não có nhu cầu thì sẽ phục vu track đó.

Theo ví dụ trên, sơ đồ dịch chuyển đầu từ đọc/ghi theo thuât toán Scan được thể hiện nhu sau:

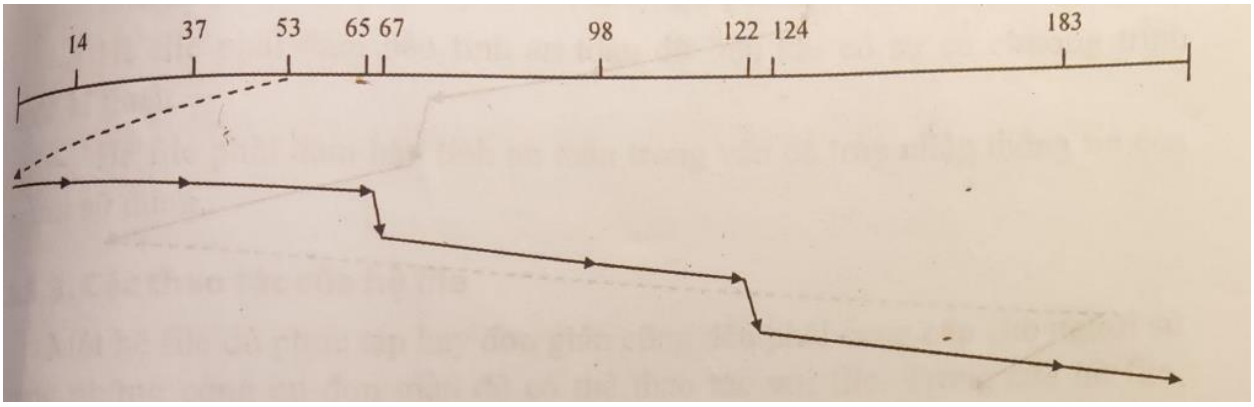


Hình 2.16. Phương pháp Scan

2.5.3.4. C-scan

Thuật toán này tương tự như Scan nhưng đầu từ đọc/ghi không phục vụ đường về (Không quét ngược lại).

Ví dụ trên, sơ đồ dịch chuyển đầu từ đọc/ghi theo thuật toán CC-scan được thể hiện như sau:

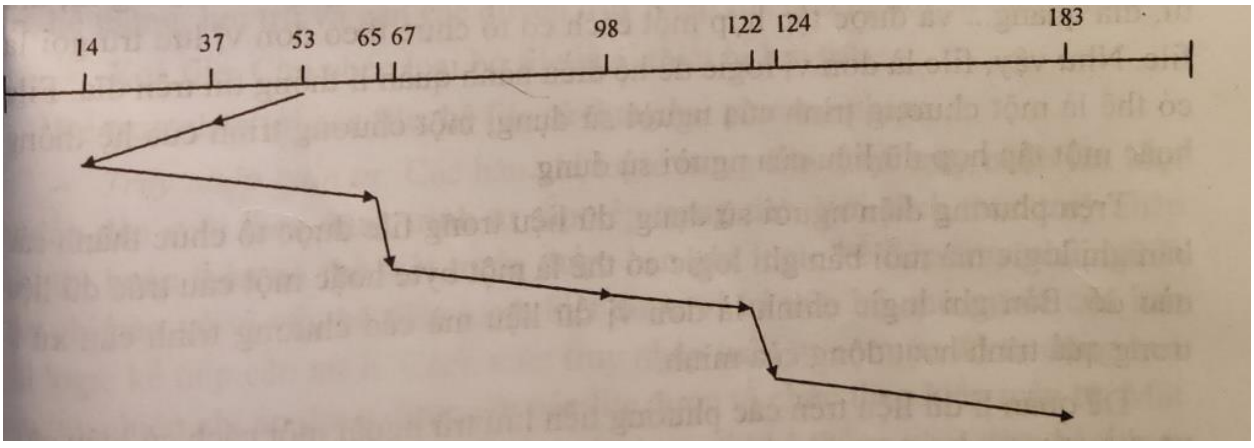


Hình 2.17. Phương pháp C-scan

2.5.3.5. Look

Tương tự như Scan nhưng trong thuật toán này, đầu từ đọc/ghi chỉ quét trong phạm vi các track có nhu cầu phục vụ, không quét tới track đầu tiên hoặc cuối cùng(nếu các track này không có yêu cầu phục vụ).

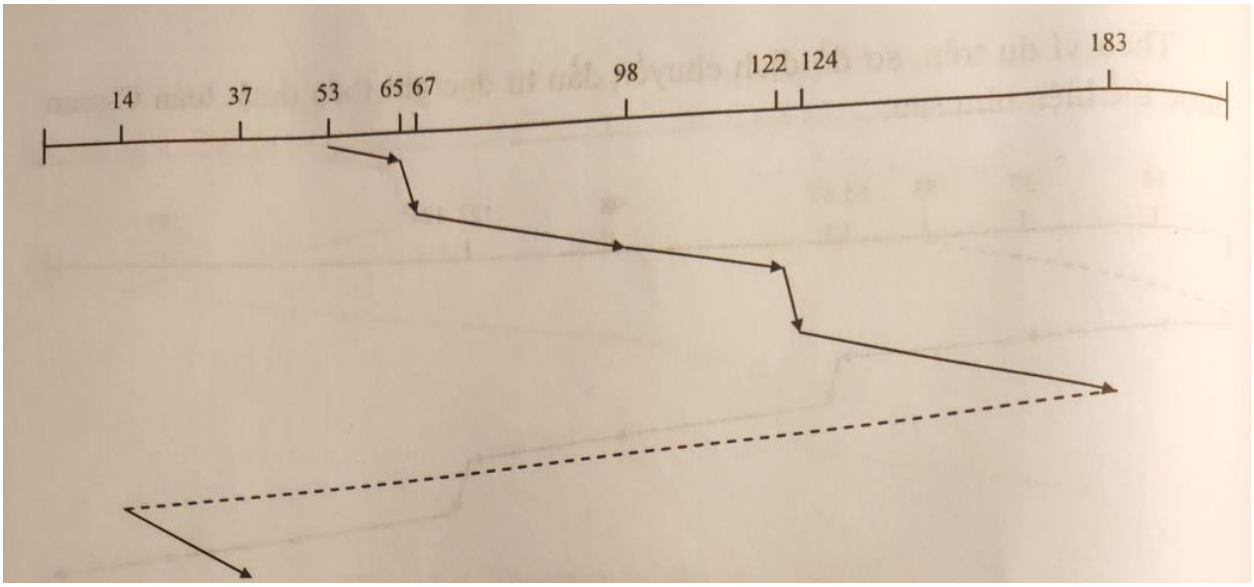
Theo ví dụ trên, sơ đồ dịch chuyển đầu từ đọc/ghi theo thuật toán Look được thể hiện như sau:



Hình 2.18. Phương pháp Look

2.5.3.6. C-look

Tương tự như look nhưng đầu từ đọc/ghi không phục vụ đường về.Theo ví dụ trên,sơ đồ dịch chuyển đầu từ đọc/ghi theo thuật toán C-look được thể hiện như sau:



Hình 2.19. Phương pháp C-look

Chú ý : Thuật toán FCFS, SSTF được áp dụng phổ biến, các thuật toán kiểu Scan, Look chỉ được áp dụng cho những đĩa chịu tải lớn.