

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương I: TỔNG QUAN**

Tổng quan về môn học và về Hệ điều hành

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết

3. Mục đích, yêu cầu:

- Kiến thức: Hiểu được nguyên lý thiết kế, tổ chức chung của Hệ điều hành cho các hệ thống tính toán.
- Có khả năng hiểu và thiết kế, xây dựng được một số chức năng tổ chức, quản lý, cấp phát tài nguyên của hệ thống cho các ứng dụng bao gồm: CPU, Bộ nhớ, Thiết bị ngoại vi.....
- Kỹ năng: Học viên cần có hiểu biết cơ bản về cấu trúc máy tính và sử dụng thành thạo một ngôn ngữ lập trình như C.
- Thái độ, chuyên cần: Cần có thái độ nghiêm túc trong nghe giảng và làm bài tập đầy đủ.

4. Nội dung:

a) Nội dung chi tiết:

I. Giáo viên giảng dạy:

Họ tên: Trịnh Minh Châu

Điện thoại: 0903266163

Email: chautm@yahoo.com

Phòng làm việc/ bộ môn: Khoa học Máy tính khoa CNTT, tầng 2 nhà A HVKTQS

II. Giới thiệu chung về nội dung môn học:

Đây là môn học bắt buộc đối với học viên thuộc nhóm Công nghệ thông tin. Môn học hướng tới trang bị cho học viên các khái niệm và định hướng phát triển, hoàn thiện, cải tiến hệ điều hành, phần mềm hệ thống cho các hệ thống tính toán nói chung. Học viên nắm được nguyên tắc tổ chức và quản lý cấp phát tài nguyên hệ thống: CPU, bộ nhớ và các thiết bị ngoại vi bao gồm thiết bị lưu trữ ngoài đặc biệt là ổ đĩa cứng thông qua các nguyên lý và thuật toán khác nhau; trên cơ sở đó có thể lựa chọn giải pháp phù hợp đối với hệ thống tính toán - máy tính điện tử số (PC, Server, Mainframe, Notebook, PDA, Smartphone, các hệ thống phát triển chuyên dụng).

III. Giáo trình và tài liệu tham khảo

Hiện môn học chưa có giáo trình, tuy nhiên tài liệu chính dùng thay cho giáo trình được cung cấp qua bản điện tử và bản “hard copy” có thể tìm thấy tại thư viện Học viện và/hoặc thư viện khoa CNTT.

- Tài liệu chính (tài liệu 1): Operating System Concepts, A. Silberschatz & P Galvin, John Wiley & Sons, sixth Edition 2002
- Tài liệu 2: Operating System: Design & Implementation, 2nd edition. Andrew S. Tanenbaum. Prentice Hall
- Tài liệu 3: Nguyên lý Hệ điều hành, Nguyễn Gia Định, Nhà xuất bản Khoa học và kỹ thuật, 2005
- Tài liệu 4: Computer Organization and Architecture, William Stallings, Prentice-Hall International, 4th Edition.
- Tài liệu 5: Modern Operating Systems, Andrew S. Tanenbaum. Prentice-Hall

IV. Thời lượng và thời khóa biểu

Thời lượng môn học theo thời khóa biểu: 45 tiết

Giờ tín chỉ đối với các hoạt động:

- Nghe giảng lý thuyết: 33 tiết.
- Làm bài tập và thảo luận trên lớp: 12 tiết.
- Thực hành, thực tập (ở PTN, nhà máy, thực tập...):
- Hoạt động theo nhóm: 15 tiết
- Tự học: 60 tiết

V. Đánh giá và cho điểm

Theo qui định của Học viện, điểm tổng kết môn học bao gồm 3 (ba) thành phần:

- Chuyên cần: 10%
- Thường xuyên: 20%

Có hai hình thức cho điểm thường xuyên: làm 2 (hai) bài kiểm tra trên lớp mỗi bài 1 (một) tiết ứng với 10% hoặc học viên viết chương trình mô phỏng các thuật toán quản lý tài nguyên chính của hệ thống máy tính bao gồm: Lập lịch cho CPU, Cấp phát bộ nhớ liên tục, Xử lý lỗi trang và Lập lịch cho đầu từ. Lớp học viên cũng có thể cùng giáo viên lựa chọn hình thức Viết tiểu luận.

- Thi kết thúc môn học: 70%

b) Nội dung thảo luận

c) Nội dung tự học

d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương I: TỔNG QUAN**

I.1 Giới thiệu tổng quan về Hệ điều hành

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 2 tiết, tự học 3 tiết

3. Mục đích, yêu cầu:

Hiểu được Bản chất, vai trò của Hệ điều hành trong các hệ thống Máy tính khác nhau,

Nắm được quá trình phát triển cũng như các dạng hệ điều hành khác nhau qua các giai đoạn phát triển khác nhau cũng như các mô hình ứng dụng/ mô hình hệ thống khác nhau.

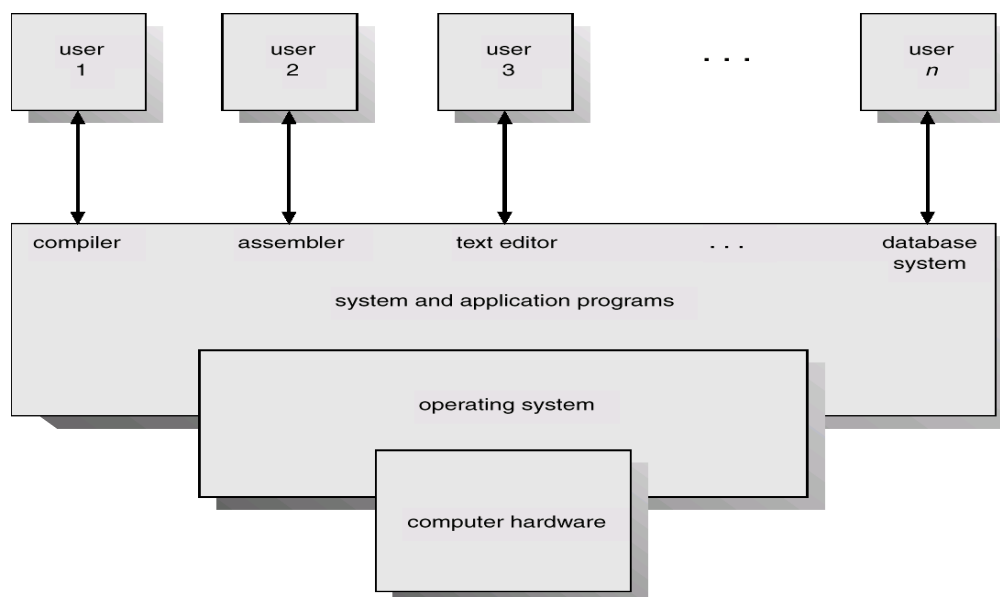
4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

I.1.1 Hệ điều hành là gì?

Máy tính thực sự vô dụng nếu không có phần mềm. Phần mềm có thể chia thành 2 dạng chính là Phần mềm ứng dụng và Phần mềm hệ thống; Hệ điều hành (HĐH) là một phần mềm hệ thống đặc biệt. HĐH là một chương trình quản lý phần cứng máy tính. Nó cung cấp nền tảng cơ sở cho các chương trình ứng dụng và hoạt động như một đối tượng trung gian giữa người sử dụng máy tính và phần cứng máy tính. HĐH là một thành phần tối quan trọng tất cả các hệ thống máy tính, luôn luôn ở trạng thái sẵn sàng tại mọi thời điểm.

Một hệ thống máy tính có thể được phân chia một cách tương đối thành 4 thành phần sau : Phần cứng, Hệ điều hành, Phần mềm ứng dụng và Người sử dụng



Mục đích, Chức năng của hệ điều hành

Dưới góc nhìn khác nhau của các đối tượng khai thác sử dụng máy tính khác nhau có thể nhận thấy mục đích cũng như chức năng của HĐH có thể khác nhau.

- Dưới góc độ hệ thống
- Dưới góc độ người sử dụng (NSD) như người khai thác máy tính
- Dưới góc độ NSD như người phát triển phần mềm

1.1.2 Lịch sử phát triển của HĐH

Có một sự gắn bó giữa lịch sử phát triển của HĐH với lịch sử phát triển của các hệ thống máy tính.

Đặc tính kiến trúc/ thiết kế của hệ thống tính toán có ảnh hưởng tới việc lựa chọn HĐH, và việc song hành tồn tại và cùng phát triển của các dạng/ mô hình hệ thống khiến cho việc tồn tại đồng thời và phát triển khá nhiều cấu hình HĐH khác nhau.

Máy tính lớn (mainframes) – được chia làm 3 loại

- Xử lý bó đơn (Simple batch)
- Đa chương (multiprogrammed)
- Đa nhiệm (time-sharing, multitasking)

Máy các nhân (personal computer)

- Máy để bàn (Desk top)
- Máy xách tay (Laptop, Note Book, Pad)
- Máy chủ PC (PC Server)

Thiết bị thông minh/nhúng (smart device)

- Điện thoại thông minh (smart phone)
- TV kết nối internet (smart TV)

Hệ thống đa xử lý – xử lý song song (Multiprocessor system)

Các thiết bị/hệ thống điều khiển bằng chương trình khác

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)
 - Tài liệu 1: Chương 1 từ trang 3 đến trang 25
 - Tài liệu 2: Chương 1 từ trang 3 đến trang 20
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương I: TỔNG QUAN**

I.1 Giới thiệu tổng quan về Hệ điều hành

I.1.3. Cấu trúc phần cứng của hệ thống máy tính

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết bài giảng và 3 tiết tự học

3. Mục đích, yêu cầu:

Nắm được tài nguyên của hệ thống, hoạt động của máy tính và các khái niệm căn bản liên quan tới phần cứng mà HĐH sẽ xem xét, xử lý

4. Nội dung:

a. Nội dung chi tiết: (công thức, định lý, hình vẽ)

I.1.3 Cấu trúc phần cứng của hệ thống máy tính (HTMT)

HĐH gắn bó chặt chẽ với phần cứng của hệ thống bởi nó quản lý cấp phát tài nguyên cho các ứng dụng và che phần cứng đối với NSD phát triển ứng dụng nên để thiết kế HĐH chúng ta cần hiểu rõ, nắm chắc phần cứng. Bởi vậy, ngoài các kiến thức/tài liệu yêu cầu tối thiểu dưới đây, cần tìm xem lại môn học “cấu trúc máy tính” và “Kỹ thuật vi xử lý”.

Kiến trúc của HTMT:

- CPU
- Bộ nhớ
- Ngoại vi/Điều khiển ngoại vi

Hoạt động của hệ thống máy tính

- Khởi động
- Xử lý các lệnh tuần tự
- Xử lý sự kiện – ngắt và xử lý ngắt

Cấu trúc vào/ra (I/O)

- Ngắt (I/O interrupt)
- DMA (Direct memory Access)

Bộ nhớ và thiết bị lưu trữ

- Bộ nhớ cơ sở/bộ nhớ trong RAM và ROM
- Thiết bị lưu trữ/ Bộ nhớ ngoài

Bảo vệ phần cứng

b. Nội dung thảo luận

c. Nội dung tự học

Tìm đọc lại về kiến trúc, nguyên tắc hoạt động của máy tính và cấu trúc ổ đĩa cứng

d. Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)

- Tài liệu 1, Operating system concepts, chương 2, từ trang 27 đến 53
- Tài liệu 4, Computer Organization and Architecture (William Stallings) chương 3, mục 3.1 trang 49; chương 4, mục 4.1 trang 100; chương 5, mục 5.1 trang 155

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương I: TỔNG QUAN**

I.2. Các thành phần quản lý và Cấu trúc của HĐH

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 2 tiết, tự học 3 tiết

3. Mục đích, yêu cầu:

Nắm được HĐH dưới các góc nhìn khác nhau bao gồm người sử dụng máy tính như công cụ làm việc, người phát triển ứng dụng và người thiết kế HĐH. Hiểu được các dịch vụ mà HĐH có thể cung cấp.

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

I.2.1. Các thành phần quản lý của HĐH

Các HĐH khác nhau, với các mục đích, yêu cầu khác nhau có thể có các thành phần khác nhau, tuy nhiên HĐH thường có các thành phần quản lý chính sau:

1. *Quản lý tiến trình (Process Management):*

Một chương trình khi nằm trên đĩa sẽ không thực hiện công việc gì, nó chỉ hoạt động khi được nạp vào bộ nhớ. Và khi đó mỗi một chương trình được xem như một tiến trình (Process).

Như vậy một tiến trình là một đơn vị công việc đối với hệ thống. Một hệ thống có thể chứa nhiều tiến trình, do đó, việc quản lý tiến trình cũng được xem như công việc hàng đầu quan trọng của HĐH. Quản lý tiến trình thực chất là việc cấp phát CPU cho các tiến trình.

(Lưu ý: một số tài liệu tiếng Việt sử dụng thuật ngữ “quá trình” thay vì tiến trình)

Các công việc liên quan tới quản lý tiến trình.

- Tạo và xoá tiến trình người sử dụng cũng như tiến trình hệ thống
- Dừng hoặc tiếp tục thực hiện tiến trình
- Liên lạc giữa các tiến trình
- Điều khiển các xung đột (deadlocks) có thể xảy ra giữa các tiến trình

2. *Quản lý Bộ nhớ trong (Memory Management):*

Bộ nhớ trong hay còn gọi là bộ nhớ chính là hữu hạn được cấp phát cho các tiến trình. Nguyên tắc quản lý bộ nhớ của HĐH khá phong phú, tuy nhiên tất cả đều phải đảm bảo các nhiệm vụ chung sau:

- Theo dõi và lưu vết xem các phần nào của bộ nhớ đang bị sử dụng và ai đang sử dụng
- Quyết định xem tiến trình nào sẽ được nạp vào bộ nhớ khi bộ nhớ rỗi
- Cấp phát và thu hồi bộ nhớ khi cần thiết.

3. *Quản lý vào/ra (Input/output System Management):*

Một trong những nhiệm vụ chính của HĐH là điều khiển các thiết bị nhập xuất thông qua các ngắt nhập xuất. Ngắt thông báo cho CPU biết trạng thái của một kênh hay một thiết bị. HĐH che sự phức tạp và các đặc trưng riêng biệt của các thiết bị I/O trên từng hệ thống tạo ra sự dễ dàng sử dụng nhất với người sử dụng bao gồm NSD phát triển ứng dụng.

Để thực hiện được điều này HĐH phải tồn tại một bộ phận điều khiển thiết bị, bộ phận này phối hợp cùng với phần cứng quản lý hoạt động và trao đổi thông tin giữa hệ thống, chương trình NSD và NSD với các thiết bị xuất nhập. Trong Unix sự khác biệt của các thiết bị xuất nhập được che bởi chính một tiêu hệ thống I/O của HĐH, nó bao gồm:

- Phân quản lý vùng nhớ đệm (buffering), vùng lưu giữ (caching) và vùng chứa (spooling).
- Giao diện trình điều khiển thiết bị chung

- Các trình điều khiển cho các thiết bị cụ thể
- Bộ phận điều khiển thiết bị thực hiện các nhiệm vụ sau:
- Gửi mã lệnh điều khiển đến thiết bị
 - Tiếp nhận yêu cầu ngắt từ các thiết bị
 - Phát hiện và xử lý lỗi

4. *Quản lý bộ nhớ ngoài (Secondary-Storage Management):*

Bộ nhớ ngoài hay thiết bị lưu trữ, bộ nhớ mở rộng bao gồm Ổ đĩa từ, đĩa quang, quang từ, băng từ hay các loại thẻ nhớ. Loại bộ nhớ mở rộng quan trọng nhất, thường chứa HĐH chính là ổ đĩa cứng.

Thành phần quản lý bộ đĩa cứng thực hiện những nhiệm vụ chính sau:

- Quản lý không gian trống trên đĩa
- Cấp phát không gian đĩa
- Định thời hay lập lịch cho đầu từ của đĩa

5. *Quản lý Tập tin (File Management)*

Thành phần này tổ chức và quản lý tập tin cũng như thư mục của các tập tin.

Trong hệ thống, các tập tin có thể có nhiều tiến trình cùng truy xuất, HĐH cần có cơ chế thích hợp để bảo vệ tập tin tránh việc ghi/đọc tập tin bất hợp lệ.

Thành phần quản lý tập tin thực hiện các nhiệm vụ sau:

- Tạo và xoá tập/thư mục
- Cung cấp thao tác và bảo vệ tập tin/ thư mục
- Hỗ trợ các hàm thao tác với tập/ thư mục, ánh xạ tập tới các ổ đĩa để tạo một môi trường làm việc dễ dàng trong suốt đối với người sử dụng
- Sao lưu tập lên các vị trí lưu trữ tin cậy

6. *Các thành phần khác*

Các thành phần sẽ được nghiên cứu kỹ hơn trong nội dung các môn học khác hoặc học viên tự nghiên cứu nâng cao, bao gồm:

- Mạng (Networking)
- Hệ Thống dịch lệnh (Command-Interpreter System)
- Hệ Bảo vệ (Protection System)

1.2.2. Các dịch vụ cung cấp bởi HĐH (Operating-System Services)

HĐH cung cấp môi trường thực thi các chương trình; nó mang đến các dịch vụ xác định cho các ứng dụng và người dùng các chương trình đó. Các dòng sản phẩm HĐH khác nhau, các phiên bản HĐH khác nhau có thể cung cấp các dịch vụ khác nhau, tuy nhiên có thể liệt chúng vào các nhóm dịch vụ sau:

- Thực thi chương trình (Program execution)
- Thao tác xuất nhập (I/O Operation)
- Thao tác hệ thống tập tin (File-System Manipulation)
- Truyền thông/ Trao đổi thông tin (Communication)
- Phát hiện lỗi (Error Detection)

HĐH không những cung cấp các dịch vụ hỗ trợ NSD, mà còn chứa một loạt các chức năng khác nhằm đảm bảo cho chính hệ thống hoạt động một cách hiệu quả, chẳng hạn như:

- Phân phối/cấp phát tài nguyên (Resource Allocation)
- Kế toán/quản lý tài nguyên đối với từng NSD (Accounting)
- Bảo vệ (Protection)

1.2.3. Lời gọi Hệ thống (System Calls)

System calls cung cấp giao diện giữa một tiến trình và HĐH. Các lời gọi hệ thống này thường được cung cấp dưới dạng tập các chỉ lệnh hợp ngữ assembly-language bởi các tài liệu kỹ thuật đi kèm giành cho NSD là lập trình viên trên ngôn ngữ bậc thấp. Cũng có nhiều hệ thống cho phép lời gọi hệ thống được thực hiện trực tiếp từ chương trình ngôn ngữ bậc cao hơn như ngôn ngữ lập trình như: C, C++ hay Perl.

Tập hợp các lời gọi hệ thống sẵn dùng được biết đến với tên gọi là Interface ứng dụng của HĐH (API Application Program Interface), chúng có thể chia vào 5 nhóm:

- Điều khiển Tiến trình (Process Control)
- Quản lý Tập tin (File Management)
- Quản lý Thiết bị (Device Management)
- Bảo trì thông tin (Information Maintenance)
- Truyền thông (Communication)

1.2.4. Cấu trúc Hệ thống (System Structure)

- Cấu trúc đơn giản (Simple Structure) và Hệ thống đơn khối (Monolithic System)
- Hệ thống/ Cấu trúc phân tầng (Layered System)
- Vi nhân (Microkernels)
- Máy ảo (Virtual Machine)
- Mô hình khách-chủ (Client-Server Model)

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)
 - Tài liệu 1: Operating system concepts, chương 3, trang 55 đến 92
 - Tài liệu 2: Chương 1 mục 1.3 và 1.4 trang 15 đến 43
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương II: QUẢN LÝ TIỀN TRÌNH**

II.1. Tiến trình

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 3 tiết, tự học 3 tiết

3. Mục đích, yêu cầu: Hiểu rõ khái niệm Tiến trình, phân biệt được chương trình, tiến trình; Nắm được các trạng thái của tiến trình, các mô hình tiến trình và nguyên tắc quản lý tiến trình.

Học xong phần này có thể giải thích được việc hệ thống chuyển từ việc thực hiện tiến trình này sang tiến trình khác như thế nào và tại sao

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

II.1.1. Khái niệm, định nghĩa tiến trình (Process).

Tiến trình là một chương trình hoặc một phần của chương trình đã được nạp vào bộ nhớ chính, đang trong quá trình thực hiện (nhưng không nhất thiết là luôn luôn được cấp CPU).

Chương trình là thực thể thụ động, tiến trình là thực thể chủ động của chương trình và có thời gian sống nhất định (life cycle).

Trong một số tài liệu tiếng Anh có sử dụng khái niệm job hay task: công việc hay chức năng, có thể hiểu là tiến trình giống như process, và một số tài liệu tiếng Việt dịch process là quá trình.

II.1.2. Các trạng thái của tiến trình

- Mới (new): tiến trình mới vừa được khởi tạo
- Sẵn sàng (ready): Tiến trình đang đợi trong hàng đợi (ready queue) sẵn sàng chiếm dụng CPU để tiếp tục thi hành
- Thực thi (running): Tiến trình đang được CPU thực thi các câu lệnh (tiến trình đang chiếm dụng CPU)
- Bị chặn-chờ (blocked - waiting): Tiến trình đang đợi một sự kiện nào đó trước khi tiếp tục thi hành. Sự kiện đó có thể là việc xuất/nhập dữ liệu hoặc tín hiệu đồng bộ từ một tiến trình khác.
- Kết thúc (terminated-exited): tiến trình kết thúc.

Một tiến trình có thể chuyển từ trạng thái này sang trạng thái khác chủ động hoặc bị động

II.1.3. Khối điều khiển tiến trình PCB (Process Control Block)

Mỗi một tiến trình được quản lý trong HĐH bởi một PCB.

Khối điều khiển tiến trình là một cấu trúc dữ liệu gồm các thông tin quan trọng sau:

1. Danh định cho tiến trình (PID)
2. Trạng thái của tiến trình
3. Bộ đếm chương trình – chứa địa chỉ của lệnh tiếp theo
4. Vùng lưu giá trị thanh ghi CPU
5. Độ ưu tiên của tiến trình
6. Thông tin định vị bộ nhớ tiến trình
7. Thông tin bảo mật
8. Con trỏ đến các tiến trình cha, con
9. Bộ vi xử lý mà tiến trình đang sử dụng (trong các hệ đa bộ xử lý)
10. Danh sách các tài nguyên mà tiến trình đang sử dụng

II.1.4. Mô hình tiến trình, các loại tiến trình (Process Models)

Các hệ thống khác nhau có thể tổ chức để các tiến trình có thể thực thi tuần tự hoặc song hành.

Ở hệ thống các tiến trình song hành có thể chia làm nhiều dạng (mô hình) khác nhau:

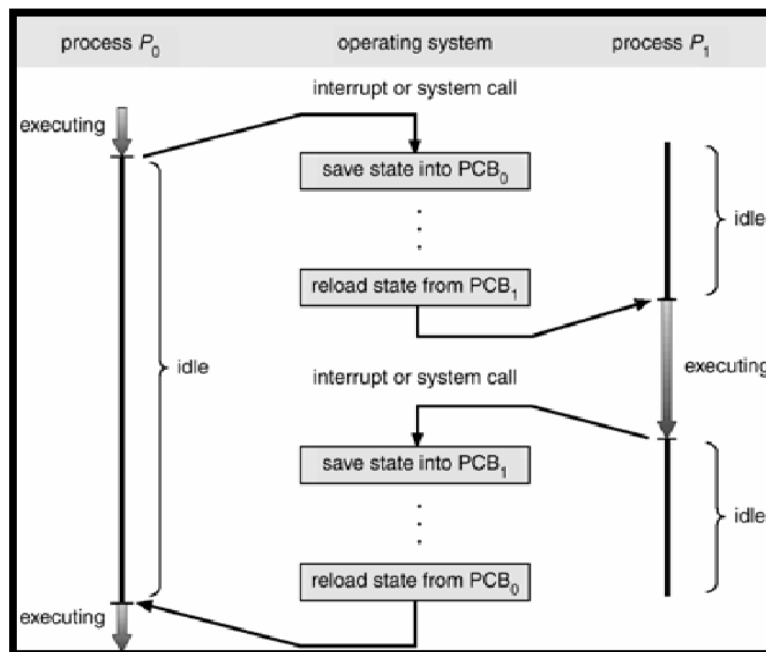
- Tiến trình song hành đồng mức
- Tiến trình song hành độc lập
- Tiến trình song hành có quan hệ thông tin
- Tiến trình song hành phân cấp

Như vậy ở góc nhìn nào đó như liên quan tới lịch sử phát triển HĐH, có thể hình dung có các mô hình tiến trình như sau:

- Mô hình theo dõi giản đơn (Simple Monitor System Models)
- Mô hình xử lý theo bó giản đơn (Batch System Models)
- Mô hình xử lý theo bó thông minh (Batch System with Block Models)
- Mô hình phân chia thời gian (Time Sharing Interactive Models)

II.1.5. Chuyển CPU giữa các tiến trình (Context Switch)

Giả sử có 2 tiến trình P_0 , P_1 . P_0 đang chạy thì xuất hiện ngắt và P_0 dừng lại, khi đó toàn bộ thông tin về P_0 từ các thanh ghi được ghi ra thực thể PCB₀ trong bộ nhớ chính. Sau đó các thông tin của P_1 từ PCB₁ trong bộ nhớ chính được nạp vào các thanh ghi và thực thi P_1

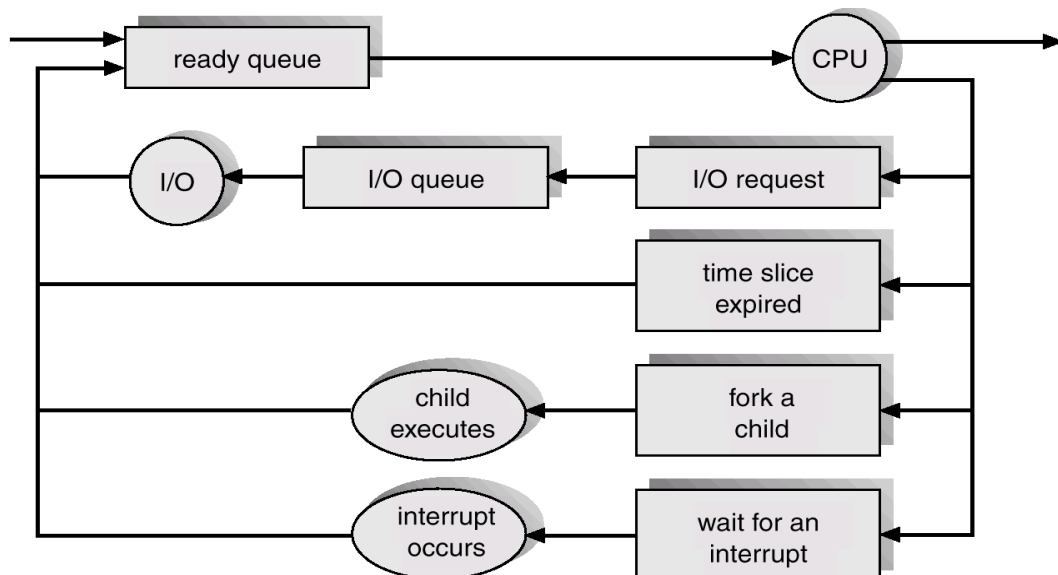


II.1.6. Lập thời biểu tiến trình (Process Scheduling)

Trong hệ thống có nhiều tiến trình song hành, các tiến trình chuyển từ trạng thái này sang trạng thái khác và CPU chuyển từ việc cấp cho tiến trình này sang tiến trình khác. CPU tại một thời điểm cụ thể chỉ có thể thực hiện một chỉ lệnh của một tiến trình, vì vậy các tiến trình khác sẽ phải chờ trong các hàng đợi khác nhau.

1. Hàng đợi lập thời biểu (Scheduling Queues)

Các tiến trình đang chờ được cấp CPU để được thực hiện được xếp vào hàng đợi sẵn sàng. Các tiến trình khác có thể đang trong quá trình chờ kết thúc hoặc ngoại vi sẵn sàng sẽ nằm trong các hàng đợi tương ứng.



2. Bộ định thời biểu (Schedulers)

Việc xếp các tiến trình vào vị trí thích hợp trong các hàng đợi khác nhau sẽ do các bộ định thời thực hiện. Các bộ định thời bao gồm:

- Bộ định thời dài (Long-term Scheduler)
- Bộ định thời ngắn (Short-term Scheduler)
- Bộ định thời trung kỳ (Middle-term Scheduler)

Bộ định thời ngắn quyết định chọn tiến trình nào trong hàng đợi sẵn sàng để thực thi tức là cấp CPU cho nó, và vì vậy cũng được gọi là bộ định thời CPU hay lập lịch cho CPU (CPU Scheduling) sẽ được nghiên cứu kỹ trong bài sau.

b) Nội dung thảo luận

c) Nội dung tự học

d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)

- Tài liệu 1: Chương 4 từ trang 95 đến trang 107
- Tài liệu 2: Chương 2 mục 2.1 từ trang 48 đến trang 52

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương II QUẢN LÝ TIẾN TRÌNH**

II.2. Luồng (Thread)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết, tự học 1 tiết

Mục đích, yêu cầu: Học xong phần này nắm được khái niệm về luồng, phân biệt được chương trình, tiến trình và luồng. Có thể giải thích được việc hệ thống chuyển từ việc thực hiện tiến trình này sang tiến trình khác như thế nào và tại sao.

3. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

II.2.1 Khái niệm

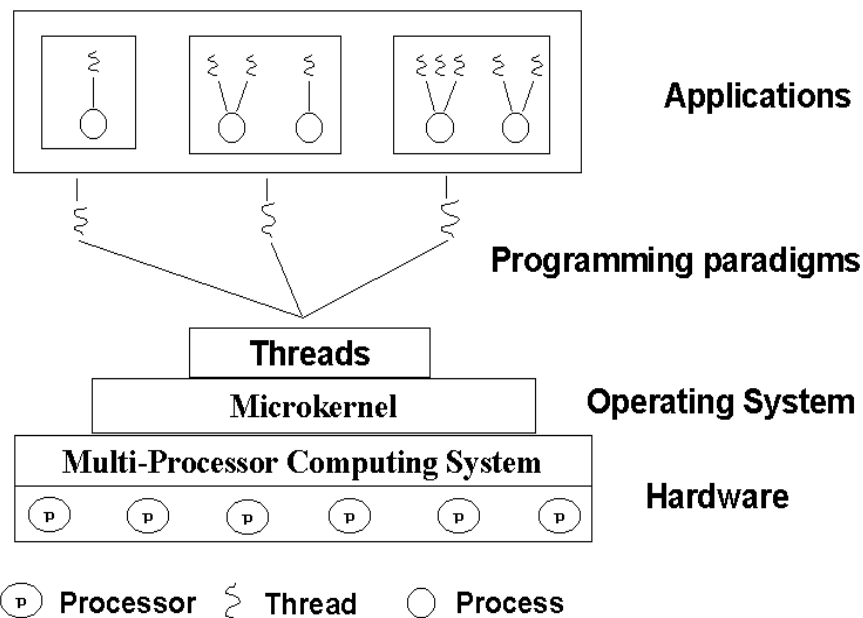
Luồng hay còn được sử dụng với khái niệm tiểu trình hay thiết trình.

Luồng được xem như một phiên bản đơn giản của tiến trình mà đã được bỏ đi nhiều đặc tính. Truyền thống, một tiến trình sẽ có một luồng thực hiện công việc. Nhưng một tiến trình có thể chứa đựng nhiều luồng.

Một tiến trình bao gồm: thread id, program counter, tập thanh ghi register, và một ngăn xếp stack. Luồng sẽ chia sẻ với các luồng trong cùng một tiến trình phần mã (code section), phần dữ liệu (data section), và các tài nguyên của HĐH mà được dùng bởi tiến trình.

II.2.2. Lợi ích của việc sử dụng luồng

- Khả năng đáp ứng nhanh: Các phần của ứng dụng có thể phức tạp với user độc lập với nhau. Một luồng của Web browser có thể web, luồng khác có thể download tệp
- Chia sẻ tài nguyên: Các luồng có thể chia sẻ không gian bộ nhớ và tài nguyên của tiến trình.
- Kinh tế: Việc cấp phát bộ nhớ và tài nguyên cho tiến trình là một việc tốn thời gian, nhưng việc này có thể giảm đi rất nhiều nếu áp dụng luồng.
- Tận dụng được ưu thế của các kiến trúc đa bộ vi xử lý.



Tiến trình đa luồng (multithreaded process) là tiến trình thực thi trong nhiều luồng. Các thread trong cùng một process chia sẻ code section, data section và tài nguyên khác (các file đang mở,...) của process.

II.2.3. Các mô hình đa luồng:

- Mô hình nhiều – một (Many to One)
- Mô hình một – một (One to One)
- Mô hình nhiều – nhiều (Many to Many)

b) Nội dung thảo luận

c) Nội dung tự học

Tự đọc về giao tiếp liên tiến trình (Interprocesses Communication)

d) Bài tập (bắt buộc, mở rộng)

4. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)
 - Tài liệu 1, chương 5 trang 129 đến 149
 - Tài liệu 2: chương 2 mục 2.1.3 trang 53
5. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương II QUẢN LÝ TIỀN TRÌNH**

II.3. Lập lịch cho CPU (CPU Scheduling)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 2 tiết, tự học 3 tiết

3. Mục đích, yêu cầu:

Nắm được các khái niệm liên quan tới lập lịch bao gồm lập lịch cho CPU, nắm sâu hơn vấn đề song hành của các tiến trình và việc thực hiện chúng.

Hiểu và vận dụng được các thuật toán lập lịch cho CPU, hướng tới viết được các modul chương trình quản lý cấp phát CPU

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

Bộ lập lịch cho CPU: Bất cứ khi nào CPU rỗi, HĐH phải chọn một tiến trình đang có trong hàng đợi sẵn sàng để thực hiện, đây là hệ thống đa nhiệm. Việc chọn này được thực hiện bởi một lập lịch gọi là lập lịch ngắn hạn (Short-term Scheduler).

II.3.1. Các khái niệm cơ bản (Basic Concepts)

- Chu kỳ CPU-I/O (CPU-I/O Burst Cycle)

Thông thường một tiến trình trong quá trình được thực hiện có thời gian làm việc mà thực chủ yếu là chờ I/O (do I/O thường rất chậm so với việc CPU thực thi các chỉ lệnh).

- Bộ lập lịch CPU

Bộ định thời hay bộ lập lịch cho CPU là chức năng cơ bản của bất cứ HĐH nào, đảm nhiệm việc chọn tiến trình để cấp CPU cho nó. Quyết định lập lịch CPU có thể xảy ra dưới điều kiện 1 trong 4 trường hợp sau:

1. Tiến trình chuyển từ Running sang trạng thái chờ
2. Tiến trình chuyển từ Running sang trạng thái sẵn sàng
3. Tiến trình chuyển từ trạng thái chờ sang sẵn sàng
4. Tiến trình kết thúc.

Trường hợp 1 và 4 tiến trình đang giữa CPU trả CPU cho hệ thống một cách tự nguyện vì bản thân nó không cần CPU lúc đó.

- Định thời không trung dụng, không bắt buộc tức là tự nguyện (NonePre-emptive)

Tiến trình giữa CPU sẽ chỉ trả CPU khi nó kết thúc hoặc chuyển sang trạng thái chờ khi làm việc với I/O chẳng hạn. Cũng có thể sử dụng thuật ngữ không ưu tiên trước như một số tài liệu.

- Định thời trung dụng hay định thời ưu tiên trước: Lập lịch ưu tiên trước (Pre-emptive) ưu tiên cho những tiến trình có CPU burst ngắn hơn

- Bộ cấp phát (Dispatcher)

II.3.2. Các tiêu chí lập lịch

Đánh giá bộ lập lịch dựa trên các tiêu chí trung dưới đây; với các hệ thống và yêu cầu khác nhau các tiêu chí có thể đóng vai trò quan trọng khác nhau:

- Việc sử dụng CPU
- Thời lượng
- Thời gian hoàn thành
- Thời gian chờ
- Thời gian đáp ứng/ thời gian phức tạp

II.2.3. Các thuật toán định thời (Scheduling Algorithms)

Thuật toán định thời áp dụng để chọn tiến trình đang trong hàng đợi sẵn sàng để cấp CPU

1. *Tiến trình đến trước phục vụ trước: Thuật toán FCFS (First Come First Serve)*

Đây là giải thuật đơn giản nhất khi mà tiến trình nào vào hàng đợi sẵn sàng trước sẽ được lựa chọn phục vụ trước tức là được chọn để cấp CPU.

Thuật toán này sẽ cấp CPU cho tiến trình cho đến khi CPU được giải phóng, như vậy FCFS là định thời không trung dụng hay định thời tự nguyện.

2. *Chọn tiến trình yêu cầu CPU ngắn nhất: Giải thuật SJF (Shortest Job First)*

Giải thuật này lựa chọn tiến trình có yêu cầu CPU ít nhất trong số các tiến trình đang xếp ở hàng đợi sẵn sàng. Khi tiến trình đã được cấp CPU nó sẽ chiếm giữ CPU đến khi nó tự nguyện nhường trả CPU nên cũng được gọi là SJF không trung dụng hay không ưu tiên

3. *Thuật toán SJF trung dụng:*

Tương tự như SJF nhưng trong khi tiến trình đang giữa CPU có tiến trình mới sẵn sàng nhưng chỉ yêu cầu CPU ngắn hơn thời lượng còn cần CPU của tiến trình hiện tại thì CPU cũng sẽ được HĐH giành lại và cấp cho tiến trình mới. Thuật toán này cũng được gọi là Pre-emptive SJF hay SJF trung dụng hay SRTF (Shortest Remaining Time First). Cũng có cách gọi khác là SJF ưu tiên.

4. *Định thời luân phiên Robin (Round Robin Scheduling)* hay còn gọi là phân chia thời gian viết tắt là RR. Các tiến trình sẽ được cấp CPU lần lượt, mỗi tiến trình được cấp một khoảng thời lượng CPU tối đa xác định.

Trong thực tế các HĐH có thể kết hợp, lồng ghép các thuật toán trên hay các giải thuật phát triển từ các thuật toán trên như định thời với hàng đợi nhiều cấp, định thời hàng đợi phản hồi đa cấp.

Đánh giá các giải thuật

- FCFS
- SJF
- SRTF
- RR

b) Nội dung thảo luận

c) Nội dung tự học

Tự đọc nâng cao phần định thời đa xử lý (đa lõi và đa chip) và Lập lịch thời gian thực.

d) Bài tập (bắt buộc, mở rộng) 3 tiết trên lớp

Câu 1: Có các tiến trình P0, P1, P2, P3, P4 và P5 đã đang sẵn sàng chờ được cấp phát CPU. Đánh giá các thuật toán lựa chọn tiến trình để cấp phát (thuật toán lập lịch cho CPU). Các tiến trình có thời gian cần CPU (CPU burst time) và có thời gian đến (vào hàng đợi Ready) được cho trong bảng dưới đây:

Tiến trình	Thời gian đến (ms)	CPU burst Time (ms)
P0	0	15
P1	5	15
P2	8	2
P3	10	15
P4	25	35
P5	29	5

Đối với mô hình Time-sharing, Quantum time hay Time slice = 10ms

Câu 2: Có các tiến trình P0, P1, P2, P3, P4 và P5 đã đang sẵn sàng chờ được cấp phát CPU. Đánh giá các thuật toán lựa chọn tiến trình để cấp phát (thuật toán lập lịch cho CPU). Các tiến trình có thời gian cần CPU (CPU burst time) và có thời gian đến (vào hàng đợi Ready) được cho trong bảng dưới đây:

Tiến trình	Thời gian đến (ms)	CPU burst Time (ms)
P0	0	15
P1	2	15
P2	7	6
P3	10	13
P4	15	2
P5	29	5

Đối với mô hình Time-sharing, Quantum time hay Time slice = 10ms

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)
 - Tài liệu 1: Chương 6, từ trang 151 đến trang 187
 - Tài liệu 2: Chương 2, mục 2.4 từ trang 82 đến trang 93
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương II: QUẢN LÝ TIỀN TRÌNH**

II.4. Đồng bộ hóa tiến trình (Process Synchronization)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết

3. Mục đích, yêu cầu:

Nắm được khái niệm đoạn găng, cơ chế hoạt động của hiệu báo Semaphore và Monitor.

Hiểu được các biện pháp điều độ tiến trình qua đoạn găng, đồng bộ hóa các tiến trình.

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

II.4.1 Đoạn găng

Mỗi một tiến trình có một đoạn mã, gọi là vùng hạn chế (Critical section), trong đó tiến trình sẽ có thể thay đổi các biến dữ liệu chung. Bài toán đặt ra là khi một tiến trình thực hiện trong đoạn hạn chế này, thì không có tiến trình nào khác thực hiện trong đoạn mã hạn chế của nó.

HĐH phải đảm bảo là tại một thời điểm chỉ có một tiến trình nằm trong đoạn găng, nếu có nhiều tiến trình cùng muốn vào (thực hiện) đoạn găng thì các tiến trình khác phải chờ tiến trình thực tại thoát khỏi đoạn găng.

Một giải pháp cho bài toán Critical section phải thỏa mãn các điều kiện dưới đây:

1. Loại trừ lẫn nhau (Mutual exclusion): Nếu tiến trình P_i đang thực hiện trong đoạn hạn chế, thì không có tiến trình nào có thể thực hiện trong đoạn hạn chế.
2. Tiến triển (progress): Nếu không có tiến trình nào đang thực hiện trong đoạn hạn chế và có một số tiến trình đang muốn thực hiện đoạn hạn chế, thì chỉ có những tiến trình nào mà hiện tại không thực hiện phần mã còn lại có thể được xem xét.
3. Giới hạn chờ đợi (bounded waiting): tồn tại một giới hạn về thời gian các tiến trình khác được cho phép thực hiện đoạn hạn chế của chúng sau khi một tiến trình yêu cầu được thực hiện đoạn hạn chế và trước khi được chấp nhận.

II.4.2 Các giải pháp điều độ tiến trình qua đoạn găng

- Giải pháp 2 tiến trình (Two-processes Solution)
- Giải pháp phần cứng (Synchronization Hardware)
- Giải pháp dùng biến khóa
- Giải pháp được hỗ trợ bởi HĐH và ngôn ngữ lập trình bao gồm:
 1. Giải thuật Bakery
 2. Giải pháp dùng đèn báo Semaphore
 3. Giải pháp dùng monitor
 4. Giải pháp trao đổi thông điệp

b) Nội dung thảo luận

c) Nội dung tự học

d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)

- Tài liệu 1, chương 7 từ trang 189 đến trang 241.

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương III. QUẢN LÝ BỘ NHỚ TRONG**

III.1. Giới thiệu chung

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 2 tiết, tự học 3 tiết

3. Mục đích, yêu cầu:

Nắm lại được một số khái niệm căn bản

Hiểu nguyên tắc cấp phát tổ chức quản lý khác nhau bộ nhớ trong,

Hiểu cách tiếp cận quản lý bộ phân trang, phân đoạn và các thuật toán cấp phát bộ nhớ liên tục cho các tiến trình

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

Quản lý bộ nhớ là một trong những nhiệm vụ quan trọng và phức tạp nhất của HĐH. Thành phần Quản lý bộ nhớ phải quản lý hiệu quả không gian hữu hạn bộ nhớ trong, đồng thời cấp phát một cách hữu hiệu cho các tiến trình nhất là mô hình đa tiến trình mà vẫn đảm bảo sự hoạt động chính xác của từng tiến trình.

III.1.1 Các khái niệm cơ bản

- Sự tái định vị (Relocation)
- Liên kết địa chỉ (Address Binding)

1. Thời gian biên dịch

2. Thời điểm nạp

3. Thời gian thực thi

- Không gian địa chỉ logic (Logic Address Space)
- Không gian địa chỉ vật lý (Physical Address Space)
- Nạp động (Dynamic Loading)
- Liên kết động (Dynamic Linking)
- Bộ nhớ chia sẻ và thư viện chia sẻ
- Phủ lấp (Overlay)

III.1.2. Hoán chuyển (Swapping)

Phủ lấp cho phép nạp một tiến trình lớn hơn dung lượng bộ nhớ cấp phát cho nó bởi một phần của tiến trình nằm ở bộ nhớ ngoài, hoán chuyển thì cho phép chuyển một tiến trình đã được cấp phát bộ nhớ xếp tạm ra bộ nhớ ngoài để giải phóng bộ nhớ nạp tiến trình khác. HĐH sẽ dùng một vùng nhớ ngoài nhanh nhất thường là đĩa cứng và gọi là Backing Store.

Tiến trình hoán chuyển ra là tiến trình đang không được thực thi.

- Swap in
- Swap out

Khái niệm hoán chuyển (swap) cũng có thể được áp dụng nội trong một tiến trình khi khi sẽ swap một phần của tiến trình, nhưng có khác với overlay (phần này sẽ nghiên cứu trong bài Bộ nhớ ảo).

III.1.3. Cấp phát bộ nhớ (Memory Allocation)

Bộ nhớ trong thường được chia thành 2 vùng riêng biệt, một cho HĐH và một cho các tiến trình.

- Cấp phát bộ nhớ liên tục (Contiguous Allocation)
Mỗi tiến trình chỉ được cấp một vùng nhớ và tuần tự. Bộ nhớ vì vậy được chia thành các vùng.
Vùng nhớ được cấp phát có thể bằng nhưng cũng có thể lớn hơn so với nhu cầu của tiến trình.

Phân vùng cố định hoặc phân vùng động.

- Cấp phát bộ nhớ không liên tục (None-contiguous Allocation)

Mỗi tiến trình sẽ được cấp nhiều hơn một vùng nhớ và không liên nhau.

Bộ nhớ trong được chia thành các vùng có độ lớn bằng nhau (gọi là trang mà thực chất là khung trang), cũng có thể thành các đoạn có độ lớn khác nhau.

Tiến trình được cấp một số các khung trang và/hoặc các đoạn đó.

III.1.4. Phân mảnh (Fragmentation)

Hiện tượng phân mảnh là có không gian nhớ rảnh chưa cấp phát cho tiến trình nào nhưng không thể cấp cho tiến trình mặc dù tổng dung lượng không gian nhớ rảnh này lớn hơn nhu cầu cần cấp phát cho tiến trình.

- Phân mảnh nội (Internal fragmentation)
Là hiện tượng kích thước vùng nhớ được cấp phát hơi lớn hơn vùng nhớ yêu cầu.
- Phân mảnh ngoại /Phân mảnh ngoài (External fragmentation)
Là hiện tượng kích thước không gian bộ nhớ còn trống đủ để thỏa mãn một yêu cầu cấp phát, tuy nhiên không gian nhớ này không liên tục

b) Nội dung thảo luận

c) Nội dung tự học

Xem kỹ lại phần tổ chức bộ nhớ (phần cứng) mục 7.3 chương 7 trang 240 tài liệu 5.

d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)
 - Tài liệu 1, chương 9 từ trang 273 đến trang 316
 - Tài liệu 3, tham khảo chương 3 trang 135
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương III. QUẢN LÝ BỘ NHỚ TRONG**

III.2. Cấp phát bộ nhớ liên tục (Contiguous Allocation)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết, 2 tiết

3. Mục đích, yêu cầu:

Hiểu nguyên tắc tổ chức và cấp phát liên tục bộ nhớ trong,

Nắm được, hướng tới viết chương trình dựa trên các thuật toán cấp phát bộ nhớ liên tục cho các tiến trình

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

Cấp phát bộ nhớ liên tục (Contiguous Memory Allocation)

Cấp phát liên tục tức là cấp cho tiến trình vùng nhớ có địa chỉ liên nhau.

III.2.1. Phân vùng cố định (Fixed Partitioning)

Chia bộ nhớ chính thành nhiều phần không trùng lắp gọi là các phân hoạch có kích thước bằng nhau hoặc khác nhau.

Tiến trình nào có kích thước nhỏ hơn hoặc bằng kích thước của một phân hoạch thì có thể nạp vào phân hoạch đó.

Nếu chương trình có kích thước lớn hơn một phân hoạch thì phải dùng cơ chế phủ lấp (overlay).

Nếu tất cả các phân hoạch đều đã đầy thì hệ điều hành sẽ tiến hành hoán chuyển (swap) các tiến trình.

III.2.2. Phân vùng động (Dynamic Partitioning)

Số lượng phân hoạch (partition) không cố định và phân hoạch có kích thước khác nhau, độ lớn quyết định bởi dung lượng tiến trình.

Mỗi tiến trình được cấp phát chính xác dung lượng bộ nhớ cần thiết.

Chúng ta sẽ nghiên cứu kỹ hơn thuật toán cấp phát bộ nhớ ở mục phía sau.

III.2.3. Cấp phát bộ nhớ phân vùng động (Dynamic Partition Allocation)

Tiến trình được cấp phát không gian nhớ liên tục và đúng bằng dung lượng yêu cầu. Sau một khoảng thời gian các tiến trình của hệ thống được kích hoạt rồi kết thúc, bộ nhớ trong của máy sẽ xuất hiện các vùng nhớ rảnh (rỗng) bên cạnh các phân hoạch được chiếm dụng bởi các tiến trình khác nhau. Các khoảng trống chưa bị chiếm dụng đó gọi là hole, vấn đề đặt ra là sẽ cấp cho tiến trình mới vào hole nào trong số đó.

Hole dùng để cấp phát phải có dung lượng bằng hoặc lớn hơn nhu cầu của tiến trình.

- Đủ lớn đầu tiên (First Fit)
Chọn vùng nhớ trống đủ lớn đầu tiên kể từ đầu bộ nhớ dành cho ứng dụng
- Vừa nhất (Best Fit)
Chọn vùng nhớ trống nhỏ nhất nhưng đủ lớn
- Lớn nhất đủ lớn (Worst Fit)
Chọn vùng nhớ trống lớn nhất nhưng đủ lớn

Nếu không còn vùng nhớ trống đủ lớn?

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng) 2 tiết

Bài 1.

Máy tính sử dụng partition để quản lý bộ nhớ, và tại một thời điểm bộ nhớ vật lý của máy đang được sử dụng như sau: 1,5M rỗng, 0.5M P1, 0.5M rỗng, 1.2M P2, 1.6M rỗng, 1.0M P3, 1.2M rỗng, 1.8M P4, 1.7M rỗng, 2.6M P5, 3.5M rỗng.

Các tiến trình mới xuất hiện và kết thúc theo thứ tự sau: P6: 1.1M, P7: 1.5M, P4 kết thúc, P8: 1.7M, P9: 1.3M, P2 kết thúc, P10: 1.7M, P11: 1.2M, P1 kết thúc, P12: 0.9M, P6 kết thúc.

Hãy giải thích và vẽ sơ đồ bộ nhớ vật lý ứng với các thuật toán cấp phát bộ nhớ chính (First Fit, Best Fit và Worst Fit) sau khi tiến trình P6 kết thúc: First Fit, Best Fit, Worst Fit

Bài 2:

Máy tính sử dụng partition để quản lý bộ nhớ, và tại một thời điểm bộ nhớ vật lý của máy đang được sử dụng như sau: 1,1M rỗng, 0.5M P1, 0.8M rỗng, 1.2M P2, 0.6M rỗng, 1.0M P3, 1.2M rỗng, 1.1M P4, 1.1M rỗng, 1.2M P5, 2.5M rỗng.

Các tiến trình mới xuất hiện và kết thúc theo thứ tự sau: P6: 0.6M, P7: 1.5M, P4 kết thúc, P8: 0.7M, P9: 1.3M, P2 kết thúc, P10: 1.1M, P11: 0.5M, P1 kết thúc, P12: 0.9M, P6 kết thúc.

Hãy giải thích và vẽ sơ đồ bộ nhớ vật lý ứng với các thuật toán cấp phát bộ nhớ chính (First Fit, Best Fit và Worst Fit) sau khi tiến trình P6 kết thúc: First Fit, Best Fit, Worst Fit

5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)
 - Tài liệu 1, chương 9 từ trang 273 đến trang 316
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương III. QUẢN LÝ BỘ NHỚ TRONG**

III.3. Cấp phát bộ nhớ không liên tục (None-contiguous Allocation)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 1 tiết, 2 tiết

3. Mục đích, yêu cầu:

Hiểu nguyên tắc tổ chức bộ nhớ bao gồm phân trang, phân đoạn và không gian địa chỉ từ nhớ

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

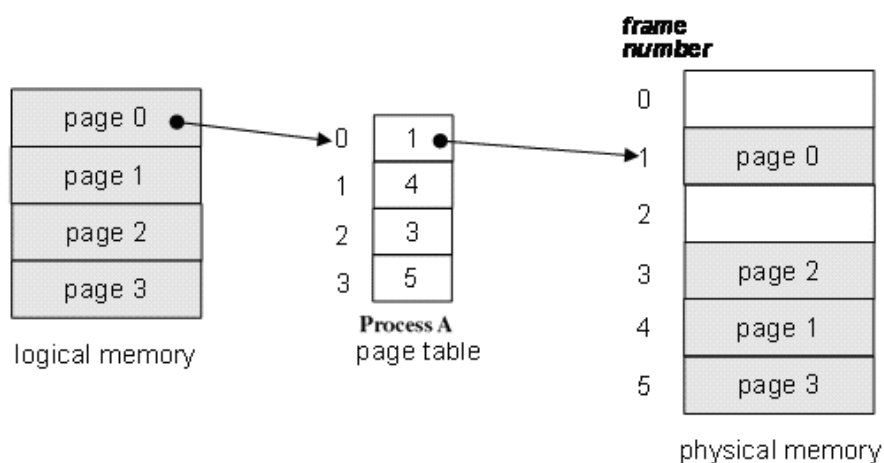
Ngược với cấp phát liên tục, các tiến trình được cấp nhiều hơn một vùng nhớ, và chúng không liên nhau.

III.3.1. Phân trang (Paging)

Kỹ thuật phân trang sẽ cho phép không gian địa chỉ vật lý cấp cho một tiến trình không cần phải là liên tục.

- Bộ nhớ thực được chia thành các khối kích thước cố định bằng nhau gọi là frame. Tiến trình được chia thành các phần bằng nhau gọi là các trang (page).
- Hệ điều hành phải thiết lập một bảng phân trang (page table) để ánh xạ địa logic thành địa chỉ thực.
- Mỗi tiến trình có một bảng phân trang (các trang bộ nhớ được cấp phát cho tiến trình này) được quản lý bằng một con trỏ và được lưu giữ trong PCB. Công việc nạp bảng phân trang vào hệ thống (do bộ điều phối CPU dispatcher thực hiện) là một phần của công việc hoán đổi tiến trình.
- Cơ chế phân trang khiến bộ nhớ bị phân mảnh nội, tuy nhiên lại khắc phục được phân mảnh ngoại

Cơ chế phân trang



- Chia sẻ trang dùng chung (Shared Page)

Có những đoạn mã lệnh thuộc một số trang nào đó có thể dùng chung cho một số các tiến trình (là các tiến trình chứa các trang mã lệnh giống nhau này).

III.3.2. Phân đoạn (Segmentation)

Trong thực tế, dưới góc nhìn của người sử dụng, một chương trình cấu thành từ nhiều phân đoạn (segment). Mỗi phân đoạn là một đơn vị logic như: main program, procedure, function, local variables, global variables, stack, symbol table, ...

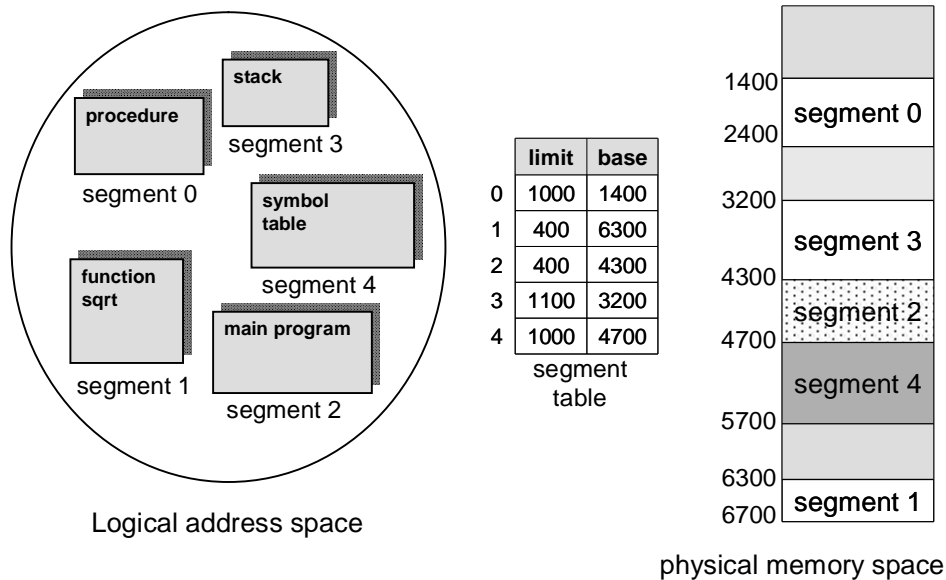
Kỹ thuật phân đoạn:

Chia không gian địa chỉ ảo thành một tập các phân đoạn (segment), mỗi phân đoạn có tên và kích thước riêng. Một địa chỉ logic được định vị bằng tên phân đoạn và độ dời (offset) bên trong phân đoạn đó. Khác với phân trang khi mà độ lớn trang là bằng nhau thì độ lớn mỗi đoạn không bắt buộc điều này.

Địa chỉ logic là một cặp giá trị: <segment-number, offset>

base – chứa địa chỉ khởi đầu của phân đoạn trong bộ nhớ

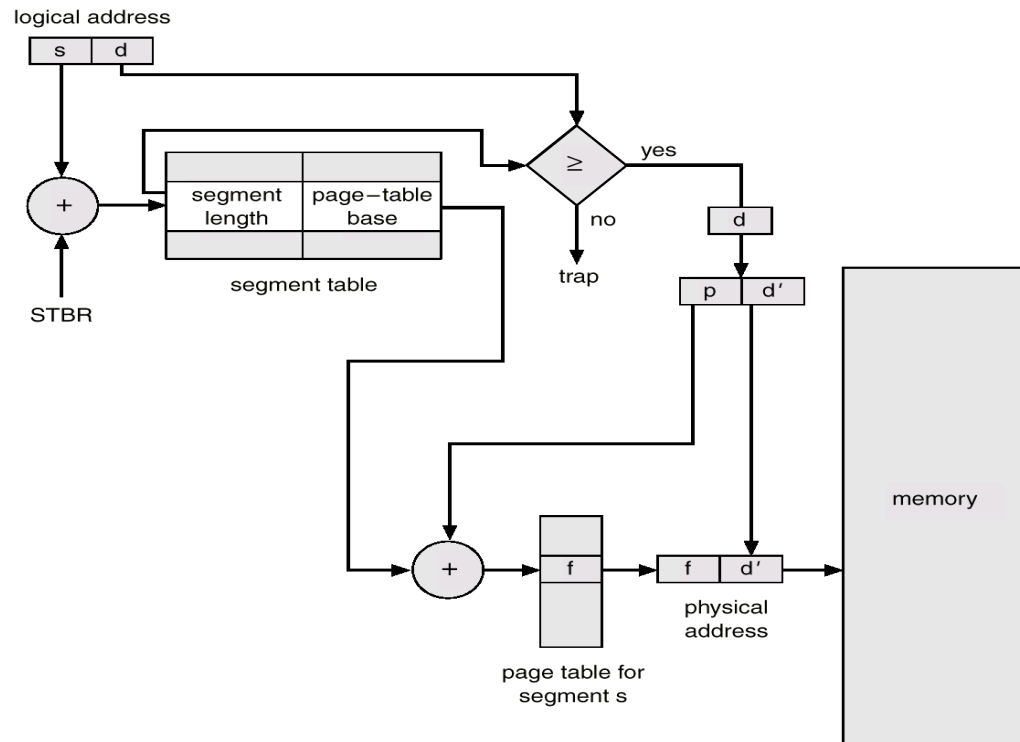
limit – xác định kích thước của phân đoạn



- Chia sẻ đoạn dùng chung (Shared Segment)
Tương tự như chia sẻ trang dùng chung, các tiến trình có thể có chung đoạn mã lệnh và đoạn mã code trong bộ nhớ này có thể chia sẻ.

III.3.3. Phân đoạn kết hợp với phân trang.

Phân trang và phân đoạn có điểm chung và điểm riêng; phân trang và phân đoạn có ưu và nhược điểm khác nhau. Có thể kết hợp đồng thời phân trang với phân đoạn khi mà một đoạn có thể được phân thành các trang.



- b) Nội dung thảo luận
 - c) Nội dung tự học
 - d) Bài tập (bắt buộc, mở rộng)
5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)
 - Tài liệu 1, chương 9 từ trang 273 đến trang 316
 - Tài liệu 3, tham khảo chương 3 trang 135
 6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương III QUẢN LÝ BỘ NHỚ**

III.4. Bộ nhớ ảo

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 2 tiết, 3 tiết tự học

3. Mục đích, yêu cầu:

4. Hiểu được khái niệm phân trang theo yêu cầu.

Hiểu được kỹ thuật bộ nhớ ảo, mục đích và ý nghĩa việc ứng dụng kỹ thuật bộ nhớ ảo;
Đồng thời nắm được các thuật toán xử lý lỗi trang, hướng tới lập trình mô phỏng các thuật toán này.

5. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

III.4.1. Khái niệm chung (Background):

- Bộ nhớ ảo là một kỹ thuật mà cho phép việc thực thi các tiến trình mà có thể không được nạp hoàn toàn vào bộ nhớ. Bộ nhớ ảo sẽ sử dụng một phần dung lượng bộ nhớ ngoài để phục vụ cho việc “mở rộng” bộ nhớ trong, và phần bộ nhớ ngoài này sẽ hoạt động như bộ nhớ trong “ảo”. Và như vậy, không gian địa chỉ logic (hay còn gọi là không gian địa chỉ ảo) hoàn toàn được tách biệt và có thể lớn hơn rất nhiều không gian địa chỉ vật lý.
- Chỉ lệnh thuộc trang nào đó của tiến trình đang thực hiện thì trang chưa chỉ lệnh đó phải nằm trong bộ nhớ trong vật lý, nếu chưa trang cần được nạp vào hoặc swapping in. Ở đây có khái niệm swap in và swap out trang

III.4.2. Phân trang theo yêu cầu (Demand Paging):

- Phân trang theo yêu cầu là hệ thống tương tự như phân trang nhưng kết hợp với hoán chuyển.
- Bộ hoán chuyển lười (lazy swapper) không hoán chuyển trang khi nó được yêu cầu
- Phối hợp với phần cứng hỗ trợ để swapping và xác định trang nào đang trong bộ nhớ cơ sở, trang nào đang ở phần ảo
- Cần không gian bộ nhớ phụ làm phần bộ nhớ ảo
- Xuất hiện lỗi trang khi trang thực hiện “nằm ngoài bộ nhớ trong” => xử lý lỗi trang cũng được gọi là thay thế trang.

III.4.3. Thay thế trang (Page Replacement) hay Xử lý lỗi trang

Khi xảy ra page fault, hệ điều hành thực hiện các bước sau

- Chuyển trạng thái của tiến trình sang Waite
- Chọn một trang để thay thế (page replacement algorithm)
- Khởi động việc nạp trang mới từ bộ nhớ phụ vào bộ nhớ trong
- Chuyển thực thi cho tiến trình khác trong lúc đang thực hiện việc vào ra
- Nhận interrupt báo I/O hoàn tất (i.e. đã nạp xong trang nhớ mới)
- Chuyển trạng thái process về ready

Vấn đề đặt ra là chọn trang nào để swap out trong số các trang đang nằm trong bộ nhớ. Quan tâm tới pgae fault bởi trang swap out lúc này có thể sẽ được swap in sau đó.

III.4.3.1. Thuật toán FIFO vào trước ra trước (First In First out)

Chọn trang swap out là trang đã được swap in đầu tiên (trong số những trang đang nằm trong bộ nhớ cơ sở)

Nêu ví dụ với bài toán cấp 3 khung trang và bài toán cấp 4 khung trang, tính số lỗi trang

III.4.3.2. Thuật toán tối ưu OPT (Optimal Replacement)

Nhằm hạn chế tối đa lỗi trang, sẽ tìm trang để thay thế là trang lâu được thực hiện nhất tức là để lại những trang sắp được thực hiện.

(Thảo luận tính khả thi)

Nêu ví dụ bài toán cấp phát mỗi tiến trình 4 khung trang, tính số lỗi trang

III.4.3.3. Thuật toán lâu nhất chưa sử dụng LRU (Least Recently Used)

Thay thế trang nằm trong bộ nhớ nhưng đã lâu nhất chưa được sử dụng/ thực hiện.

Nêu ví dụ với bài toán cấp cho tiến trình 4 khung trang, tính số lỗi trang.

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng) 3 tiết

Bài 1. Trong các hệ thống áp dụng kỹ thuật Bộ nhớ ảo, mỗi tiến trình được cấp phát hạn chế số lượng “khung trang” trong khi số lượng trang thực tế tiến trình yêu cầu có thể nhiều hơn nhiều. Hãy giải thích ngắn gọn các thuật toán “Thay thế trang” và tính số lượng lỗi trang với mỗi thuật toán tương ứng đối với Tiến trình có yêu cầu truy nhập theo trình tự các trang sau: 1, 2, 3, 7, 8, 2, 2, 4, 5, 7, 1, 2, 3, 7, 8, 4, 1, 5, 6, 7, 1 với trường hợp số lượng khung trang được cấp là 4.

Bài 2. Trong các hệ thống áp dụng kỹ thuật Bộ nhớ ảo, mỗi tiến trình được cấp phát hạn chế số lượng “khung trang” trong khi số lượng trang thực tế tiến trình yêu cầu có thể nhiều hơn nhiều. Hãy giải thích ngắn gọn các thuật toán “Thay thế trang” và tính số lượng lỗi trang với mỗi thuật toán tương ứng đối với Tiến trình có yêu cầu truy nhập theo trình tự các trang sau: 1, 2, 3, 5, 7, 8, 4, 2, 3, 4, 7, 4, 2, 3, 8, 4, 8, 5, 6, 7, 1 với trường hợp số khung trang được cấp là 4.

- 6. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)
 - Tài liệu 1, Chương 10 trang 317 đến 370
 - Tài liệu 2: Chương 4 mục 4.4 từ trang 331 đến trang 336
- 7. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương IV QUẢN LÝ NGOẠI VI**

IV.1. Tổng quan Hệ thống xuất nhập và Thiết bị ngoại vi

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học) 3 tiết, tự học 3 tiết

3. Mục đích, yêu cầu:

Hiểu được cơ sở của phần cứng xuất nhập

Nắm được các dịch vụ do HĐH cung cấp và chức năng cầu nối của HĐH giữa ứng dụng với phần cứng

Nắm được nguyên lý thiết kế HĐH để cải tiến năng lực xuất nhập

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

IV.1.1 Các khái niệm cơ bản

Một trong những quan tâm chính của người thiết kế HĐH là điều khiển ngoại vi. Điều khiển ngoại vi bao gồm cả phần cứng lẫn phần mềm, hơn nữa ngoại vi lại rất phong phú, rất khác nhau giữa các chủng loại. Công nghệ thiết bị ngoại vi song hành hai xu hướng trái ngược nhau, bên cạnh việc chuẩn hóa phần mềm cùng với giao diện phần cứng là sự tăng sự đa dạng của thiết bị.

Một số khái niệm:

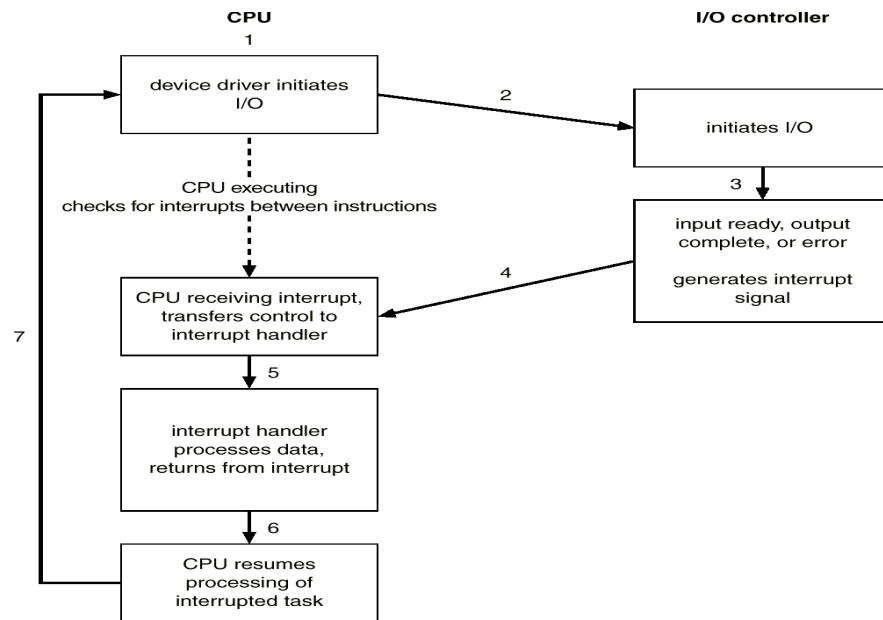
- Liên quan tới thành phần phần cứng xuất nhập: cổng (Port), BUS, Bộ điều khiển (Controller)
- Thiết bị (Device)
- Trình điều khiển thiết bị (Device Driver)
- Sự khác biệt giữa các ngoại vi về tốc độ, giao thức, cấu trúc ...

IV.1.2 Phần cứng xuất nhập

Có rất nhiều và rất đa dạng thiết bị ngoại vi. Ngoài các thiết bị phổ biến nhưng cũng rất phong phú, kết nối với máy tính nói chung như bàn phím, màn hình, con chuột, thiết bị lưu trữ, máy in ... rồi tới thiết bị nối mạng, loa, micro, máy vẽ, máy chiếu hay thẻ nhớ ngoài....còn phải kể đến các ngoại vi đặc chủng như bộ phận đếm và trả tiền của máy ATM, kênh đóng mở van cấp nhiệt của hệ thống điều khiển nhiệt hay đơn giản là một máy cắt gọt cơ khí của hệ thống máy tiện chính xác điều khiển bằng vi tính ... Một máy tính có thể kết nối với nhiều ngoại vi.

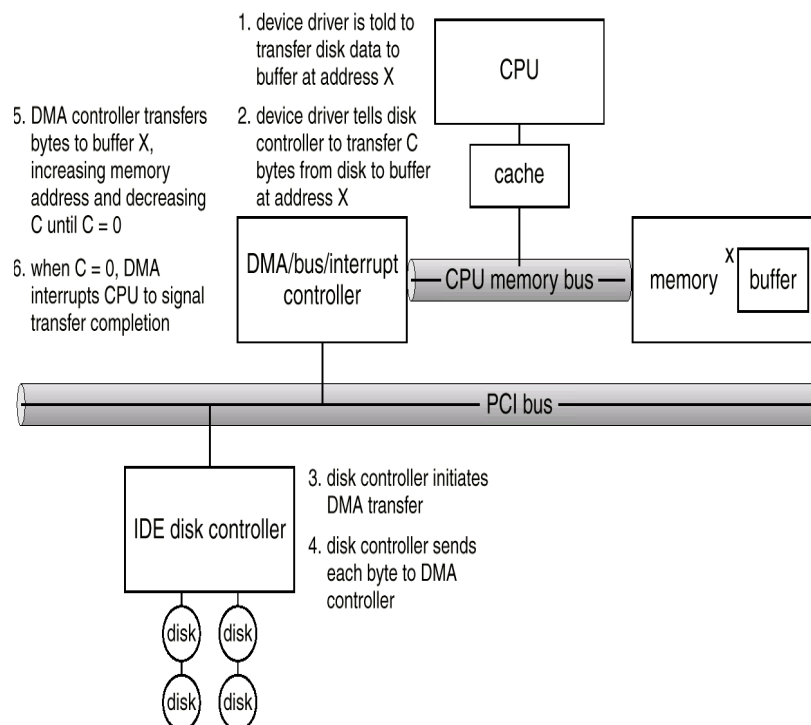
Chúng ta không nghiên cứu cái thiết bị hoạt động như thế nào mà tìm hiểu máy tính làm việc với thiết bị ngoại vi ra sao.

- Hồi vòng (Polling)
Xác định trạng thái của thiết bị (State of device) bao gồm
 - Sẵn sàng (command-ready)
 - Bận (Busy)
 - Gặp lỗi (Error)Cơ chế này có thể xuất hiện chu kỳ chờ I/O chậm chạp làm việc
- Ngắt (Interrupt)
Khác hẳn so với hồi vòng, với cơ chế ngắt, ngoại vi sẵn sàng sẽ báo cho hệ thống thông qua tín hiệu ngắt. Vấn đề đặt ra là phục vụ ngắt, và vì vậy có một số khái niệm:
 - Phục vụ ngắt
 - Ngắt che được và ngắt không che
 - Độ ưu tiên
 - Vector ngắt



- Truy nhập trực tiếp bộ nhớ DMA (Direct Memory Access)

Với việc trao đổi với ngoại vi một lượng lớn dữ liệu, CPU sẽ tốn nhiều công sức để thực hiện các lệnh in/out thông thường, với hỗ trợ của phần cứng, nhiều hệ thống cho phép chuyển phần lớn công việc này cho bộ phận đặc biệt được gọi là bộ điều khiển truy nhập trực tiếp DMA.



IV.1.3 Giao diện xuất nhập ứng dụng (Application I/O Interface)

Phần này sẽ tìm hiểu kỹ hơn các kỹ thuật cấu trúc và giao diện của HĐH sao cho các thiết bị xuất nhập được nhìn nhận một cách thống nhất và chuẩn hóa.

- I/O system calls encapsulate device behaviors in generic classes
- Device-driver layer hides differences among I/O controllers from kernel
- Devices vary in many dimensions
 - Chuỗi ký tự hay là khối
 - Truy cập tuần tự hay trực tiếp
 - Đồng bộ hay bất đồng bộ
 - Chia sẻ hay độc chiếm
 - Tốc độ hoạt động
 - Quyền truy nhập

IV.1.3.1 Thiết bị ký tự và thiết bị khối (Block and Character Devices)

- Block devices/ thiết bị khối bao gồm ổ đĩa và một số dạng thiết bị lưu trữ. Các lệnh truy nhập chẳng hạn như read(), write() và đối với thiết bị cho phép truy nhập trực tiếp (hay còn gọi là ngẫu nhiên) là seek().
 - Raw I/O or file-system access
 - Memory-mapped file access possible
- Character devices/ thiết bị ký tự bao gồm các thiết bị tiêu biểu như bàn phím, con chuột và cổng nối tiếp. Các lệnh truy nhập đặc trưng như get() hay put() từng ký tự.
 - Thảo luận về USB: tên, giao thức, kết nối và thiết bị kết nối.

IV.1.3.2 Nhập xuất khóa và nhập xuất không khóa (Blocking and Nonblocking I/O)

Một khía cạnh khác của giao thức lời gọi hệ thống liên quan tới việc chọn giữa xuất nhập chặn (blocking I/O) và xuất nhập không chặn (nonblocking I/O) hay còn được gọi là không đồng bộ (asynchronous). Khi ứng dụng gặp lời gọi hệ thống chặn thì việc chạy ứng dụng tạm dừng, ứng dụng sẽ được chuyển sang hàng đợi “chờ” (wait queue). Sau khi lời gọi hệ thống hoàn thành, tiến trình được chuyển vào hàng đợi sẵn sàng để chờ được cấp CPU thực hiện tiếp. Ở một góc khác chúng ta nhận có những hành động vật lý được thực hiện bởi thiết bị ngoại vi diễn ra không đồng bộ. Một số tiến trình ở mức người sử dụng cần các xuất nhập không chặn này, chẳng hạn như giao tiếp người sử dụng nhận ký tự từ bàn phím và đầu vào con chuột trong khi đang xử lý và hiển thị dữ liệu trên màn hình.

Có một số khái niệm và vấn đề liên quan cần nghiên cứu:

- Giao diện và sao dữ liệu người sử dụng (buffered I/O)
- Đa luồng (multithreading)
- Hệ thống con xuất nhập của nhân bao gồm:
 - Định thời xuất nhập,
 - Vùng đệm (buffering),
 - Vùng lưu chứa (caching)
 - Vùng chứa và vùng chứa giữ trước thiết bị (spooling and device reservation)
 - Quản lý lỗi (error handling),
 - Cấu trúc dữ liệu nhân (Kernel data structures)

IV.1.4 Hiệu năng hệ thống (Performance)

Tất cả các cấu thành của hệ thống tác động tới hiệu năng của nó; và hệ thống xuất nhập cũng là một yếu tố quan trọng.

- Yêu cầu CPU thực hiện các device driver, kernel I/O code
- Chuyển context khi có ngắt (Context switches due interrupts)
- Sao lưu dữ liệu (Data copying)

- Xử lý xung đột đường truyền mạng (Network traffic)

Từ các căn cứ và nguyên tắc trên đưa ra các biện pháp hay vấn đề cần lưu tâm nhằm nâng cao hiệu năng của hệ thống.

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)

- Tài liệu 1: Chương 13 từ trang 455 đến trang 490.
- Tài liệu 2: chương 3, từ trang 153 đến trang 165

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương IV QUẢN LÝ NGOẠI VI**

IV.2: Tắc nghẽn (Deadlocks)

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học)

3. Mục đích, yêu cầu:

Hiểu mô hình hệ thống với vấn đề tắc nghẽn, các đặc điểm của chúng

Nắm được các phương pháp quản lý, ngăn chặn và tránh tắc nghẽn

Hiểu được cách phát hiện và phục hồi từ tắc nghẽn

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

IV.2.1 Tắc nghẽn

Trong môi trường đa trình đa nhiệm, khi có nhiều tiến trình cùng đồng hành lại yêu cầu một số lượng hữu hạn tài nguyên phần cứng, đặc biệt là các ngoại vi khá chậm (một cách tương đối), chúng xung đột với nhau. Một tiến trình khi yêu cầu các tài nguyên, nếu tài nguyên đó không thể đáp ứng tài thời điểm hiện thời thì tiến trình sẽ phải chuyển sang trạng thái chờ. Các tiến trình có thể chờ mãi mãi bởi vì tài nguyên mà nó yêu cầu bị giữ bởi một tiến trình khác chưa thể giải phóng trong khi tiến trình này lại cũng đang chờ tài nguyên.

Tắc nghẽn thường xảy ra do xung đột về tài nguyên thuộc dạng không chia sẻ, tuy nhiên trong một số ít trường hợp tắc nghẽn vẫn có thể xảy ra với tài nguyên chia sẻ được.

IV.2.2. Mô hình hệ thống

Hệ thống gồm các loại tài nguyên, kí hiệu R_1, R_2, \dots, R_m , ví dụ: CPU, không gian bộ nhớ, thiết bị I/O nào đó.

Mỗi loại tài nguyên R_i có W_i thực thể (instance).

Để sử dụng tài nguyên, tiến trình thực hiện theo các bước sau:

- Yêu cầu (request): process phải chờ nếu yêu cầu không được đáp ứng ngay
- Sử dụng (use): process sử dụng tài nguyên
- Hoàn trả (release): process hoàn trả tài nguyên

Trong đó các tác vụ yêu cầu (request) và hoàn trả (release) đều là system call

IV.2.3. Điều kiện hình thành tắc nghẽn (Deadlock Characterization)

Tắc nghẽn chỉ xảy ra khi đồng thời cả bốn điều kiện sau thỏa mãn:

1. Điều kiện ngăn chặn lẫn nhau (mutual exclusion): các tiến trình thực hiện loại trừ lẫn nhau trên đoạn găng. VD: Cập nhật dữ liệu một bản ghi vào một CSDL trên mạng.
2. Điều kiện giữ và đợi (hold & wait): tiến trình đang giữ ít nhất một tài nguyên và yêu cầu thêm tài nguyên khác mà chúng lại đang bị giữ bởi một tiến trình khác.
3. Điều kiện không có ưu tiên trước Nonpre-emption: tài nguyên chỉ được giải phóng khi tiến trình đang giữ nó kết thúc công việc.
4. Điều kiện đợi vòng (circular-wait): các tiến trình giữ và đợi tài nguyên tạo thành vòng luân chuyển. VD: P0 đợi tài nguyên bị giữ bởi P1, P1... P2, P2... Pn, Pn .. P0.

Đồ họa phân phối tài nguyên R cho các tiến trình P với deadlock và không có deadlock xem kỹ trang 246 đến trang 249 chương 8 tài liệu 1.

IV.2.4. Ngăn ngừa tắc nghẽn (deadlock prevention)

Đảm bảo ít nhất một trong bốn điều kiện sinh ra deadlock không được xảy ra.

- Không để xảy ra điều kiện ngăn cản lẫn nhau (mutual exclusion)
Đối với các tài nguyên không thể chia sẻ (nonsharable resource - vd: printer): Hệ điều hành sử dụng kỹ thuật SPOOL để tạo ra nhiều tài nguyên ảo cung cấp cho các tiến trình đồng thời. Đối với sharable resource (vd: read-only file): không cần thiết
- Không để xảy ra điều kiện Hold and Wait
Cách 1: mỗi process yêu cầu toàn bộ tài nguyên cần thiết một lần. Nếu có đủ tài nguyên thì hệ thống sẽ cấp phát, nếu không đủ tài nguyên thì process phải bị blocked.
Cách 2: khi yêu cầu tài nguyên, process không được giữ bất kỳ tài nguyên nào. Nếu đang giữ thì phải trả lại trước khi yêu cầu.
- Không sử dụng có chế không có ưu tiên trước (NoPre-emption):
Nếu process A có giữ tài nguyên và đang yêu cầu tài nguyên khác nhưng tài nguyên này chưa cấp phát ngay được thì:
Cách 1: Hệ thống đòi lại mọi tài nguyên mà A đang giữ, A chỉ bắt đầu lại được khi có được các tài nguyên ban đầu cùng với tài nguyên đang yêu cầu.
Cách 2: Hệ thống sẽ xem tài nguyên mà A yêu cầu: Nếu tài nguyên được giữ bởi một process khác đang đợi thêm tài nguyên, tài nguyên này được hệ thống đòi lại để cấp phát cho A. Nếu tài nguyên được giữ bởi process không đợi tài nguyên nào khác nữa, A phải đợi và tài nguyên mà A đang giữ bị hệ thống đòi lại. Tuy nhiên hệ thống chỉ đòi lại các tài nguyên mà có process khác yêu cầu.
- Không để xảy ra chờ đợi vòng tròn Circular Wait: Điều kiện này có thể ngăn chặn bằng cách phân lớp tài nguyên hệ thống. Theo đó mỗi tiến trình được cấp phát tài nguyên lớp L thì nó chỉ có thể yêu cầu các tài nguyên ở lớp thấp hơn L.

IV.2.5 Tránh deadlock (deadlock avoidance)

Tránh deadlock là vẫn đảm bảo hiệu suất sử dụng tài nguyên tối đa đến mức có thể. Mỗi tiến trình process phải khai báo số lượng tài nguyên tối đa mà nó cần để thực hiện công việc.

Giải thuật tránh deadlock phải kiểm tra trạng thái cấp phát của tài nguyên (resource-allocation state) để bảo đảm hệ thống không bao giờ rơi vào trạng thái deadlock. Trạng thái cấp phát tài nguyên được định nghĩa dựa trên số tài nguyên còn lại, số tài nguyên đã được cấp phát và số tài nguyên tối đa mà mỗi process yêu cầu.

- Trạng thái an toàn và không an toàn (Safe and Unsafe State)
- Quan hệ giữa trạng thái safe/unsafe và deadlock
- Thuật toán nhà băng (Banker's Algorithm).

IV.2.6. Phát hiện deadlock (deadlock detection)

Nếu chấp nhận xảy ra deadlock trong hệ thống, thì ta cần kiểm tra trạng thái của hệ thống bằng giải thuật phát hiện deadlock.

Nếu có deadlock thì tiến hành phục hồi hệ thống

Các giải thuật phát hiện deadlock thường sử dụng mô hình RAG.

Hệ thống cấp phát tài nguyên được khảo sát trong các trường hợp sau

Mỗi loại tài nguyên chỉ có một thực thể (instance)

Mỗi loại tài nguyên có thể có nhiều thực thể (instances)

IV.2.7. Khắc phục deadlock (Recovery from Deadlock)

Cách 1: Báo cho người vận hành (operator) hệ thống. Khi đó người vận hành sẽ tự xác định biện pháp và khắc phục tắc nghẽn một cách thủ công.

Cách 2: Hệ thống tự động phục hồi bằng cách phá vỡ điều kiện đợi vòng (circular wait) hoặc cách khác là cưỡng bức (preempt) tài nguyên của một hay một số tiến trình.

- Phá điều kiện đợi vòng bằng cách kết thúc tiến trình
 - Thoát/ hủy tất cả các tiến trình tạo thành deadlock
 - Thoát/hủy từng tiến trình cho đến khi hệ thống thoát khỏi deadlock
- Cưỡng bức tài nguyên: giải phóng bắt buộc tài nguyên khỏi tiến trình; xuất hiện các vấn đề cần quan tâm:

- Chọn nạn nhân: tức là tài nguyên nào của tiến trình nào?
 - Vấn đề phục hồi công việc của tiến trình bị mất tài nguyên.
 - Đảm bảo là nạn nhân không chỉ tập trung vào một tiến trình
- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng)
5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)
- Tài liệu 1: Chương 8, từ trang 241 đến trang 270.
 - Tài liệu 2: Chương 3 mục 3.3 từ trang 166 đến trang 179
6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương IV QUẢN LÝ NGOẠI VI**

IV.3 Thiết bị lưu trữ

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học)

3. Mục đích, yêu cầu:

Nắm được và phân biệt các loại thiết bị lưu trữ ngoài, đặc biệt là đĩa cứng

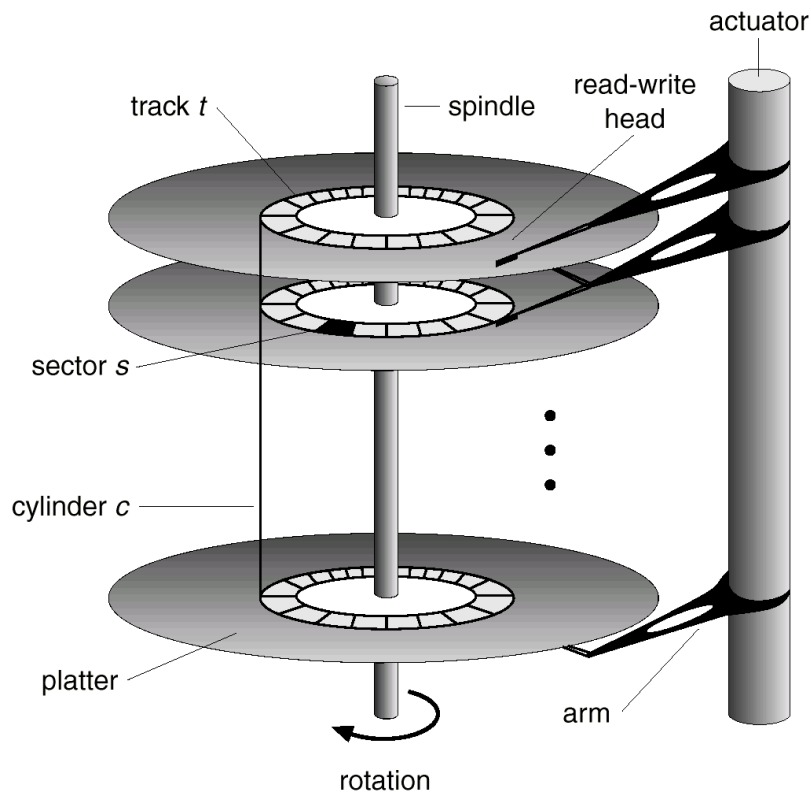
Nắm được các thuật toán căn bản lập lịch cho đầu từ đĩa cứng, hướng tới lập trình mô phỏng dịch chuyển đầu từ.

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

IV.3.1 Giới thiệu chung về thiết bị lưu trữ

- Các loại thiết bị lưu trữ dạng đĩa bao gồm đĩa mềm, đĩa MO, ZIP, đĩa CD, và DVD
- Đĩa cứng (Hard Disk):



- Tape và tape backup

IV.3.2 Lập lịch cho đầu từ (Arm Scheduling)

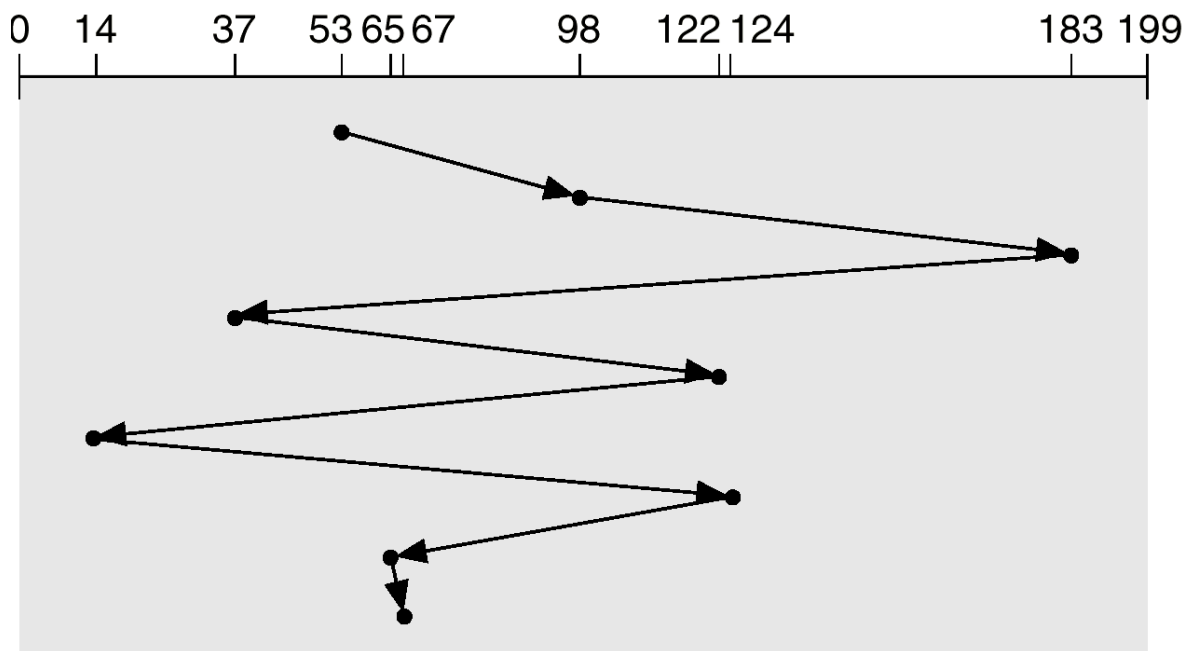
Một nhiệm vụ quan trọng của HĐH là khai thác phần cứng hiệu quả. Đối với đĩa cứng đó chính là thời gian truy cập nhanh và lưu trữ được nhiều trên đĩa nhất.

Thời gian truy cập bao gồm hai thành phần: Thời gian dịch chuyển đầu từ và thời gian trễ vòng quay, trong đó thời gian trễ vòng quay có thể nói là xác định và phụ thuộc phần cứng, trong khi thời gian định vị đầu từ tới các rãnh chứa thông tin cần truy cập của tệp tin trên đĩa là tổng thời gian đầu từ dịch chuyển tới tất cả các rãnh chứa thông tin. Bởi vậy việc dịch chuyển đầu từ tới rãnh nào trước, rãnh nào sau có ảnh hưởng rất lớn tới thời gian truy cập.

- Thời gian định vị đầu từ (Seek Time) là thời gian đầu từ dịch chuyển từ vị trí hiện tại tới rãnh chứa block thông tin cần truy cập.
- Thời gian trễ vòng quay (Rotational Latency) là thời gian chờ để đĩa quay thêm cho đến khi đầu từ đọc được thông tin trên rãnh đó.
- Có một số thuật toán dưới đây để dịch chuyển đầu từ nhằm đọc được một tệp tin được lưu trên đĩa tại các rãnh theo trình tự: 98, 183, 37, 122, 14, 124, 65, 67, và đầu từ bắt đầu từ rãnh số 53, đĩa có 200 rãnh đánh số từ 0 đến 199.

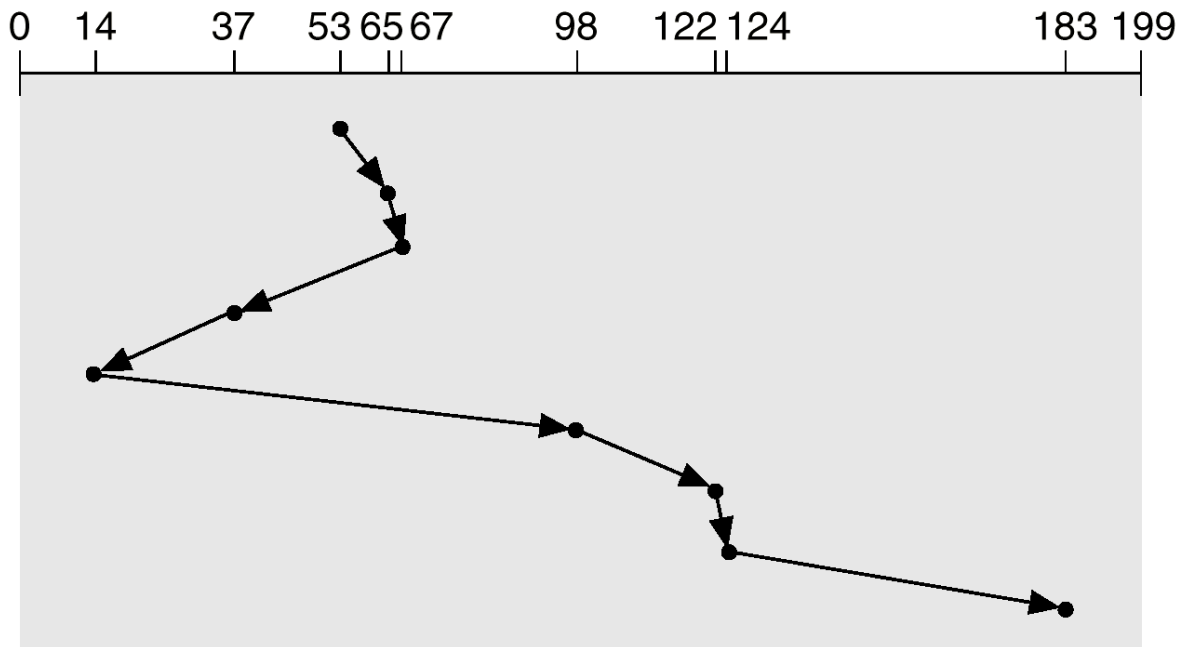
1. Thuật toán đến trước phục vụ trước FCFS (First Come First Served):

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



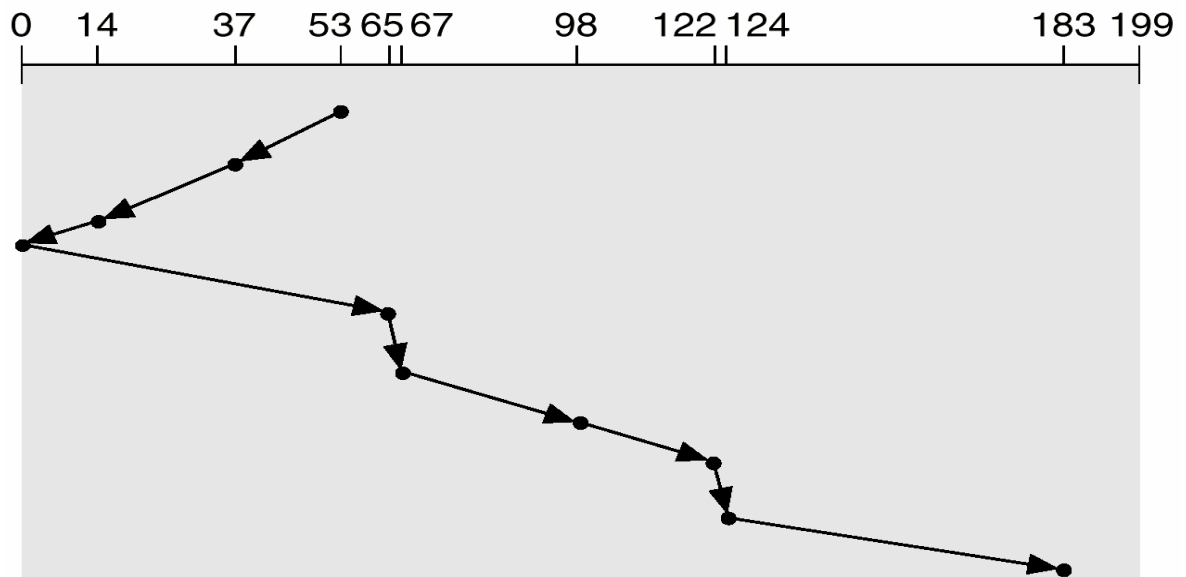
2. Thuật toán phục vụ yêu cầu gần nhất SSTF (Shortest Seek Time First)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



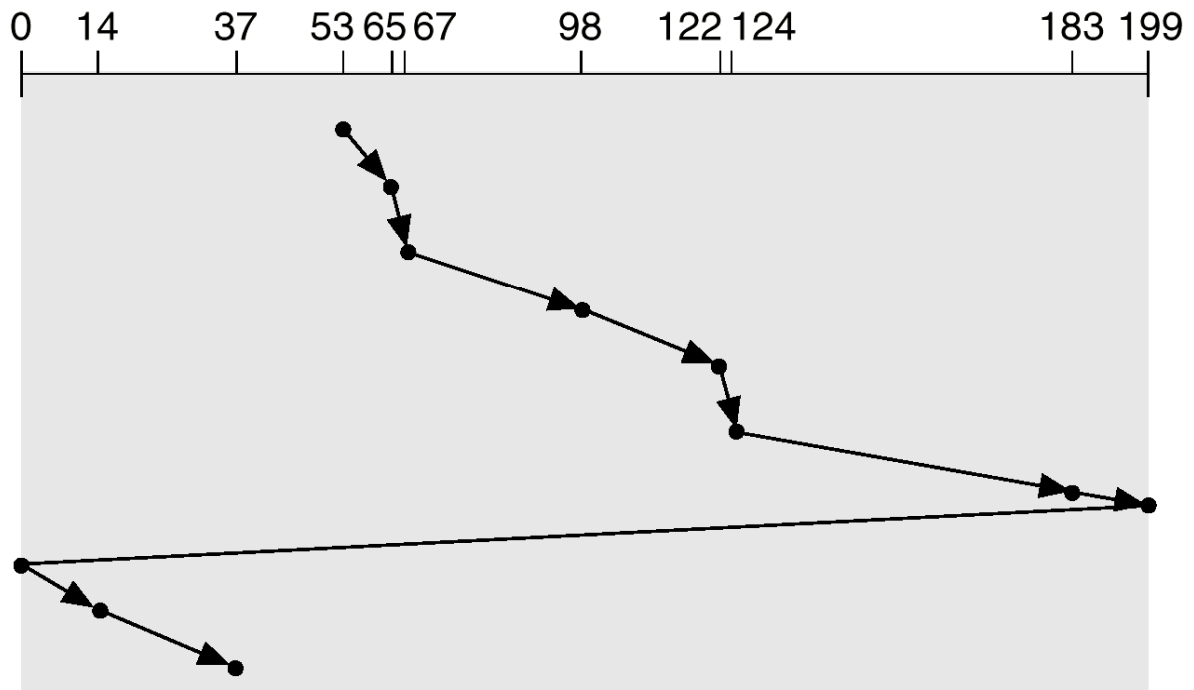
3. Thuật toán thang máy SCAN (Scan Algorithm)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



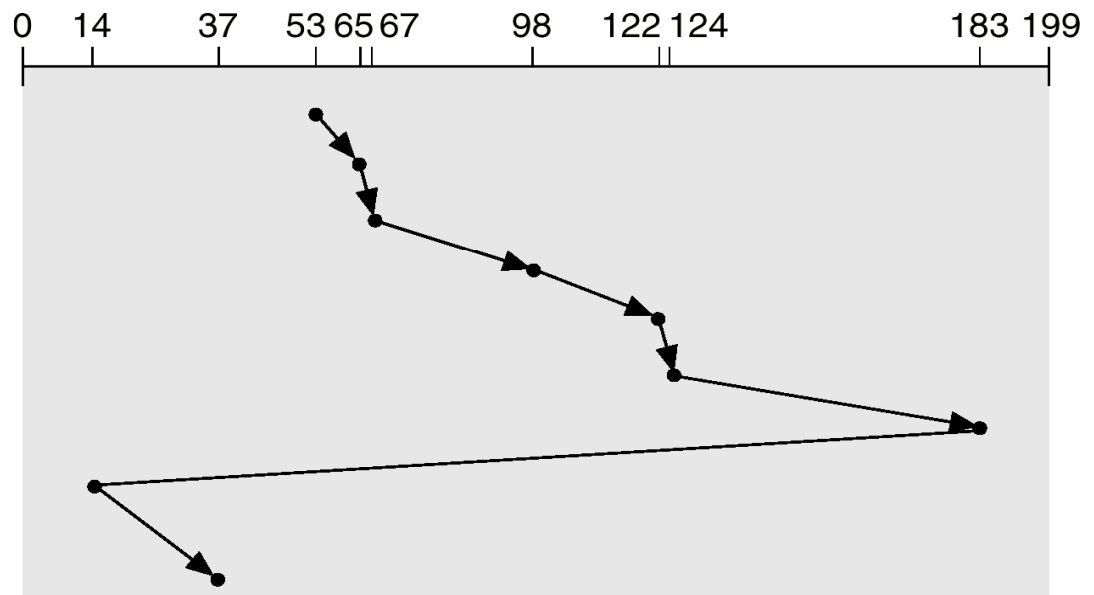
4. Thuật toán C-SCAN

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



5. Thuật toán C-LOOK

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



b) Nội dung thảo luận

c) Nội dung tự học:

Đọc mở rộng và nâng cao về RAID, các loại thẻ nhớ và flash disk (thường gọi là USB)

d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chỉ tiết đến chương, mục, trang)

- Tài liệu 1: Chương 14 từ trang 491 đến trang 530.

6. Câu hỏi ôn tập

HỌC PHẦN: **LÝ THUYẾT HỆ ĐIỀU HÀNH**

Bộ môn: Khoa học Máy tính

Giáo viên: 1) TRỊNH Minh Châu
2)

1. Bài (chương, mục): **Chương IV QUẢN LÝ NGOẠI VI**

IV.4 Hệ thống tệp tin

2. Thời lượng: (GV giảng, thảo luận, thực hành, tự học)

3. Mục đích, yêu cầu:

Hiểu được các khía cạnh khác nhau đối với tệp tin và cấu trúc thư mục

Nắm được các cơ chế quản lý, kiểm soát, bảo vệ tệp tin trong môi trường đa trình đa nhiệm với nhiều người sử dụng; hiểu cách chia sẻ tệp tin

4. Nội dung:

a) Nội dung chi tiết: (công thức, định lý, hình vẽ)

Khác với phần thiết bị lưu trữ, trong phần này vấn đề quan tâm là các khía cạnh khác nhau của tệp tin và cấu trúc thư mục các tệp tin được cất giữ trên thiết bị lưu trữ. Chúng ta sẽ nghiên cứu cách tổ chức, quản lý và bảo vệ tệp tin khi có nhiều người có quyền truy cập tệp tin.

IV.4.1 Tệp tin (File Concept)

Thông tin lưu trữ trên các loại thiết bị lưu trữ khác nhau về mặt vật lý có thể khác nhau. Để tiện dụng cho người sử dụng, HĐH cung cấp một cái nhìn logic thống nhất đối với việc lưu trữ thông tin. File hay tệp tin là đơn vị nhỏ nhất của HĐH quản lý dưới góc nhìn người sử dụng. File chính là một tập hợp thông tin có liên quan được ghi trên bộ nhớ ngoài.

Có các loại file sau: chương trình, dữ liệu, âm thanh, hình ảnh, Multimedia,...

File có thể có/ không có cấu trúc.

IV.4.1.1 Thuộc tính tệp tin (File Attributes)

- Tên file (Name),
- Định danh (Identifier),
- Kiểu file (Type),
- Vị trí (Location),
- Kích cỡ file (Size),
- Các thông tin bảo vệ như quyền truy cập (Protection – acces control), thời gian và định danh người dùng (User Identifier)

Thông tin về tất cả các tệp tin được lưu trong cấu trúc thư mục (directory)

IV.4.1.2 Tác nghiệp trên tệp tin (File Operations)

Tệp tin là một kiểu dữ liệu trừu tượng. Để định nghĩa chính xác một tệp tin, cần xem xét các tác nghiệp có thể thực hiện đối với một tệp tin. HĐH cung cấp các lời gọi hệ thống để thực hiện các công việc này, bao gồm:

- Khởi tạo tệp tin (Creating),
- Mở (Opening)
- Đóng (Closing)
- Đọc (Reading)
- Ghi (Writing)
- Chèn vào cuối (Appending)
- Xóa tệp tin (Deleting)
- Tìm/định vị (Repositioning/Seeking)

- Đổi tên (Renaming)
- Xóa nội dung tệp tin (Truncating)

IV.4.1.3 Các kiểu tệp tin (File Type)

Thông thường phần tên file mở rộng thể hiện kiểu của tệp tin. Chẳng hạn như file thực hiện được có phần tên mở rộng là .com hay .exe hay .bin, file chương trình mã nguồn: .c, .pas, .cpp hay .asm (xem thêm trang 377 tài liệu 1).

Ngoài ra cũng có thể nhóm chúng lại làm mấy dạng tệp tin sau:

- Tệp tin thường
- Thư mục
- Tệp có ký tự đặc biệt
- Tệp tin dạng khối

IV.4.2 Các phương thức truy nhập tệp tin (Access Methods)

(Xem thêm phần các phương pháp cấp phát – Allocation Methods, ở mục sau)

Tệp tin chứa thông tin lưu trữ trên bộ nhớ ngoài. Khi cần thiết chúng phải được đọc vào bộ nhớ trong của hệ thống máy tính. Có một số phương thức truy cập. Các mô hình hệ thống khác nhau có thể cho phép đọc bằng một phương thức, có hệ thống có thể cho phép truy cập bằng nhiều nhiều phương thức:

- Truy cập tuần tự (Sequential Access)
Read next
Write next
- Truy cập trực tiếp (Direct Access)
Read n
Write n
- Các phương pháp truy cập khác (Other Access Methods)
Các phương pháp phát triển trên cơ sở của phương thức truy cập trực tiếp, thường liên quan tới việc xây dựng chỉ mục cho tệp tin. Chỉ mục là con trỏ chỉ tới các khối khác.

IV.4.3 Cấu trúc thư mục (Directory Structure)

Một thiết bị lưu trữ thường chứa rất nhiều tệp tin, có thể là hàng ngàn tới hàng triệu, vì thế cần phải tổ chức lại chúng. Việc tổ chức này thường được thực hiện bởi hai phần:

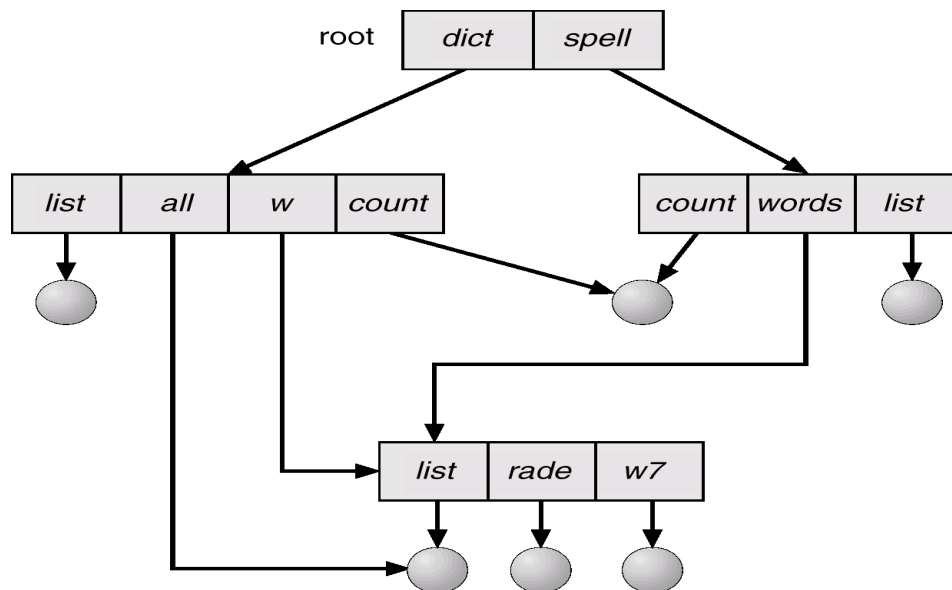
Thứ nhất là chia chúng thành hơn các phân khu (partitions) hay phân vùng (Volumes), thứ hai là mỗi phân khu chứa thông tin về các tệp tin thuộc phân khu đó. Thông tin này được lưu ở phần đầu mỗi phân khu gọi là thư mục hay bảng mục lục phân vùng.

Các thao tác có thể thực hiện trên thư mục:

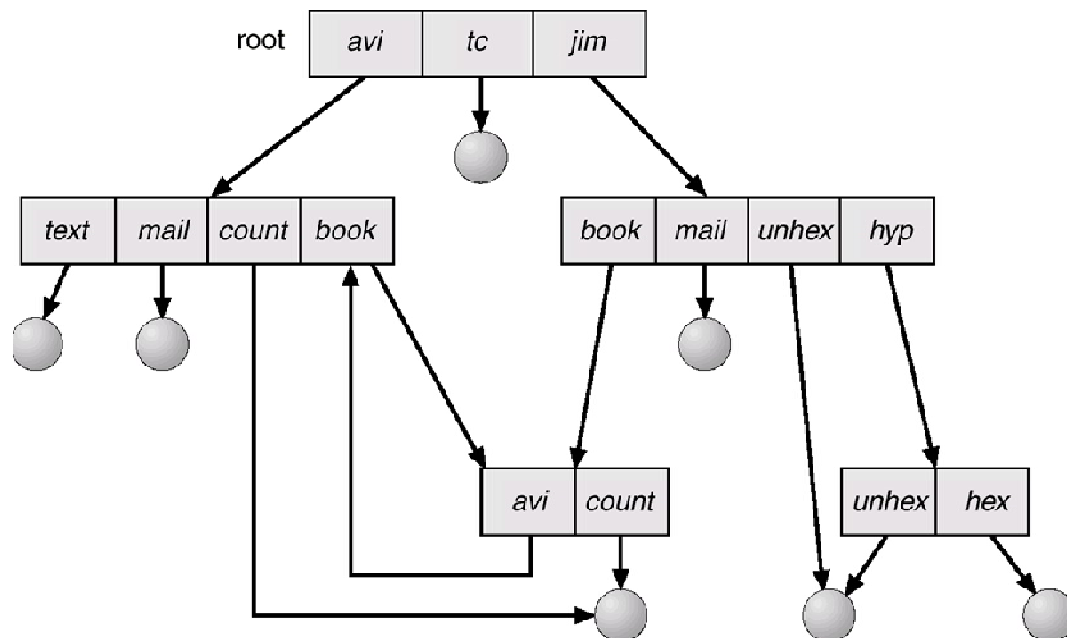
- Tìm kiếm tệp tin
- Khởi tạo tệp tin,
- Xóa tệp tin,
- Liệt kê thư mục,
- Đổi tên thư mục,
- Duyệt hệ thống tệp tin

Cấu trúc thư mục:

- Đơn cấp,
- Hai cấp,
- Dạng hình cây (Tree-Structure Directory)



- Cấu trúc thư mục đồ thị tổng quát (General Graph Directory)



IV.4.4 Bảo vệ tệp tin (Protection)

Thông tin của hệ thống máy tính dù được lưu trữ trên thiết bị lưu trữ ngoài vẫn cần được bảo đảm an toàn ở những mức độ theo yêu cầu quan trọng của thông tin. Khả năng tin cậy của các tệp tin thường được đảm bảo bằng cách nhân bản chúng dưới các hình thức khác nhau với mục đích là thông tin bị hỏng có thể khôi phục lại nhanh chóng và dễ dàng. Sao lưu và Backup tự động là biện pháp hữu hiệu, về vấn đề này có thể đọc thêm về RAID. Ở đây sẽ bàn về vấn đề bảo vệ tệp tin trước những truy nhập từ người dùng.

IV.4.4.1 Các kiểu truy cập (Type of Access)

Quản lý, giới hạn quyền truy cập tới tệp tin để thực hiện các thao tác sau trên tệp tin:

- Đọc (Read)
- Ghi (Write)
- Thực thi (Execute)

- Thêm vào cuối (Append)
- Xóa (Delete)
- Hiển thị (List)

IV.4.4.2 Kiểm soát truy cập (Access Control)

Người dùng, nhóm người sử dụng khác nhau có thể được quản lý bởi việc cấp các quyền truy cập khác nhau.

Về mode truy cập lên tệp tin có thể chia thành ba loại bao gồm:

- Đọc (Read – ví dụ mã hóa là R) không làm thay đổi nội dung tệp tin,
- Ghi (Write – ví dụ mã hóa là W) có thể thay đổi nội dung tệp tin
- Thực hiện (Execute – ví dụ mã hóa là X) thực hiện file

Có thể chia người sử dụng thành các lớp ứng với các quyền và mức độ bị quản lý khác nhau như:

- Sở hữu (Owner) là người tạo ra file đó; có quyền RWX
- Nhóm (Group) là tập hợp những NSD chia sẻ tệp tin có quyền truy cập tương tự hay cùng nhóm làm việc; có quyền RW
- Người dùng khác (Public) là những NSD, khai thác khác; có quyền R

IV.4.4.3 Các tiếp cận bảo vệ khác (Other Protection Approaches)

Khá phổ biến và dễ phát triển như cơ chế mật khẩu (Password). Có hệ thống cho phép gắn mật khẩu với từng tệp tin, có hệ thống còn cho phép gắn mật khẩu với cả thư mục con

IV.4.5 Các phương pháp cấp phát (Allocation Methods)

Đối với đĩa, việc truy cập trực tiếp cho phép khả năng linh hoạt trong việc cài đặt tệp tin. Vấn đề đáng quan tâm là cấp phát không gian lưu trữ thế nào để sử dụng hiệu quả và các tệp tin có thể truy cập nhanh chóng. Có ba phương pháp chính cấp phát không gian đĩa: cấp phát kề, liên kết và chỉ mục:

- Cấp phát kề (Contiguous Allocation)

Với phương pháp này, mỗi tệp tin chiếm một tập hợp các khối liên kề nhau trên đĩa.

- Ưu điểm
- Nhược điểm
- Cấp phát liên kết
- Cấp phát theo chỉ số

- b) Nội dung thảo luận
- c) Nội dung tự học
- d) Bài tập (bắt buộc, mở rộng)

5. Tài liệu tham khảo (sách, báo – chi tiết đến chương, mục, trang)

- Tài liệu 1, chương 11 từ trang 371 đến trang 410

6. Câu hỏi ôn tập