

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN MÔN HỌC
NGUYÊN LÝ HỆ ĐIỀU HÀNH

**Tên đề tài: Nghiên cứu tìm hiểu về quản lí Bộ nhớ ngoài
trong hệ điều hành Linux**

Giáo viên hướng dẫn: Nguyễn Bá Nghiễn

Lớp: IT6025.4

Nhóm thực hiện: Nhóm 11

Hà Nội-2022

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN MÔN HỌC
NGUYÊN LÝ HỆ ĐIỀU HÀNH

**Tên đề tài: Nghiên cứu tìm hiểu về quản lí Bộ nhớ ngoài
trong hệ điều hành Linux**

Giáo viên hướng dẫn: Nguyễn Bá Nghiễn

Lớp: IT6025.4

Nhóm thực hiện: Nhóm 11

Thành viên trong nhóm:

- 1. Phan Thị Lan Anh**
- 2. Hoàng Minh Huệ**
- 3. Nguyễn Linh Trường**
- 4. Mai Thị Ánh Tuyết**
- 5. Dương Thị Hồng Vân**

LỜI MỞ ĐẦU

Vài năm vừa qua. Linux đã thực sự tạo ra một cuộc cách mạng trong lĩnh vực máy tính. Sự phát triển và những gì chúng ta mang lại cho máy tính thật đáng kinh ngạc: một hệ điều hành đa nhiệm, đa người dùng. Linux có thể chạy trên nhiều bộ vi xử lý khác nhau như Intel, Motorola, MC68K, Dec Alpha. Nó tương tác tốt với các hệ điều hành: Apple, Microsoft và Novell. Không phải ngẫu nhiên mà ngành công nghệ thông tin Việt Nam chọn Linux làm hệ điều hành nền cho các chương trình ứng dụng chủ đạo về kinh tế và quốc phòng. Với mã nguồn mở, sử dụng Linux an toàn hơn các ứng dụng Windows. Linux đem lại cho chúng ta lợi ích về kinh tế với rất nhiều phần mềm miễn phí. Mã nguồn mở của hệ điều hành và các chương trình trên Linux là tài liệu vô giá để chúng ta học hỏi về kỹ thuật lập trình vốn là những tài liệu không được công bố đối với các ứng dụng Windows.

Trong bài tập lớn này, chúng ta sẽ tìm hiểu một phần quan trọng trong hệ điều hành Linux đó là: quản lý bộ nhớ ngoài trong Linux, Một hệ điều hành muốn chạy ổn định thì phải có một cơ chế quản lý bộ nhớ hiệu quả. Cơ chế này sẽ được trình bày một cách chi tiết trong bài tập.

MỤC LỤC

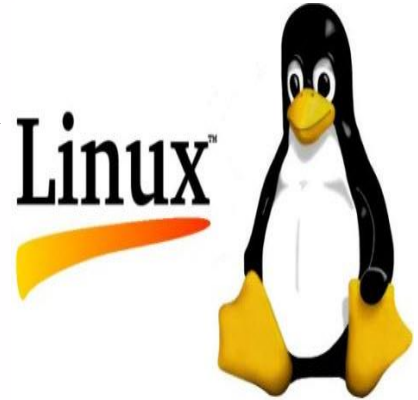
LỜI MỞ ĐẦU	1
MỤC LỤC.....	2
CHƯƠNG 1: MỞ ĐẦU	4
1.1.Hệ điều hành Linux.....	4
1.2.Tổng quan về quản lí bộ nhớ ngoài.....	6
CHƯƠNG 2: CẤU TRÚC VẬT LÍ-THƯ MỤC THIẾT BỊ	7
2.1.Cấu trúc vật lí.....	7
2.2.Thư mục thiết bị.....	7
CHƯƠNG 3: CÁC PHƯƠNG PHÁP QUẢN LÍ KHÔNG GIAN NHỚ TỰ DO.....	9
3.1.Phương pháp dùng bit vector (bitmap)	9
3.2.Phương pháp liệt kê Free List.....	9
3.3.Phương pháp lập nhóm (Grouping).....	10
3.4.Phương pháp đếm (Counting).....	11
CHƯƠNG 4. CÁC PHƯƠNG PHÁP CẤP PHÁT KHÔNG GIAN NHỚ TỰ DO	13
4.1.Cấp phát kề (contiguous).....	13
4.2.Cấp phát liên kết (Linked).....	15
4.3.Cấp phát theo chỉ số.....	16
CHƯƠNG 5: LẬP LỊCH CHO Đĩa.....	21
5.1. First Come First Served (FCFS).....	21
5.2.Shortest Seek Time Firtst (SSTF).....	21
5.3. Scan.....	22
5.4. C – Scan.....	23
5.5.Look.....	24

5.6. <i>C – Look</i>	24
CHƯƠNG 6: So sánh với nguyên lý quản lý bộ nhớ ngoài đã học.....	26
6.1. <i>Lịch sử phát triển</i>	26
6.2. <i>Xử lý, can thiệp vào mã nguồn</i>	26
6.3. <i>Ứng dụng – phần mềm</i>	27
6.4. <i>Đối tượng sử dụng</i>	27
6.5. <i>Bảo mật</i>	28
KẾT LUẬN	29
Tài Liệu Tham khảo	30

CHƯƠNG 1: MỞ ĐẦU

1.1. Hệ điều hành Linux

Linux là hệ điều hành. Về mặt nguyên tắc hệ điều hành cũng là một software; nhưng đây là một software đặc biệt – được dùng để quản lý, điều phối các tài nguyên (resource) của hệ thống (bao gồm cả hardware và các software khác). Linux còn được gọi là Open Source Unix (OSU), Unix-like Kernel, clone of the UNIX operating system.



Linux do Linus Torvalds, một sinh viên tại trường Đại Học ở Helsinki (Phần Lan) phát triển dựa trên hệ điều hành Minix, một hệ điều hành có cấu trúc tương tự Unix với các chức năng tối thiểu được dùng trong dạy học.

Hiện nay, Linux là một hệ điều hành với mã nguồn mở (Open Source) và miễn phí (free) dưới bản quyền của tổ chức GNU (Gnu's Not Unix).

Khởi đầu, Linux được thiết kế để hoạt động trên nền tảng của kiến trúc i386 Intel với khả năng đa tác vụ (multitasking). Tuy nhiên ngày nay, Linux đã có các phiên bản trên các họ chip khác chẳng hạn như chip Alpha.

Linux có nguyên lý hoạt động tương tự hệ điều hành Unix (Unix-like). Mặc dù Linux không phải là Unix nhưng người ta vẫn xem Linux như là phiên bản Unix trên PC (PC version of Unix OS).

Do là Unix-like, Linux có đầy đủ tất cả các đặc tính của Unix (fully functional). Ngoài ra nó còn hỗ trợ thêm một số tính năng mà trên Unix không có, như long file name (tên file có ký tự space “ ”).

Hiện tại có nhiều hãng, nhiều tổ chức, nhiều nhóm khác nhau cùng phát triển Linux. Tất cả các phiên bản (release) Linux đều có chung phần kernel (phần nhân của hệ điều hành) và hầu hết các tính năng đặc trưng, tuy nhiên các tool (công cụ) và utility (tiện ích) có đôi chút dị biệt.

Có rất nhiều các ứng dụng cho Linux, tuy nhiên hầu hết các ứng dụng cho Linux hiện có đều là các ứng dụng mang tính chuyên dụng. Để đưa Linux vào từng gia đình, các tổ chức, các hãng đang cố gắng phát triển các ứng dụng mang tính phổ cập trên Linux chẳng hạn hãng SUN đưa ra phiên bản Star Office tương tự như MS Office – và cũng tương thích với MS Office - cho những người sử dụng Linux ở gia đình, văn phòng.

Hãng Borland (nay là hãng Inprise) đang phát triển một dự án có tên là KyLix, nhằm đưa ra một môi trường lập trình cấp cao trên Linux, đồng thời các ứng dụng trên Windows được viết bằng Delphi/C++Builder sẽ dễ dàng compile (biên dịch) lại dưới Linux bằng KyLix. Hiện tại Kylix đã có phiên bản thử nghiệm (beta).

Dự án này hứa hẹn một loạt các ứng dụng thông thường đã có trên MS Windows sẽ mau chóng được chuyển sang Linux, và điều này sẽ giúp cho hệ điều hành Linux dễ dàng thâm nhập vào thị trường PC nhanh chóng hơn.

Các ứng dụng được viết trên Linux đều có thể hoạt động trên các hệ thống UNIX (có thể cần phải compile lại).

Các release hiện nay gồm có:

RedHat Linux (Fedora Core): Là phiên bản khá phổ biến. Cung cấp khá nhiều tool và utility để hỗ trợ user (người sử dụng) từ các thao tác setup (cài đặt) đến config (cấu hình) hệ thống.

Mandrake Linux: Một dòng khác thoát thai từ RedHat Linux, tương thích hoàn toàn với RedHat. Thường có nhiều phần mềm mới đang ở giai đoạn thử nghiệm.

Slackware Linux: Đây là một trong phiên bản Linux lâu đời. Hỗ trợ các dịch vụ mạng rất mạnh, tuy nhiên việc setup và config đòi hỏi user có kiến thức tốt về hệ điều hành này.

S.u.S.E. Linux: Do hãng S.u.S.E (Đức) phát hành, khá phổ biến tại Âu châu, nhưng không được phổ biến tại các nước khác. Có các công cụ riêng để hỗ trợ setup và config tương đối dễ sử dụng.

Free BSD Linux: Được phát triển bởi Đại Học Berkeley, đây không phải là phiên bản thương mại, do vậy ít được phổ biến. Có rất nhiều tiện ích dành cho việc phát triển hệ thống và lập trình. Hỗ trợ đầy đủ các shell trên Unix.

Corel Linux: Phát triển bởi hãng Corel, dễ setup, có graphic interface (giao diện đồ họa) khá giống Windows NT kể các tool và utility. Tuy nhiên các config tool chưa hoạt động tốt.

Open Linux: Do hãng Caldera phát triển, dễ cài đặt cũng như sử dụng. Giao diện KDE. Thích hợp cho người sử dụng tại gia đình.

Và còn rất nhiều release khác như Turbo Linux, Linux PPC, Debian Linux, Infomagic Linux, Softlanding Linux System Release (SLS) v.v..

Ngoài ra, hiện nay còn có một dòng Linux gọi là Live-CD Linux (có thể trực tiếp trên CD - không cần cài đặt) như Ubuntu, Knoppix, thích hợp với các beginner Linux

1.2. Tổng quan về quản lí bộ nhớ ngoài

Khi cần lưu trữ các chương trình, dữ liệu để sử dụng trong thời gian lâu dài thì bắt buộc cả hệ thống phải sử dụng đến bộ nhớ ngoài (đĩa từ, băng từ, CD, DVD...). Để cho hệ thống làm việc có hiệu quả và ổn định thì cần thiết phải có cơ chế quản lí thích hợp để đảm bảo cho quá trình vận hành máy không xảy ra xung đột, ảnh hưởng đến hiệu năng của máy. Muốn làm được điều này thì các hệ điều hành cần đảm bảo được các chức năng sau:

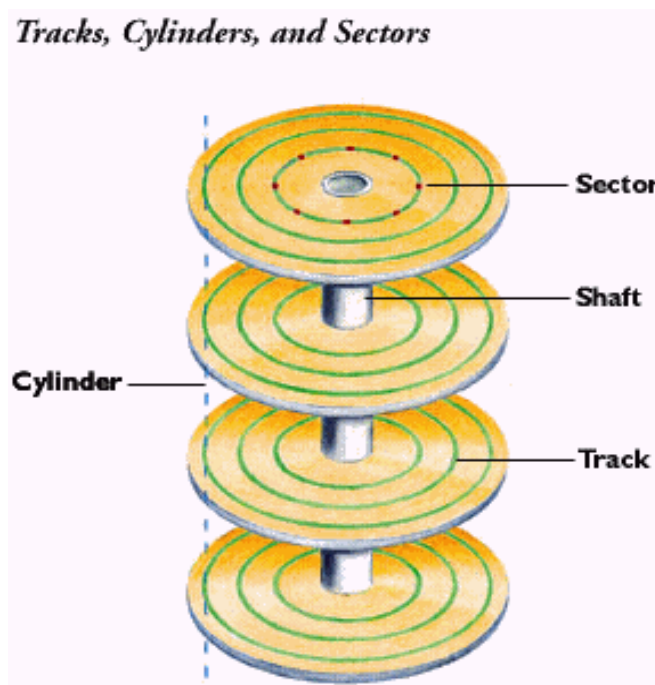
- Quản lý không gian nhớ tự do trên bộ nhớ ngoài (Free space Manage)
- Cấp phát không gian nhớ tự do (Allocation Methods)
- Cung cấp các khả năng định vị bộ nhớ ngoài
- Cơ chế lập lịch cho bộ nhớ ngoài

CHƯƠNG 2: CẤU TRÚC VẬT LÝ-THƯ MỤC THIẾT BỊ

2.1. Cấu trúc vật lý

Xét cấu trúc vật lý của đĩa từ: đĩa bao gồm một hoặc nhiều lá đĩa đặt đồng trục. mỗi mặt đĩa chia thành các rãnh tròn đồng tâm gọi là track, mỗi track được chia thành các cung gọi là sector, tập hợp các track cùng thứ tự trên mặt đĩa gọi là cylinder (từ trụ)

Trên mỗi mặt đĩa có một đầu từ đọc/ghi dữ liệu (Read/Write Heads), để điều khiển đầu từ đọc/ghi dữ liệu cần có một trình điều khiển đĩa (Disk Controller).



Hình 2.1: Tracks, cylinder và sectors.

Thông tin trên đĩa được tham chiếu bởi các thành phần: ổ đĩa, mặt đĩa, track, sector.

Hệ điều hành xem đĩa như mảng một chiều mà thành phần các khối đĩa (Disk Block). Mỗi đĩa ghi các thông tin về mặt đĩa, track, sector, mà hệ điều hành có thể định vị trên đó.

2.2. Thư mục thiết bị

Trên mỗi đĩa thường có một thư mục thiết bị (device Directory) cho biết đĩa gồm những thông tin gì, độ dài, kiểu, người sở hữu, thời điểm khởi, vị trí, được phân bổ không gian như thế nào. Thư mục thiết bị nằm trong vùng nhớ đặc biệt.

CHƯƠNG 3: CÁC PHƯƠNG PHÁP QUẢN LÝ KHÔNG GIAN NHỚ TỰ DO

3.1 Phương pháp dùng bit vector (bitmap)

Không gian đĩa được chia thành các khối (block) và được đánh số từ 0...max.

Ví dụ: đĩa mềm 1.44Mb, 2 mặt, 80 track/1 mặt, 18 sector/1 track được đánh số như sau:

Head 0, track 0, sector 1	Block 0
.....	...
Head 0, track 0, sector 18	Block 18
Head 1, track 0, sector 1	Block 19
.....	...
Head 0, track 1, sector 1	Block 37
.....	...
.....	...
Head 1, track 79, sector 18	Block 2879

Mỗi đĩa khi sử dụng một bit để đánh dấu trạng thái. Khối đĩa nào đã sử dụng thì bit trạng thái có giá trị bằng 1, chưa sử dụng thì có giá trị bằng 0. Tập hợp các kí hiệu 0,1 tạo thành bitvector (bitmap). Đọc thông tin trong bitmap hệ điều hành có thể xác định được không gian tự do trên đĩa.

Ví dụ: Cho không gian đĩa từ hình 1.1 các khối 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 17, 18, 25, 26, 27 là các khối đĩa tự do. Khi đó bitmap quản lí không gian nhớ tự do như sau:

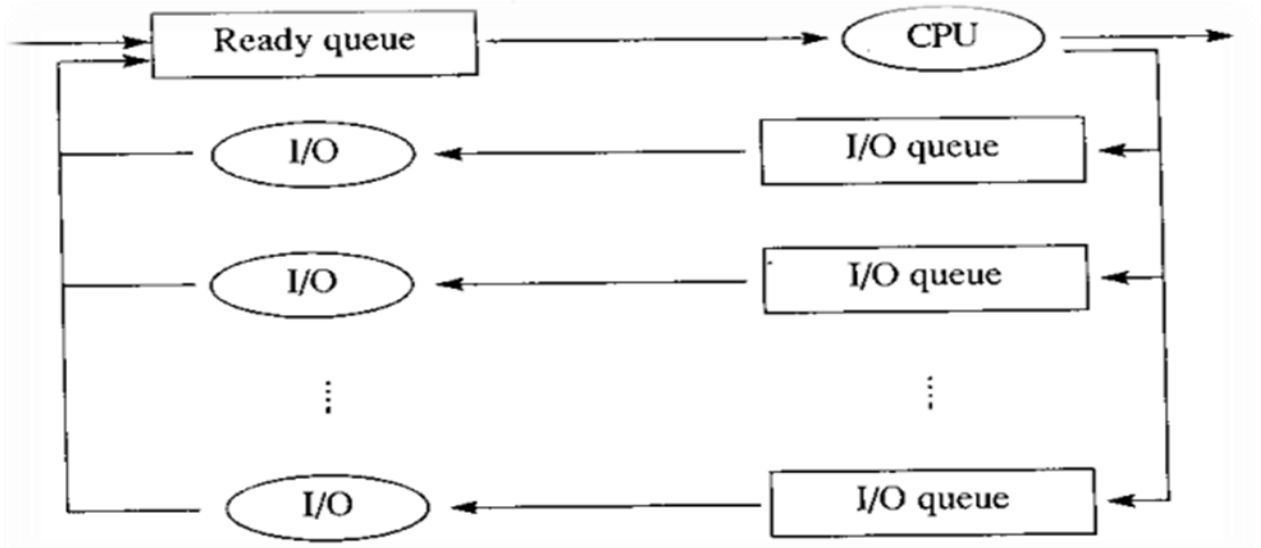
11000011000000111001111110001111...

Phương pháp bitmap có ưu điểm là cài đặt đơn giản, dễ quản lí, dễ tìm kiếm những khối đĩa tự do liên tục trên đĩa nhưng tốn không gian lưu trữ dành cho bitmap (mỗi khối sẽ tốn 1 bit).

3.2. Phương pháp liệt kê Free List

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

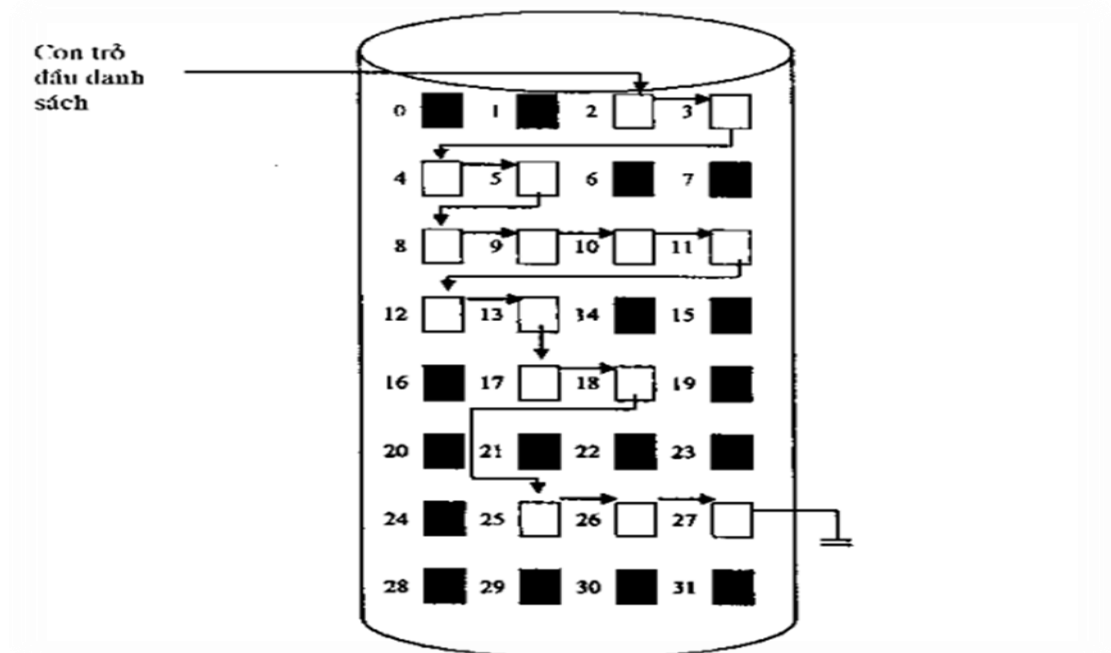
- Ưu điểm của phương pháp này là tiết kiệm không gian nhớ nhưng làm



tăng thời gian truy nhập dữ liệu.

3.3. Phương pháp lập nhóm (Grouping)

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



Hình 3.1 Mô tả không gian đĩa từ

Ví dụ như hình 1.1 ta có bảng quản lí không gian nhớ tự do 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 (...)

3.4. 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 lí 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.

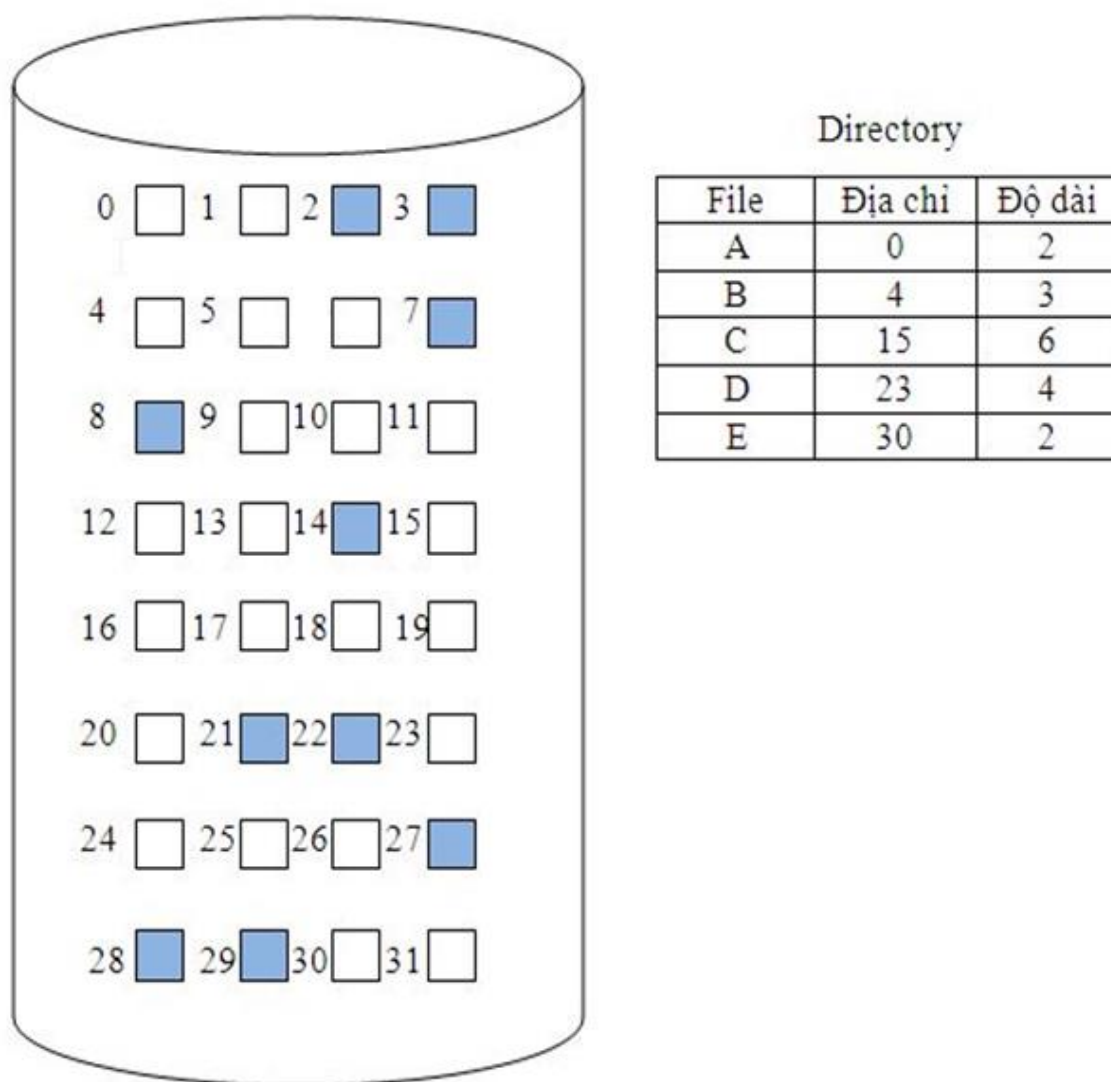
Ví dụ theo hình 3.1, ta có danh sách quản lí không gian nhớ tự do như sau:

Danh sách	Số lượng
2	4
8	6
17	2
25	3

CHƯƠNG 4. CÁC PHƯƠNG PHÁP CẤP PHÁT KHÔNG GIAN NHỚ TỰ DO

4.1. Cấp phát kề (contiguous)

Để phân bổ 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.



Hình 4.1 : Không gian nhớ được cấp phát liền kề

Các thuật toán sử dụng để tìm không gian nhớ tự do thích hợp để cấp phát cho file lưu trữ :

First –fit : Cấp phát vùng nhớ tự do đầu tiên đủ lớn tìm được cho file. Tìm kiếm có thể bắt đầu tại đầu tập hợp các vùng nhớ trống hay tại điểm kết thúc của tìm kiếm first-fit trước đó. Thuật toán dừng khi tìm kiếm được vùng nhớ đầu tiên đủ điều kiện cho file. Do đó thuật toán này thời gian cấp phát quyền sử dụng ô nhớ nhanh chóng nhưng gây lãng phí bộ nhớ.

Best-fit : Cấp phát vùng nhớ tự do nhỏ nhất đủ lớn tìm được. Chúng ta phải tìm toàn bộ danh sách, trừ danh sách được xếp thứ tự theo kích thước, thuật toán này tạo ra lỗ trống nhỏ nhất còn lại dư thừa. Do đó có thể tận dụng tối ưu không gian nhớ của đĩa. Nhược điểm thời gian chờ cấp phát tương đối dài, hiệu năng hoạt động chung của hệ thống giảm khi có nhiều chương trình thực thi đòi hỏi quyền cấp phát bộ nhớ.

Worst-fit: Cấp phát vùng nhớ tự do có kích thước lớn nhất cho file. Thuật toán này phải tìm danh sách trừ khi nó được xếp theo thứ tự kích thước. **Thuật toán này tạo ra vùng nhớ trống còn lại lớn nhất và có thể có ích hơn lỗ trống nhỏ từ thuật toán best-fit.**

Tuy nhiên, các giải thuật trên gặp phải vấn đề “**Phân mảnh ngoài (external fragmentation).**” Khi các quá trình được nạp và được xóa khỏi bộ nhớ, không gian bộ nhớ trống bị phân rã thành những mảnh nhỏ. Phân mảnh ngoài tồn tại khi tổng không gian bộ nhớ đủ để thỏa mãn một yêu cầu, nhưng nó không liên tục; vùng lưu trữ bị phân mảnh thành một số lượng lớn các mảnh nhỏ. Vấn đề phân mảnh này có thể rất lớn. Trong trường hợp xấu nhất, chúng có thể có một khối bộ nhớ trống nằm giữa mỗi hai quá trình. Nếu tất cả bộ nhớ này nằm trong một khối trống lớn, chúng ta có thể chạy nhiều quá trình hơn.

Chọn lựa first-fit so với best-fit có thể ảnh hưởng tới lượng phân mảnh. First-fit là tốt hơn đối với 1 số hệ thống, ngược lại best-fit là tốt hơn cho một số hệ thống khác. Một yếu tố khác là phần cuối của khối trống nào được cấp phát. Vấn đề không do giải thuật nào được dùng mà do phân mảnh ngoài.

☞ Ở hệ điều hành Linux, Linux trải tất cả các tập tin của hệ thống trên toàn

bộ bề mặt đĩa (các tập tin không ghi sát nhau như ở hệ điều hành Windows) và dữ liệu của tập tin khi ghi xuống đĩa được hệ thống cấp phát dung lượng lớn hơn kích thước của tập tin cần ghi, do đó khi kích thước thay đổi phần dữ liệu mới sẽ được thêm vào phần trống (đã được chứa sẵn từ trước khi tạo tập tin này), trong trường hợp vùng trống này không đủ thì hệ thống sẽ tìm một vị trí khác còn trống có thể chứa được tập tin này và dời tập tin này đến vị trí đó.

Sự phân mảnh chỉ xảy ra khi dung lượng của đĩa không còn đủ để chứa tập tin mà không cần phải chia nhỏ tập tin này ra. Trường hợp này chỉ xảy ra khi dung lượng của đĩa đã sử dụng quá 80%.

Với cách thực hiện như trên như vậy Linux đã chống phân mảnh ngay từ khi thực hiện ghi tập tin trên đĩa, sự phân mảnh chỉ xảy ra khi dung lượng đĩa còn ít hơn 20%.

**** Ưu điểm của phương pháp cấp phát liên tục :**

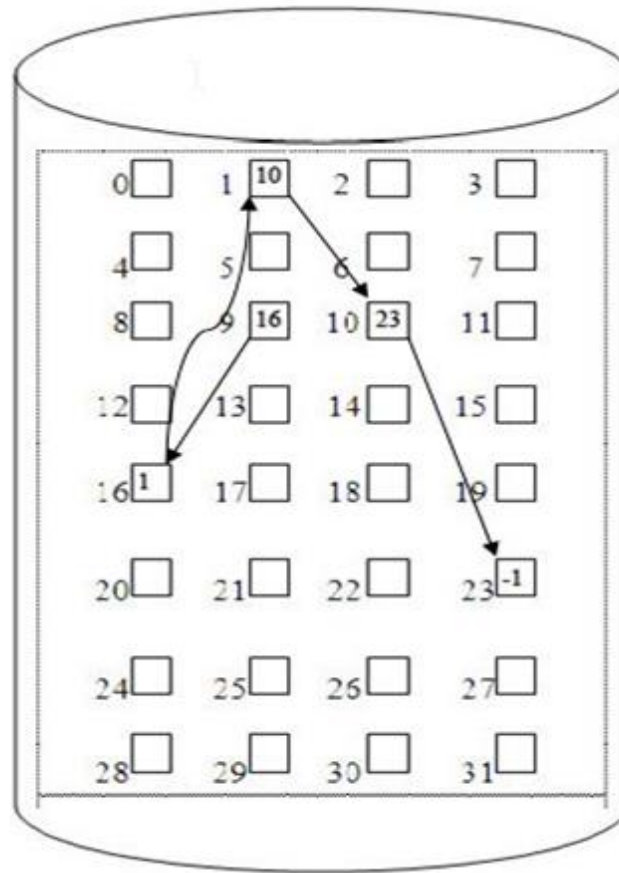
Hỗ trợ cho phương pháp truy nhập tuần tự và truy nhập trực tiếp

**** Hạn chế :**

- Phải chọn được thuật toán tối ưu để tìm các cùng không gian nhớ tự do cấp phát cho file (một trong 3 thuật toán first-fit, Best-fit và worst-fit).
- Có thể xảy ra trường hợp không đủ số khối đĩa tự do liên tiếp chưa sử dụng để cấp phát cho file (kích thước file lớn hơn cùng các khối đĩa tự do liên tục lớn nhất trong bộ nhớ còn trống).
- Trong trường hợp các khối đĩa tự do nằm tản mạn sẽ không sử dụng được gây tình trạng lãng phí bộ nhớ trong khi dung lượng không đủ để cấp phát quyền truy xuất cho file có kích thước dung lượng lớn.

4.2. Cấp phát liên kết (Linked).

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



File	Start	End
ABH	9	23

Hình 4.2 : Cấp phát không gian đĩa liên kết

**** Ưu điểm của phương pháp cấp phát liên kết :**

- Hỗ trợ phương pháp truy nhập tuần tự. Tận dụng được các khối đĩa tự do nằm tản mạn trong vùng nhớ.

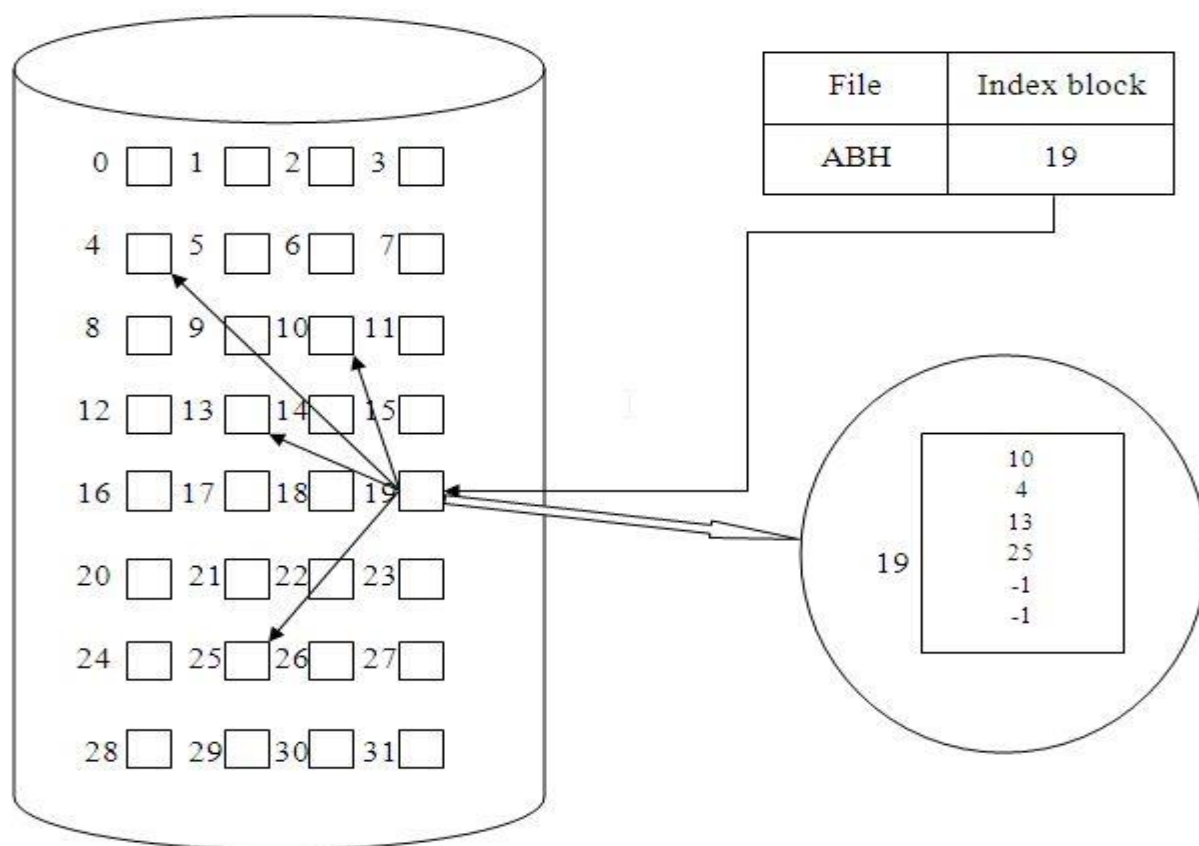
**** Hạn chế**

- Không hỗ trợ truy nhập trực tiếp.
- Độ tin cậy không cao do có nguy cơ bị mất các con trỏ liên kết.
- Tốn không gian nhớ để lưu trữ các con trỏ.

4.3. Cấp phát theo chỉ số.

Sử dụng 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 của 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 ra khỏi danh sách các khối đĩa tự do và cập nhật vào khối chỉ số của file đó.



Hình 4.3.1 :Cấp phát theo chỉ số

**** Ưu điểm của phương pháp cấp phát theo chỉ số :**

- Hỗ trợ truy nhập trực tiếp.
- Tận dụng được khối nhớ tự do nằm tản mạn.

**** Hạn chế:**

- Không thể truy nhập tuần tự.
- Tồn không gian nhớ giành cho khối chỉ số.

Để giải quyết vấn đề tồn dung lượng bộ nhớ cần đưa ra cơ chế phù hợp để điều chỉnh dung lượng của khối chỉ số sao cho tối ưu nhất.

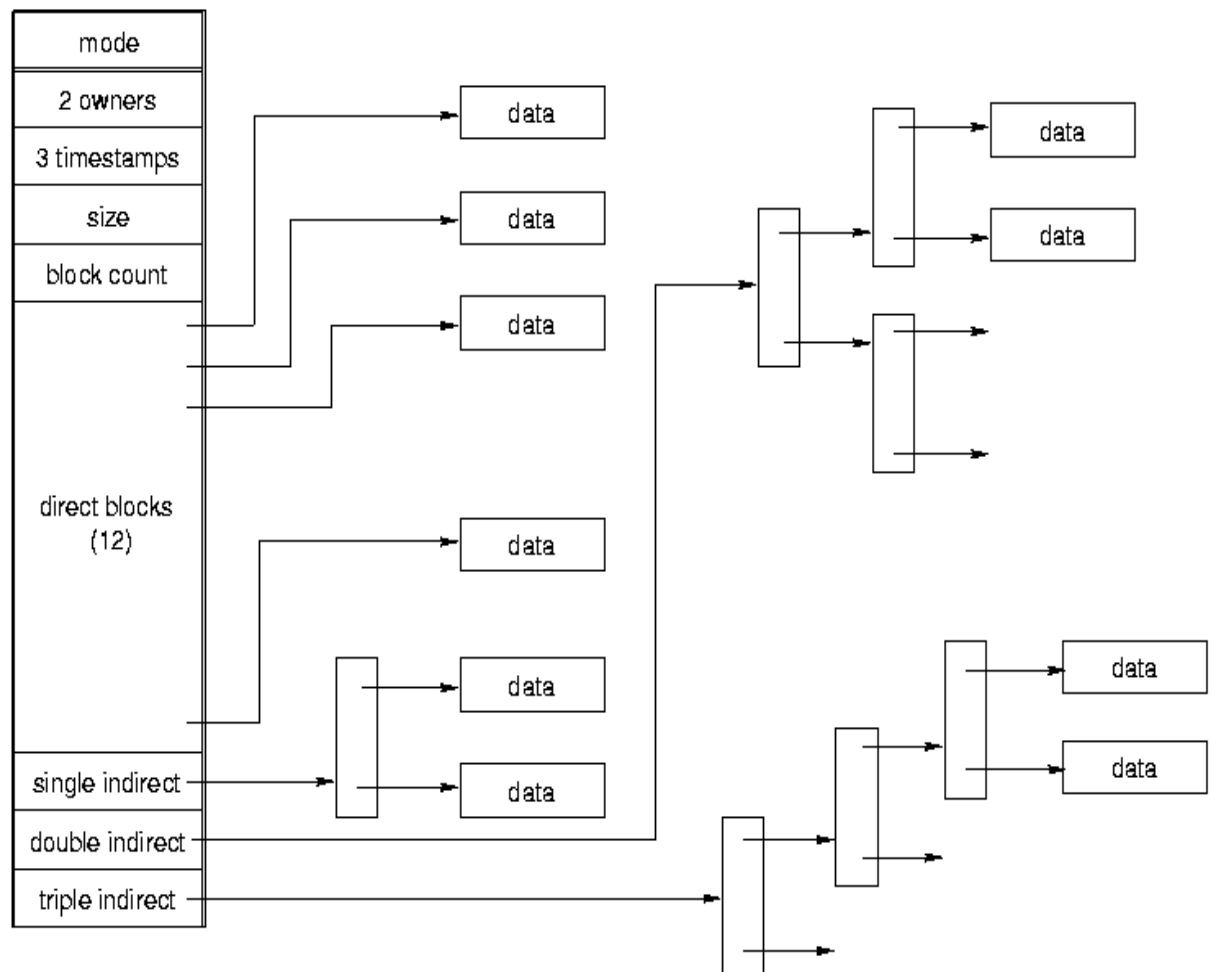
*** Cơ chế liên kết (link scheme) :** Một khối chỉ số thường là một khối đĩa. Do đó, nó có thể được đọc và viết trực tiếp bởi chính nó. Để cho phép đối với các tập tin lớn, chúng ta có thể liên kết nhiều khối chỉ số với nhau. Thí dụ, một khối chỉ số có thể chứa một header nhỏ cho tên tập tin và một tập hợp của các địa chỉ 100 khối 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 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 chỉ số cấp 1 để chỉ tới khối chỉ số cấp 2. Khối 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 tiếp tục tới cấp 3 hay cấp 4, tùy thuộc vào kích thước tập tin lớn nhất 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 1,048,567 khối dữ liệu, cho phép tập tin có kích thước tối đa 4GB.

*** Cơ chế kết hợp (combined scheme):** một biến dạng khác được dùng trong USF 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 tới 48 KB 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 địa 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 chứa dữ liệu. Sau đó, có con trỏ khối gián tiếp đôi (double indirect blocks). 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 blocks). Với phương pháp này, số khối có thể được cấp phát tới một tập tin vượt quá lượng không gian có thể đánh địa chỉ bởi các con trỏ tập tin 4 bytes hay 4GB.

***** Inode (index node) là một khái niệm cơ bản trong Linux filesystem.** Mỗi đối tượng của filesystem được đại diện bởi một inode. inode là một cấu

trúc dữ liệu trong hệ thống tệp truyền thống của các họ Unix ví dụ như UFS hoặc EXT,EXT3. Inode lưu trữ thông tin về 1 tệp thông thường, thư mục, hay những đối tượng khác của hệ thống tệp tin.



Hình 4.3.2: cấu trúc Inode

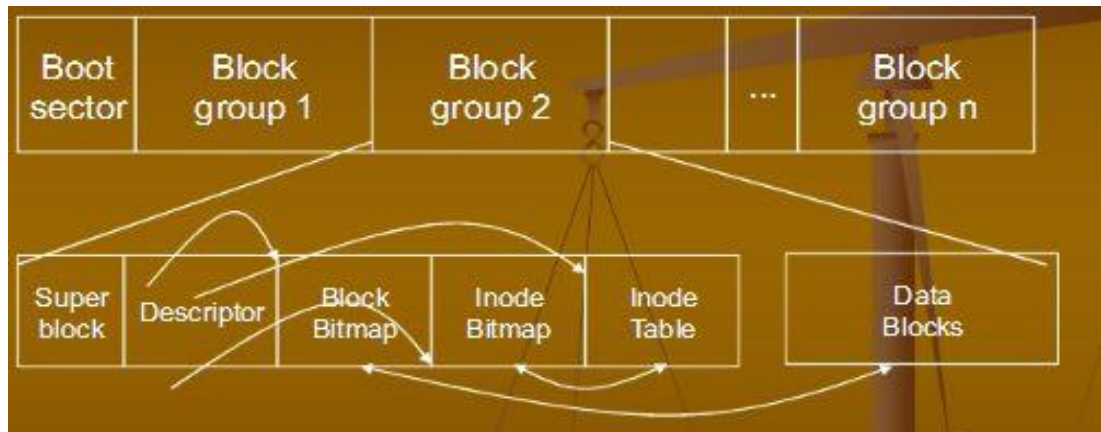
Cơ chế cấp phát lập chỉ số có một số hạn chế về năng lực cấp phát liên kết. Các khối chỉ số có thể được lưu trữ trong cache trong bộ nhớ; Nhưng khối dữ liệu có thể trải rộng khắp phân khu.

- Cấu trúc phân vùng trong linux



Hình 4.3.3: Cấu trúc phân vùng trong linux

- Boot block: Chứa thông tin vùng khởi động.
- Super block: Chứa thông tin về phân vùng.
- Inode list: Danh sách các inode trong file system.
- Data/Index block: Danh sách các block dữ liệu.



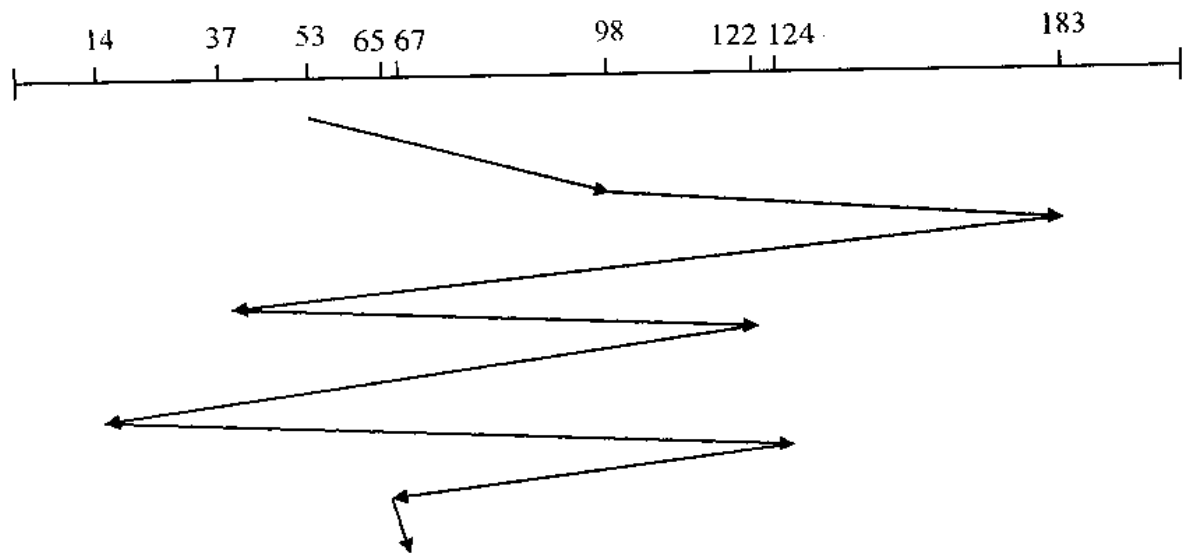
Linux sử dụng các inode table chứa danh sách các inode trong FS. Các inode có các chỉ số khác nhau, có một inode gọi là root inode trong mỗi FS. inode là khởi đầu để vào được các FS sau khi gọi thực hiện hệ thống phép ghép (mount) FS đó vào cây thư mục gốc. Mỗi đối tượng của filesystem được đại diện bởi một inode. Tất cả các file đều có những thuộc tính sau đây: Loại file, permissions, chủ sở hữu, nhóm, kích thước file, thời gian truy cập, thay đổi, sửa đổi file, thời gian file bị xóa, số lượng liên kết, thuộc tính mở rộng, danh sách truy cập file.

CHƯƠNG 5: LẬP LỊCH CHO ĐĨA

5.1. First Come First Served (FCFS)

Để truy nhập tới một file, hệ thống sẽ tổ chức một hàng đợi các yêu cầu phục vụ của các track (lưu trữ dữ liệu của file cần truy nhập). Track nào có yêu cầu phục vụ trước thì đầu từ đọc/ghi sẽ dịch chuyển tới đó trước. Đây là một thuật toán lập lịch đơn giản, dễ lập trình nhưng không cung cấp được một dịch vụ tốt, số track mà đầu đọc phải di chuyển là nhiều, và hiệu quả của thuật toán còn phụ thuộc vào các thứ tự trên hàng đợi.

Ví dụ: File F1 được phân bổ lần lượt tại các track có số thứ tự sau đây: 98, 183, 37, 122, 14, 124, 65, 67. Giả sử hiện tại đầu đọc/ghi đang ở vị trí 53. Sơ đồ dịch chuyển đầu đọc/ghi theo thuật toán FCFS được thể hiện như hình sau:

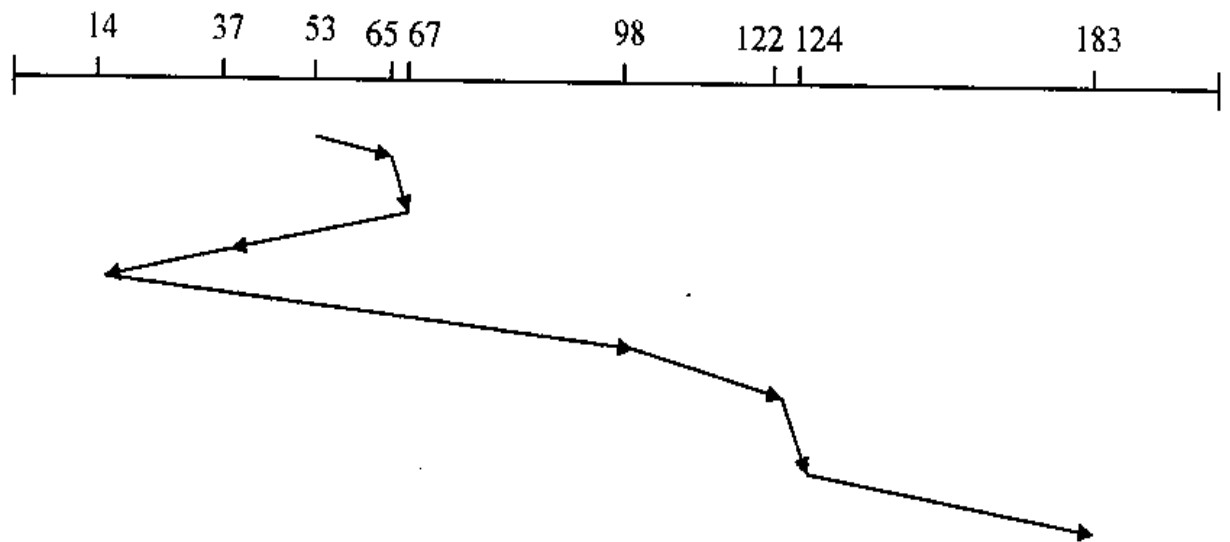


Hình 5.1 – Thuật toán lập lịch FCFS

5.2. Shortest Seek Time First (SSTF)

Thuật toán này chọn track nào có thời gian di chuyển đầu đọc/ghi ngắn nhất thì thực hiện trước. Có số di chuyển giảm nhưng có thể gây ra một số yêu cầu không được phục vụ.

Theo ví dụ trên, sơ đồ của thuật toán được thể hiện như sau:

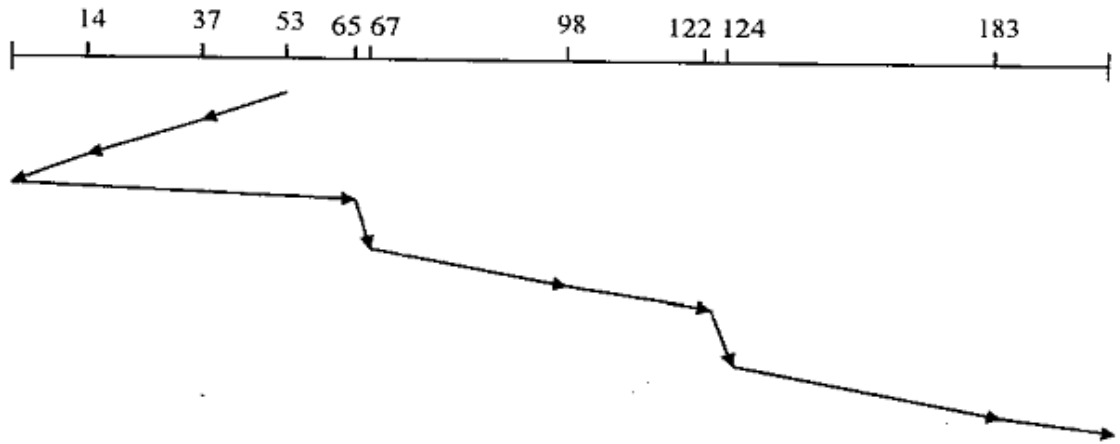


Hình 5.2 – Thuật toán lập lịch SSTF

5.3 Scan

Trong thuật toán này, đầu đọc của đĩa di chuyển từ 1 phía (ví dụ bên ngoài hoặc bên trong đĩa) sang phía bên kia để phục vụ một số yêu cầu đọc, sau đó di chuyển ngược lại...quá trình này được lặp đi lặp lại. Đặc điểm của thuật toán này là phương thức hoạt động như thang máy, số bước đầu đọc/ghi phải di chuyển giảm.

Cũng theo ví dụ như trên ta có sơ đồ sau:

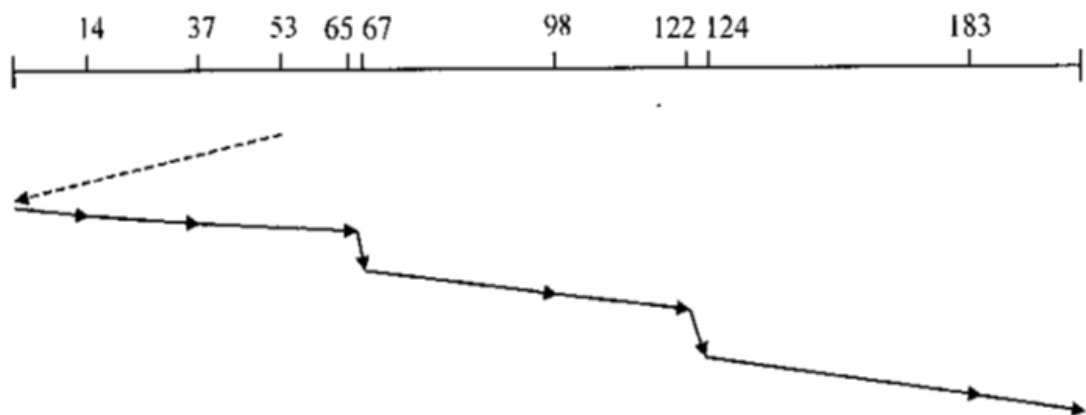


Hình 5.3 Thuật toán lập lịch Scan

5.4. C – Scan

Đầu đọc di chuyển từ một phía (trong/ngoài) sang phía bên kia và thực hiện một số yêu cầu. Khi sang phía bên kia, đầu đọc quay trở lại nhưng trong khi quay trở lại không thực hiện một yêu cầu nào (quét một chiều).

Theo ví dụ trên, sơ đồ của thuật toán được thể hiện như sau:

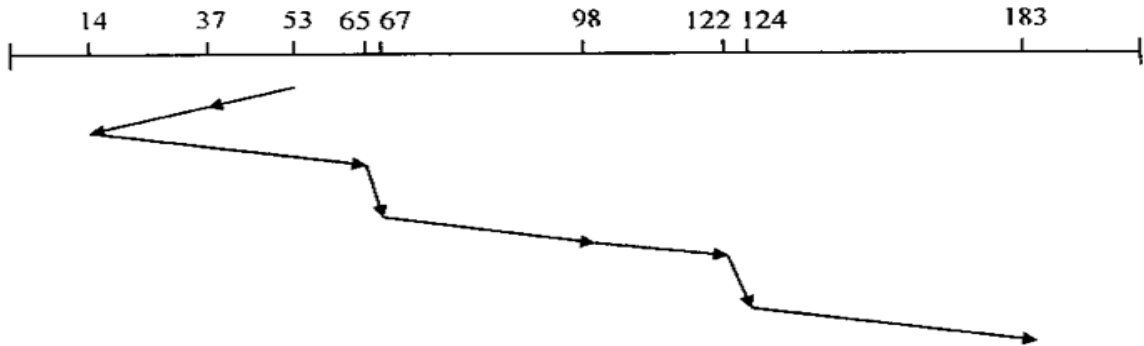


Hình 5.4 Thuật toán lập lịch C - Scan

5.5. Look

Tương tự Scan nhưng trong thuật toán này đầu 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).

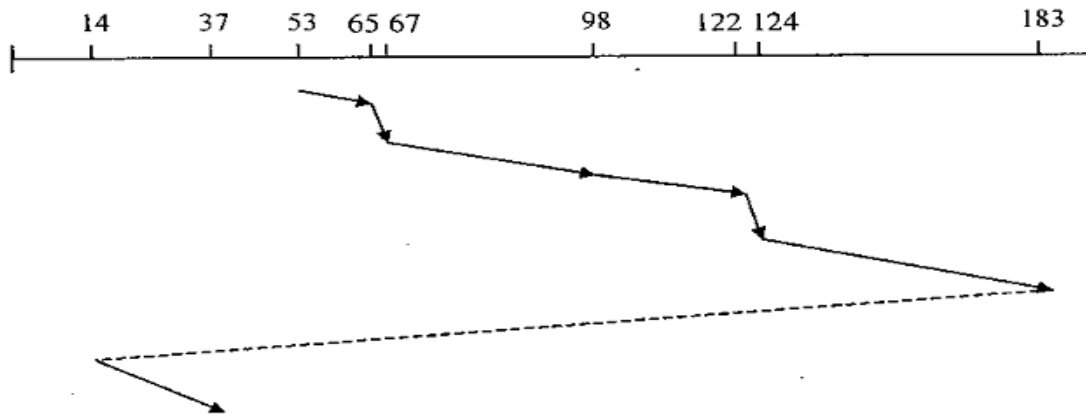
Như ví dụ trên, sơ đồ của thuật toán này như sau:



Hình 5.5 Thuật toán lập lịch Look

5.6. C – Look

Thuật toán này tương tự Look nhưng đầu từ đọc/ghi không phục vụ đường về. Theo ví dụ trên, sơ đồ của thuật toán này như sau:



Hình 5.6 Thuật toán lập lịch C - Look

Chú ý: Lựa chọn một số giả thuật lập lịch cho đĩa:

- FCFS là thuật toán phù hợp khi các track cần truy xuất là liên tục.
- SSTF phổ biến và có hiệu quả tốt
- SCAN và LOOK thích hợp cho những hệ thống phải truy xuất dữ liệu lớn

CHƯƠNG 6: So sánh với nguyên lý quản lý bộ nhớ ngoài đã học.

6.1. Lịch sử phát triển

Linux được bắt đầu như một dự án cá nhân của một sinh viên người Phần Lan tên là Linus Torvalds, mục đích ban đầu là để tạo ra một nhân hệ điều hành miễn phí hoàn toàn.

Linux miễn phí và là mã nguồn mở ngay từ khi mới ra đời – năm 1991. Linux bắt đầu như một dự án “chỉ làm cho vui”, nhưng nó đã nhanh chóng trở thành một trong những dự án mã nguồn mở lớn nhất từ trước đến nay.

Windows, phiên bản Windows 1.0 của Microsoft được phát hành vào năm 1985 và không giống như Linux, nó là một sản phẩm mã nguồn đóng hoàn toàn được Microsoft bán theo chương trình cấp phép.

6.2. Xử lý, can thiệp vào mã nguồn

- Điểm khác biệt lớn nhất của 2 Linux và Window là:

- + Khả năng truy cập
- + Chỉnh sửa mã nguồn

Linux	Window
Được cấp phép GNU Public License nên nó cho phép người dùng truy cập mã nguồn đến tận lõi của hệ điều hành.	Được bảo mật vô cùng cẩn thận
Cho phép người dùng chỉnh sửa, nâng cấp các phần mềm và hệ điều hành nhanh hơn. cho phép các nhà phát triển truy cập vào mã nguồn, nhiều kẻ xấu sẽ tìm	Điều này là không thể, nhưng không có nghĩa là nó an toàn 100% nhé. Một lần nữa, với một người sử dụng bình thường thì họ thường không quan tâm đến việc liệu họ có xem được mã nguồn tạo nên hệ điều

kiểm và lợi dụng các lỗ hổng để phát tán các phần mềm độc hại đến người dùng.	hành mà họ đang sử dụng hay không.
---	------------------------------------

6.3. Ứng dụng – phần mềm

Linux	Window
Dễ dàng thêm các ứng dụng mới và xóa chúng đi khi không còn dùng đến.	Không có kho ứng dụng của riêng mình, phải lên Google và tìm kiếm những phần mềm của bên thứ 3 để cài đặt
Không có Registry Các ứng dụng trên Linux sẽ lưu thiết lập trên cơ sở chương trình, dưới sự phân cấp người dùng	Registry như là xương sống của con người vậy. Nó là một cơ sở dữ liệu chủ cho toàn bộ các thiết lập của người dùng nằm trên máy tính. Registry nắm giữ thông tin ứng dụng, mật khẩu người dùng, thông tin thiết bị... và rất rất nhiều thứ khác nữa.

6.4. Đối tượng sử dụng

Linux	Window
Những người thành thạo, chuyên nghiệp, họ biết họ đang làm gì	Chiếm đông đảo người dùng vì nó đơn giản không cần phải có kiến thức sâu về xử lý hệ điều hành

6.5. Bảo mật

	Windows	Linux
Malware	As of 2009, well over 2 million malware programs target Windows.	As of 2006, more than 800 pieces Linux malware had been discovered.
Open vs. Closed	Claims its platform is more secure because of a comprehensive approach to security using the Security Development Lifecycle	Claims its platform is more secure because all of its code is reviewed by so many people that bugs are detected (referred to as Linus Law).
Response speed	Critical bug fixes are released only once a month after extensive programming and testing and certain bugs have been known to go unpatched for months or even year	Bugs can be fixed and rolled out within a day of being reported (often within hours), though usually it takes a few weeks before the patch is available on all distributions.
User Accounts	In Windows Vista, all logged-in sessions (even for those of "administrator" users) run with standard user permissions, preventing malicious programs (and inexperienced users) from gaining total control of the system. Processes that require administrator privileges can be run using the User Account Control framework.	Users typically run as limited accounts, having created both administrator and at least one user account during installation. In most Linux distributions, there are commands that will temporarily grant elevated permissions to processes that need it. In practice, this can be very dangerous, as any error can lead to severe damage to the system.

Hình 6.1 Ảnh thông tin bảo mật

Cả hai hệ điều hành đều bảo mật rất tốt

⇒ Nhìn chung, nếu bạn chỉ là một người dùng bình thường, không quan tâm đến cách hoạt động bên trong của hệ điều hành thì Windows là một sự lựa chọn phù hợp nhất dành cho bạn.

Còn nếu bạn là một kỹ sư hoặc là một người dùng máy tính chuyên nghiệp thực thụ, thậm chí là một người đam mê với các dòng lệnh, muốn toàn quyền kiểm soát hệ thống thì Linux là một lựa chọn tuyệt vời dành cho bạn.

KẾT LUẬN

Như vậy ta có thể thấy từ khi ra đời đến nay, tính đến thời điểm hiện tại, Linux đã có rất nhiều biến thể và các phiên bản khác nhau, được xây dựng và phát triển riêng biệt bởi các công ty phần mềm và các cá nhân. Hiện nay, sau hơn 20 năm tồn tại và phát triển, Linux được sử dụng rộng rãi trên toàn thế giới, trên các máy tính cá nhân, các máy chủ, đến các thiết bị di động, máy nghe nhạc, máy tính bảng, các máy ATM và thậm chí trên cả các siêu máy tính... Ngày nay, Linux được xem là biểu tượng của sự chia sẻ cộng đồng, được phát triển bởi cộng đồng và được ủng hộ vì hoàn toàn miễn phí. Với những tính năng ngày càng tiên tiến, hệ thống File được nâng cấp, hoàn thiện để đáp ứng nhu cầu của người sử dụng. Hệ điều hành Linux sẽ ngày càng phát triển trong tương lai.

Tài Liệu Tham khảo

- Nguyên lý hệ điều hành (Nguyễn Hải Châu – ĐH công nghệ).
- Nguyễn Thanh Hải, Giáo trình Nguyên lý hệ điều hành, 2016.
- Abraham Silberschatz, Galvin, Gagne, Operating System Concepts 8th edition.
- Tài liệu điện tử trên internet.
- Vi.scribd.com.
- www.nhchau.com.
- Uet.vnu.edu.vn.
- Tailieutonghop.com.
- Tailieu.vn.
- <https://www.getwox.com/vi/linux-vs-windows/>.