[Câu 1: Ý nghĩa của System Call. Hãy giải thích bình thường các phần mềm ứng dụng “khai thác” phần cứng của máy như thế nào? 3](#_Toc325060469)

[Câu 2: Một tiến trình yêu cầu Hệ điều hành chuyển trạng thái cho nó “ngủ” 5 giây. Hệ điều hành có đảm bảo được là sẽ kích hoạt tiến trình này đúng 5 giây sau đó không? Tại sao? 3](#_Toc325060470)

[Câu 3: Nêu định nghĩa “Hệ điều hành”. Hệ điều hành thực hiện những chức năng chính gì? 3](#_Toc325060471)

[Câu 4: Multi-tasking là gì? Giải thích sự khác nhau giữa các hình thức Multi-tasking (cooperative and pre-emtive multi-tasking). 4](#_Toc325060472)

[Câu 5: Sự giống nhau và khác nhau giữa Chương trình, Tiến trình và Luồng (Program, Process and Thread). 4](#_Toc325060473)

[Câu 6: So sánh ưu và nhược điểm giữa Hệ điều hành client-server và hệ điều hành đơn khối (monolithic) truyền thống. 5](#_Toc325060474)

[Câu 7: Hãy giải thích cơ chế CPU chuyển từ việc thực hiện process này sang process khác. Làm sao có thể đảm bảo được việc thực hiện đúng các lệnh của process tương ứng? 6](#_Toc325060475)

[Câu 8: Mục đích và nguyên tắc của việc lập lịch cho CPU (Process Scheduling) 7](#_Toc325060476)

[Câu 9: Ý nghĩa căn bản của Lập lịch, Sự khác nhau giữa Lập lịch dài kì và lập lịch ngắn hạn? 7](#_Toc325060477)

[Câu 10, 11, 12, 13: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện. Đánh giá thuật toán FCFS, SJF, SRF, RR 7](#_Toc325060478)

[Câu 14: Giải thích tại sao phần địa chỉ cao (top-half) của hệ điều hành không phải là tiến trình (process). Giải thích tại sao phần địa chỉ thấp (bottom-half) của hệ điều hành cũng không phải là tiến trình (process). 9](#_Toc325060479)

[Câu 15: Giải thích ngắn gọn trạng thái của các tiến trình, các mô hình tiến trình hệ thống (process models) chung. 10](#_Toc325060480)

[Câu 16: Giải thích vai trò của PCB (process control block). 10](#_Toc325060481)

[Câu 17: Giới thiệu và giải thích ngắn gọn các thành phần (components) của Hệ điều hành. 10](#_Toc325060482)

[Câu 18: Nêu vắn tắt các kĩ thuật cấp phát bộ nhớ (nạp chương trình vào bộ nhớ). 10](#_Toc325060483)

[Câu 19. Trong kỹ thuật cấp phát bộ nhớ phân vùng động, khi nạp tiến trình mới vào bộ nhớ cần lựa chọn một vùng còn “ rỗng” nào đó. Nêu các phương pháp/thuật toán lựa chọn. 10](#_Toc325060484)

[Câu 20: Phân tích hai khái niệm Page và Paging (trang và phân trang). 10](#_Toc325060485)

[Câu 21: Giả thích cơ chế phân đoạn (segmentation) trong quản lý bộ nhớ của hệ điều hành. 10](#_Toc325060486)

[Câu 22: Mô tả ngắn gọn “Bộ nhớ ảo” – Virtual memory. Bộ nhớ ảo có lợi và thiệt hại gì đối với việc tối ưu hoá sử dụng CPU. 10](#_Toc325060487)

[Câu 23: Trong kỹ thuật bộ nhớ ảo thường sử dụng Phân trang theo yêu cầu (demand paging). Hãy giới thiệu cơ chế này. 10](#_Toc325060488)

[Câu 24: Trong kỹ thuật sử dụng Bộ nhớ ảo Hệ điều hành cần có bộ phận quản lý việc hoán chuyển các trang/đoạn giữa bộ nhớ thực và bộ nhớ ảo và có thể xảy ra “Lỗi trang” (page fault). Hệ điều hành giải quyết vấn đề đó như thế nào và mục tiêu cần đạt được? 11](#_Toc325060489)

[Câu 25: Đánh giá các thuật toán thay trang (Page Replacement) trong kỹ thuật sử dụng Bộ nhớ ảo. 11](#_Toc325060490)

[Câu 26: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán FIFO. 12](#_Toc325060491)

[Câu 27: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán Tối ưu OPT. 12](#_Toc325060492)

[Câu 28: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán LRU. 13](#_Toc325060493)

[Câu 29: Phân biệt hai hiện tượng phân mảnh nội (internal fragmentation) và phân mảnh ngoài (external fragmentation), chúng xuất hiện khi nào và tại sao? 13](#_Toc325060494)

[Câu 30: Giải thích khái niệm “hoán chuyển” (swapping); Ý nghĩa, ứng dựng của việc áp dụng kĩ thuật này. 13](#_Toc325060495)

[Câu 31: Hãy giải thích vì sao DMA thường được “ưu ái” sử dụng như là phương thức thực thi trao đổi với ngoại vi? DMA có lợi điểm gì không đối với input thông qua bàn phím? Giải thích tại sao hoặc tại sao không? 13](#_Toc325060496)

[Câu 32: Các máy tính nguyên chiếc sử dụng phổ biến hiện nay thường cài đặt Hệ điều hành Windows XP hoặc Vista; đây là các hệ thống/ Hệ điều hành Đa trình đa nhiệm. Hãy giới thiệu một số dạng hệ thống/ hệ điều hành khác. 14](#_Toc325060497)

[Câu 33: Khi học môn “Tin học đại cương” chúng ta đã được biết đến một cách chia các thiết bị ngoại vi thành 2 hoặc 3 loại (nhóm) là “VÀO’, “RA” và/hoặc vừa “VÀO vừa RA”; Dưới góc nhìn người thiết kế Hệ điều hành chúng ta có thể chia thiết bị ngoại vi thành 2 loại. Đó là gì và cách chia này dựa trên căn cứ nào? Hãy nêu ví dụ một số ngọai vi thuộc từng loại trên. 14](#_Toc325060498)

[Câu 34: Giải thích sự khác biệt giữa “polled I/O” và “interrupt-driven I/O”. Nêu những ưu điểm của “interrup-driven I/O” so với “polled I/O”. 14](#_Toc325060499)

[Câu 35: Một cách tương đối, ổ đĩa cứng chậm hơn rất nhiều so với CPU. Nêu các yếu tố đánh giá tốc độ truy xuất dữ liệu trên đĩa. Hãy nêu ngắn gọn các kĩ thuật mà Hệ điều hành sử dụng để nâng cao thông lượng (throughput) trung bình của đĩa. 14](#_Toc325060500)

[Câu 36, 37, 38, 39, 40: Hãy giải thích thuật toán Lập lịch cho đầu từ FCFS, SSTF, SCAN, C-SCan, C-LOOK là gì, cho ví dụ để mô tả để tính được seek time. 15](#_Toc325060501)

[Câu 41: Có 2 dạng thiết bị đầu cuối nối tiếp (serial) và memory-mapped đều đẩy kí tự ra màn hình. Hãy giải thích tại sao thiết bị đầu cuối nối tiếp phải sử dụng ngắt trong khi thiết bị đầu cuối memory-mapped không bắt buộc. 19](#_Toc325060502)

[Câu 42: Tắc nghẽn (deadlock) và các điều kiện để xảy ra tắc nghẽn. 19](#_Toc325060503)

[Câu 43: Nguyên tắc/ phương thức chung để nâng cao hiệu năng của hệ thống. 20](#_Toc325060504)

[Câu 44: Đánh giá ngắn gọn các loại thiết bị lưu trữ; sự khác nhau căn bản giữa Đĩa cứng và Băng từ; Nguyên tắc tổ chức lưu trữ tệp tin trên 2 loại thiết bị lưu trữ này. 20](#_Toc325060505)

[Câu 45: Các thuộc tính chính và các trạng thái của File các hệ điều hành khác nhau thường có? 20](#_Toc325060506)

[Câu 46: Giới thiệu cấu trúc thư mục trong việc tổ chức lưu trữ tệp tin trên đĩa. 20](#_Toc325060507)

[Câu 47: Các phương pháp được sử dụng để quản lí, cấp phát các block đĩa cho các File. 20](#_Toc325060508)

[Câu 48: Tại sao ta có thể nói “Hệ điều hành thực hiện Vào/Ra”. Hãy giải thích cơ chế Chương trình ứng dụng thực hiện I/O như thế nào? 20](#_Toc325060509)

[Câu 49: Phân biệt Device, Driver và Device Controller. 20](#_Toc325060510)

[Câu 50: Giải thích khái niệm Đoạn găng (Critical Section). Nêu ngắn gọn biện pháp giải pháp điều phối tiến trình qua đoạn găng. 20](#_Toc325060511)

# Câu 1: Ý nghĩa của System Call. Hãy giải thích bình thường các phần mềm ứng dụng “khai thác” phần cứng của máy như thế nào?

Trả lời:

1. Định nghĩa về System call:Các lời gọi hệ thống cung cấp các giao diện giữa hệ điều hành và một tiến trình đang chạy. Các lời gọi này thường là chỉ thị hợp ngữ. Những hệ thống xác định cho phép lời gọi hệ thống được thực hiện trực tiếp từ một chương trình ngỗn ngữ cấp cao.

2. Ý nghĩa của System call:

- Tiết kiệm thời gian cho người sử dụng . Người dùng có thể gõ các lệnh hệ thống để tìm các thông tin mình cần một cách nhanh chóng

- Giúp cho hệ điều hành quản lý tốt các thông tin trong máy

- Giúp cho các phần mềm ứng dụng khai thác được triệt để phần cứng của máy tính

- Nó làm cho người dùng có một cái nhìn hoàn toàn mới mẻ về các thông tin mà mình cần,à các thông tin được lấy ra thì có thể theo ý của người dùng cần mà không phải mất nhiều thời gian nhìn từ bảng này sang các bảng khác để kết nối thông tin mà mình cần trên cùng một bảng

3. Các phần mềm khai thác phần cứng của máy như sau: bình thường các phần mềm khai thác phần cứng thông qua sự quản lý và điều phối của hệ điều hành.Khi đó các phần mềm ứng dụng khai thác phần cứng máy tính thông qua các lời gọi hệ thống nó được che dấu dưới các hàm được sử dụng trực tiếp trong các ngôn ngữ lập trình. Khi phần mềm ứng dụng được khởi động, các chức năng của nó được thực hiện trên máy thông qua rất nhiều lời gọi hệ thống.

//Ví dụ khi chúng ta muốn biết được thời gian máy tính khởi động hết bao nhiêu giây , máy tính của mình sử dụng RAM loại nào , CPU loại nào , tốc độ xử lý là bao nhiêu v.v... thi khi chúng ta bật máy tính lên thì lúc này việc nhập dữ liệu chính là thời gian khi mà máy tính được bật và các linh kiên của máy tính của mình thì đã có sẵn trong máy tính khi chúng ta lắp các thiết bị đó lên máy thì nó thay cho việc nhập dữ liệu của chúng ta . Chúng ta muốn biết thì khi đó chúng ta chỉ việc làm như sau :

Mở hộp thoại Run sau đó gõ cmd khi đó màn hình dos sẽ hiện lên và chúng ta sẽ gõ chữ systeminfo thế là hệ thống sẽ cho chúng ta biết là thời gian khởi động máy tính là bao nhiêu lâu và cấu hình máy như thế nào v.v...

      Trong các ngôn ngữ bậc cao thì người lập trình đã lập sẵn các câu lệnh để gọi các chương trình dữ liệu khi mà chúng ta đã nhập dữ liệu vào đó , người sử dụng không cần quan tâm đến chi tiết mà chỉ cần thông qua các hàm hay các lệnh để thực hiện . Lời gọi hệ thống có thể diễn ra theo một cách khác . Kiểu và khối lượng thông tin tùy thuộc vào hệ thống và lúc gọi

Ví dụ : Khi học môn SQL server 2000 thì khi đó ta có thể nhập dữ liệu vào  trong máy bằng  cách nhập bình thường và có thể sử dụng các câu lệnh như “ insert into “ để chèn dữ liệu vào máy v.v... và chúng ta có thể gọi dữ liệu bằng câu lệnh “ select      from     where “ Và dữ liệu khi được gọi ra nó có thể liên kết với các bảng khác để cho ta một cái nhìn hoàn toàn đầy đủ mà có khi ở một bảng thì không có đủ những thứ mà ta cần đó chính là cái hay của hệ thống

Lời gọi hệ thống có thể được chia thành các loại : Kiểm soát tiến trình , thao tác tập tin , thao tác thiết bị , thông tin .

# Câu 2: Một tiến trình yêu cầu Hệ điều hành chuyển trạng thái cho nó “ngủ” 5 giây. Hệ điều hành có đảm bảo được là sẽ kích hoạt tiến trình này đúng 5 giây sau đó không? Tại sao?

Trả lời

    Khi một tiến trình yêu cầu hệ diều hành cho nó chuyển trạng thái 5 giây thì hệ điều hành sẽ cho phép nó được nghỉ nhưng hệ điều hành sẽ không đảm bảo sẽ kích hoạt nó sau đúng 5 giây được vì khi tiến trình đó nghỉ thì hệ điều hành sẽ cho phép tiến trình khác đang được đứng trong hàng đợi được thực hiện và thời gian thực hiện của tiến trình này phải là khoảng thời gian nhỏ nhất trong tất cả các tiến trình đang đứng trong hàng đợi, khi đó sẽ có 3 khả năng xảy ra:

- Nếu khoảng thời gian này vừa đúng 5 giây thì hệ điều hành sẽ cho phép tiến trình mà lúc trước đã yêu cầu hệ điều hành cho phép nó ngủ làm việc tiếp vì khi đó khoảng thời gian mà tiến trình đó thực hiện sẽ là nhỏ nhất trong tất cả các tiến trình đang đợi trong hàng đợi.

- Nếu khoảng thời gian thực hiện của tiến trình này mà vượt quá 5 giây thì tiến trình đang ngủ 5 giây sẽ không thể thực hiện được khi mà tiến trình đang được thực hiện vẫn chưa thực hiên xong cho nên vì thế mà hệ điều hành trong trường hợp này không thể đảm bảo việc cho tiến trình đó khởi động lại sau đúng 5 giây

- Nếu khoảng thời gian thực hiện của tiến trình này mà nhỏ hơn 5 giây thì khi đó hệ điều hành sẽ cho phép một tiến trình khác được thực hiện (đây chính là tiến trình mà có thời gian thực hiên lớn hơn tiến trình vừa mới thực hiện nhưng mà lại là tiến trình có khoảng thời gian nhỏ hơn các tiến trình còn lại trong hàng đợi trừ tiến trình đang được ngủ 5 giây ) .Vì vậy trong trường hợp này thì hệ điều hành cũng không thể cho phép tiến trình đang được ngủ 5 giây đó thực hiện được đúng sau 5 giây

# Câu 3: Nêu định nghĩa “Hệ điều hành”. Hệ điều hành thực hiện những chức năng chính gì?

Trả lời

Hệ điều hành là một chương trình quản lý phần cứng của máy tính. Nó cung cấp nền tảng cho những người sử dụng và nó đóng vai trò trung gian giao tiếp giữa người dùng máy tính và phần cứng của máy tính. Mục tiêu của hệ điều hành là cung cấp một môi trường để người sử dụng có thể thi hành các chương trình. Nó làm cho máy tính dể sử dụng hơn, thuận lợi hơn và hiệu quả hơn. Bên cạnh đó hệ điều hành còn cung cấp các ứng dụng cơ bản và là môi trường hoạt động cho các ứng dụng tương thích.

Hệ điều hành là một thành phần quan trọng trong hầu hết mọi hệ thống máy tính. Một hệ thống máy tính có thể được chia thành bốn thành phần: phần cứng, hệ điều hành, các chương trình ứng dụng và người dùng.

Chức năng chính của hệ điều hành là :

- Hệ điều hành điều khiển và phối hợp việc sử dụng phần cứng cho những chương trình ứng dụng khác nhau của nhiều người sử dụng khác nhau

- Hệ điều hành cung cấp một môi trường để các chương trình có thể làm việc hiệu quả trên đó .

- Hệ điều hành quản lý và phân phối tài nguyên của máy tính: Các tài nguyên của máy tính như là thời gian sử dụng CPU, vùng bộ nhớ, vùng lưu trữ tập tin, thiết bị nhập xuất v.v... Hệ điều hành hoạt động như một bộ quản lý các tài nguyên và phân phối chúng cho các chương trình và người sử dụng khi cần thiết .

- Quản trị phân phối công việc cho hệ thống thiết bị để hệ thống thiết bị hoạt động hiệu quả nhất .

- Phục vụ nhu cầu đa dạng của người sử dụng một cách tốt nhất

# Câu 4: Multi-tasking là gì? Giải thích sự khác nhau giữa các hình thức Multi-tasking (cooperative and pre-emtive multi-tasking).

Trả lời

Multitasking là khả năng thực hiện nhiều nhiệm vụ cùng lúc, một nhiệm vụ là một chương trình. Những thuật ngữ đa nhiệm và đa xử lý thường được sử dụng thay thế cho nhau được. Hay ta cũng có thể nói Multitasking là một phần mềm làm nhiệm vụ phân chia thời gian phục vụ của CPU. Cho các tác vụ (tasks hay processes) để đảm bảo các tác vụ này hoàn thành một công việc nào đó hệ đa chương nhưng thời gian mỗi lần chuyển đổi diễn ra rất nhanh.

***Sự khác nhau*** giữa các hình thức Multi-tasking (cooperative and pre-emtive multi-tasking ):

- Trong mô hình đa nhiệm hợp tác(Cooperative Multitasking), chỉ có tiến trình đang sở hữu processor mới quyết định khi nào trả lại processor cho tiến trình khác hoạt động.

- Trong mô hình đa nhiệm ưu tiên(Preemptive Multitasking), thì việc chuyển processor từ tiến trình hiện tại cho tiến trình khác được thực hiện bởi bộ phận lập lịch của hệ điều hành. Bộ phận lập lịch của hệ điều hành quyết định thời gian mà mỗi tiến trình được sở hữu processor. Bộ phận lập lịch thường dựa vào độ ưu tiên của tiến trình để quyết định việc cấp processor cho nó.

# Câu 5: Sự giống nhau và khác nhau giữa Chương trình, Tiến trình và Luồng (Program, Process and Thread).

Trả lời

Chương Trình, tiến trình, luồng:

- Chương trình : Một chương trình là một thực thể thụ động chứa đựng các chỉ thị điều khiển máy tính để tiến hành một tác vụ nào đó. Chương trình có thể là một đoạn lệnh để thực thi về một vấn đề. Trong chương trình chính có thể có các chương trình con mà mỗi chương trình con lại thực hiện 1 nhiệm vụ riêng.

- Tiến trình: Là một thực thể chủ động, trong máy tính và đã được nạp vào bộ nhớ chính. Là một thực thể chủ động của một chương trình, có thời gian sống nhất định. Có nhiều trạng thái và có thể chuyển từ trạng thái này sang trạng thái khác

- Luồng : là một đơn vị cơ bản của sự sử dụng CPU , là một dòng điều khiển trong một tiến trình. Nếu tiến trình có nhiều luồng, nó có thể thực hiện nhiều tác vụ tại một thời điểm.các luồng trong một tiến trình chia sẻ với nhau đoạn mã, đoạn dữ liệu và các tài nguyên hệ thống khác như các tệp mở, các tín hiệu.

Giống nhau:Tiến trình, chương trình và Luồng giống nhau là cùng nói về việc xử lý các dữ liệu và chúng có mối liên hệ chặt chẽ với nhau mỗi thứ làm một vấn đề để khai thác dữ liệu và chúng cùng được sử lý bởi CPU.

Khác nhau là:

- Chương trình là một thể thụ động, như nội dung của các  tập tin được lưu trên đĩa, chứa đựng các chỉ thị để điều khiên máy để tiến hành các tác vụ nào đó khi cho thực hiên các chỉ thị này chương trình sẽ chuyển thành tiến trình

- Tiến trình là một thể chủ động với con trỏ lệnh hay bộ đếm chương trình sẽ xác định chỉ thị lệnh tiếp theo sẽ thực thi và kèm theo các tài nguyên phục vụ cho hoạt động của tiến trình. Các tiến trình hoàn toàn độc lập với nhau chỉ có thể liên lạc thông qua các cơ chế thông tin giữa các tiến trình mà hệ điều hành cung cấp.có thời gian sống nhất định. Có nhiều trạng thái và có thể chuyển từ trạng thái này sang trạng thái khác

- Luồng nó chỉ là một phần nhỏ trong tiến trình nó chỉ là các đoạn lệnh được nằm trong tiến trình. các luồng nằm trong cùng một tiến trình lại chia sẻ một không gian địa chỉ chung điều này có nghĩa là các luồng có thể chia sẻ các biến toàn cục của tiến trình. Một luồng có thể truy xuất dến cả các stack của những luồng khác trong cùng tiến trình.

# Câu 7: Hãy giải thích cơ chế CPU chuyển từ việc thực hiện process này sang process khác. Làm sao có thể đảm bảo được việc thực hiện đúng các lệnh của process tương ứng?

Trả lời

Cơ chế CPU chuyển từ việc thực hiện process này sang process khác.

Nói chung CPU sẽ cất giữ toàn bộ trạng thái của tiến trình đang thực hiện đưa tiến trình về trạng thái blocked, sau khi thực hiện xong tiến trình vừa yêu cầu CPU sẽ khôi phục lại hoàn toàn trạng thái của tiến trình được cất giữ. Theo sơ đồ sau:

Proess 0 Hệ điều hành process1

Ngắt hoặc lời gọi hệ thống

CÊt gi÷ tr¹ng th¸i trong PCB0

Nghỉ

Kh«i phôc tr¹ng th¸i tõ PCB1

Hoạt động

Ngắt hoặc lời gọi hệ thống

CÊt gi÷ tr¹ng th¸i trong PCB1

Nghỉ

Kh«i phôc tr¹ng th¸i tõ PCB0

Hoạt động

Dựa vào hình vẽ ta giải thích cơ chế chuyển từ tiến trình này sang tiến trình khác.

Giả sử có 2 tiến trình P0, P1. P0 đang chạy thì xuất hiện ngắt và P0 dừng lại, khi đó toàn bộ thông tin về P0 từ các thanh ghi được ghi ra thực thể PCB0 trong bộ nhớ chính. Sau đó các thông tin của P1 từ PCB1 trong bộ nhớ chính được nạp vào các thanh ghi và thực thi P1, P1 đang thực thi khi xuất hiện ngắt hay lời gọi hệ thống P1 dừng lại khi đó toàn bộ các thông tin về P1 từ các thanh ghi được ghi ra thực thể PCB1 trong bộ nhớ chính. Sau đó các thông tin của P0 từ PCB0 trong bộ nhớ chính được load lên nạp vào các thanh ghi.

Cơ chế:

- CPU lưu tất cả các thông tin về tiến trình được dừng lại từ các thanh ghi vào một PCB nào đó trong bộ nhớ chính, và nạp lại các thông tin về một tiến trình được tiếp tục từ PCB đã lưu nó vào các thanh ghi.

- Để đảm bảo thực hiện đúng các lệnh của process tương ứng thì:

Một tiến trình gồm các thông tin về:

- Con trỏ đến các tiến trình cha, con.

- Không gian địa chỉ.

- Các tài nguyên các trạng thái của tiến trình.

- Các luồng.Một luồng lại chứa các thông tin:

+ Program counter.

+ Các thanh ghi.

+ Stack.

Khi chuyển giữa các tiến trình tất cả các thông tin này được lưu. Khi được nạp lại chúng được đưa nguyên trạng thái vào các thanh ghi, và chúng tiếp tuc được thực hiện như không có chuyện ghì xảy ra. Quan trọng quyết định là program counter bộ đếm chương trình được lưu.

# Câu 8: Mục đích và nguyên tắc của việc lập lịch cho CPU (Process Scheduling)

0Trả lời

Lý do phải lập lịch cho CPU:

- Do số người sử dụng và số lượng tiến trình luôn lớn hơn số lượng CPU của máy tính rất nhiều và tại một thời điểm, chỉ có duy nhất một tiến trình được thực hiện trên một CPU. Mà Nhu cầu người sử dụng nhiều hơn tài nguyên CPU đang có do đó ta cần lập lịch để phân phối thời gian sử dụng CPU cho các tiến trình của người sử dụng và hệ thống.

- Khi một máy tính đa chương trình, tức là có nhiều tiến trình cần phải xử lý tại một thời điểm. Thì khi có nhiều hơn một tiến trình ở trạng thái sẵn sàng và chỉ có 1 CPU thì lúc này hệ điều hành phải quyết định tiến trình nào sẽ được chạy trước, đó chính là khái niệm lập lịch CPU.

Với mỗi hệ thống khác nhau, việc lập lịch cho CPU đáp ứng những mục đích khác nhau. Nhưng nhìn chung, việc lập lịch cho CPU nhằm đạt được những mục đích sau:

* Tối ưu hoá việc sử dụng CPU. Đây là trọng tâm của việc lập lịch, bởi vì, CPU là một tài nguyên vô dùng quí giá. Trong hệ máy tính đơn giản thì CPU sẽ rảnh rỗi trong quá trình nhập/xuất; tất cả thời gian chờ này là lãng phí. Vì vây, ta phải cố gắng tận dụng thời gian này để CPU có thể phục vụ cho quá trình khác.
* Làm tăng số lượng tiến trình thực hiện được trong 1 đơn vị thời gian.
* Tối thiểu hoá thời gian quay vòng, thời gian chờ và thời gian đáp ứng.
* Phối hợp cân đối Cpu thực hiện các tiến trình

Nguyên tác việc lập lịch:  
xem xét và quyết định khi nào thì dừng tiến trình hiện tại để thu hồi processor và chuyển processor cho tiến trình khác, và khi đã có được processor thì chọn tiến trình nào trong số các tiến trình ở trạng thái ready để cấp processor cho nó.

Nguyên tắc lập lịch cho Cpu là :

* CPU được sử dụng tối đa
* Sô tiến trình hoàn thành trong 1 đơn vị t là max
* Thời gian thưc hiện một tiến trình(bắt đầu -> xong) là min
* Thời gian chờ min
* Thời gian thực hiện min.

# Câu 9: Ý nghĩa căn bản của Lập lịch, Sự khác nhau giữa Lập lịch dài kì và lập lịch ngắn hạn?

Trả lời:

* Ý nghĩa: Xác định process mới nào được tiếp tục vào “sâu hơn” trong hệ thống.
* CPU – bound process có thời gian sử dụng CPU nhiều hơn thời gian sử dụng I/O.
* I/O – bound process dung phần lớn thời gian để đợi I/O.
* Long – term scheduler (lập lịch dài kỳ) còn được gọi là job scheduler lựa chọn những tiến trình nào nên được đưa từ đĩa vào trong hàng đợi ready được sử dụng đến rất không thường xuyên (seconds, minutes), dùng trong các hệ xử lý theo lô, chậm, ảnh hưởng đến độ đa lập trình (số quá trình đang ở trong bộ nhớ). Nếu càng nhiều process trong bộ nhớ thì khả năng mọi process bị block có xu hướng giảm.

Sử dụng CPU hiệu quả hơn.

Nhưng mỗi process được phân chia khoảng thời gian sử dụng CPU nhỏ hơn.

Thường có xu hướng đưa vào một tập lẫn lộn các CPU – bound process và I/O – bound process.

CPU không thực hiện tiến trình nào khi 1 tiến trình nào đó kết thúc.

* Short – term scheduler (lập lịch ngắn kỳ) còn được gọi là CPU scheduler; lựa chọn tiến trình nào nên được thực hiện kế tiếp và phân phối CPU sao cho nó được sử dụng đến được thường xuyên (miliseconds), nhanh.

# Câu 10, 11, 12, 13: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện. Đánh giá thuật toán FCFS, SJF, SRF, RR

Trả lời

Các trạng thái của quá trình là:

- Khởi tạo(new): Tiến trình đang được tạo ra.

- Sẵn sằng(ready): Tiến trình chờ để kết nối với processor.

-Thực hiện(running): Các lệnh đang được thực hiện

- Chờ đợi(waiting): Tiến trình chờ một sự kiện vào/ra hoặc chờ nhận một tín hiệu nào đó.

- Kết thúc(terminated): Tiến trình kết thúc thực hiện.

Khi các tiến trình được đưa vào hệ thống, chúng được đặt vào hàng đợi công việc. Hàng đợi chứa tất cả tiến trình trong hệ thống. Các tiến trình đang nằm trong bộ nhớ chính sẳn sàng và chờ để thực thi được giữ trên một danh sách được gọi là hàng đợi sẳn sàng . Nó chờ trong hàng đợi sẳn sàng cho tới khi nó được chọn thực thi.

Ngắt

Được chấp nhận

Điều phối

Kết thúc một sự kiện Chờ đợi một sự kiện hoặc một

hoặc một tín hiệu vào/ra tín hiệu vào /ra

Một khi quá trình được gán tới CPU và đang thực thi , có các trường hợp “ready to run” để thực hiện sau:

- Quá trình có thể phát ra một yêu cầu nhập/xuất và sau đó được đặt vào trong hàng đợi nhập/xuất.

Khi một ngắt xảy ra, quá trình có thể bị buộc trả lại CPU và được đặt trở lại trong hàng đợi sẳn sàng.

- Trong cả hai trường hợp cuối cùng quá trình chuyển từ trạng thái chờ tới trạng thái sẳn sàng và sau đó đặt trở lại vào hàng đợi sẳn sàng để được thực thi.

**Các cách lựa chọn tiến trình từ “ready to run”:**

10. First come-first served (FCFS) . (vào trước phục vụ trước)

Ưu điểm:

- Đơn giản, dễ lập trình.

- Quản lý dễ dàng.

Hạn chế:

- Không áp dụng được cho hệ thống máy chủ.

- Không có sự ưu tiên (non preemtive) không công bằng với những tiến trình có thời gian phục vụ (burst time) ngắn.

- Xảy ra hiện tượng có tiến trình độc chiếm CPU trong thời gian dài.

- Thời gian chờ đợi trung bình dài.

- Không phù hợp với các hệ thống chia sẻ thời gian.

Ta xét ví dụ sau.

Xét tập hợp các tiến trình sau đến tại thời điểm 0, với chiều dài thời gian chu kỳ CPU được cho theo mini giây.

|  |  |  |  |
| --- | --- | --- | --- |
| Tiến trình | P1 | P2 | P3 |
| thời gian xử lý | 24 | 3 | 3 |

Nếu các tiến trình đến theo thứ tự P1, P2, P3 và được phục vụ theo thứ tự FCFS, chúng ta nhận được kết quả thời gian chờ để được thực hiện cảu các tiến trình như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tiến trình | P1 | P2 | P3 |
| Thời gian chờ đợi | 0 | 24 | 27 |

Thời gian chờ đợi trung bình là (0+24+27)/3=17 mili giây. Tuy nhiên, nếu các tiến trình đến theo thứ tự P2, P3, P1 thì các kết quả được hiển thị thi như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tiến trình | P1 | P2 | P3 |
| Thời gian chờ đợi | 0 | 3 | 6 |

Thời gian chờ đợi trung bình bây giờ là (6+0+3)/3=3 mili giây

11. Shorted- job-first (SJF)-(thời gian phục vụ ngắn thì thực hiện trước)

Ưu điểm:

- Tốt hơn so với FCFS vì hạn chế được việc một tiến trình độc chiếm CPU.

- Thời gian chờ đợi giảm.

- Thích hợp để có thể sử dụng cho các tiến trình tương tác ( Các tiến trình tương tác nói chung thường theo một mẫu chung đó là chờ đợi lệnh, kích hoạt lệnh, rồi lại đợi lệnh, kích hoạt lệnh...)

Hạn chế:

- Phải mất thời gian cho việc ước lượng về mặt thời gian để lựa chọn tiến trình.

- Không biết chiều dài của chu kì CPU kế tiếp( Khó khăn trong việc xác định tiến trình nào trong số các tiến trình đang có thể chạy được xem là có thời gian phục vụ ngắn nhất).

Ta xem xét ví dụ sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tiến trình | P1 | P2 | P3 | P4 |
| thời gian xử lý | 6 | 8 | 7 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tiến trình | P4 | P1 | P3 | P2 |
| thời gian chờ đợi | 0 | 3 | 9 | 16 |

Do đó, thời gian chờ đợi trung bình là (3+16+9+0)/4 = 7 mili giây.

13. Round – robin scheduling (RR - Định thời luân phiên )

- Mỗi tiến trình được gán một khoảng thời gian nội bộ gọi là quantum. Đó chính là thời gian tối đa cho phép nó chiếm CPU để chạy.

Ưu điểm:

- Đây là giải thuật đơn giản, công bằng và phổ dụng.

- Tránh bị rôi vào vòng lặp vô hạn.

- Xóa bỏ việc độc chiếm CPU trong thời gian dài.

Hạn chế:

- Khó xác định khoảng thời gian chia cho mỗi tiến trình là bao lâu.

- Mất thời gian để chuyển, bật, tắt các tiến trình.

- Nếu khoảng thời giân chia cho các tiến trình không hợp lý sẽ lãng phí thời gian.

12. Prỉoity scheduling ( Lập lịch ưu tiên):

- Mỗi tiến trình được gán một độ ưu tiên và các tiến trình đang ở trạng thái có thể chạy tiến trình nào có độ ưu tiên cao nhất sẽ được chạy. Đồng thời để tránh các tiến trình có độ ưu tiên cao chạy vô định - khi các tiến trình này chạy cứ ứng với mỗi khoảng thời gian nhất định độ ưư tiên của nó sẽ giảm đi với một mức. khi chạy một khoảng thời gian nhất định độ ưu tiên của nó thấp hơn tiến trình khác thì thao tác chuyển bật tiến trình sẽ diễn ra.

Ưu điểm:

- Tiết kiệm thời gian hơn round robin (định thời luân phiên) do xác định được tiến trình nào quan trọng cho thực hiện trước.

- Mức độ ưư tiên có thể thay đổi tùy theo chức năng nhiệm vụ và yêu cầu của từng cơ quan từng người sử dụng.

# Câu 14: Giải thích tại sao phần địa chỉ cao (top-half) của hệ điều hành không phải là tiến trình (process). Giải thích tại sao phần địa chỉ thấp (bottom-half) của hệ điều hành cũng không phải là tiến trình (process).

Trả lời

Top - half: được dùng trong quản lý các tiến trình chứ không phải là các tiến trình. Thực hiện các yêu cầu của I/O.

Bottom – half: Là các interrupt handle là các hàm, các thủ tục phục vụ ngắt => dùng để hỗ trợ, cộng tác với các process.

# 

# Câu 15: Giải thích ngắn gọn trạng thái của các tiến trình, các mô hình tiến trình hệ thống (process models) chung.

Trả lời:

- Tại một thời điểm xác định chỉ có duy nhất một tiến trình ở trạng thái Running.

- Mới tạo: tiến trình đang được tạo lập.

- Trạng thái Ready: ngay sau khi khởi tạo tiến trình, đưa tiến trình vào hệ thống và cấp phát đầy đủ tài nguyên (trừ processor) cho tiến trình, hệ điều hành đưa tiến trình vào trạng thái Ready. Hay nói cách khác, trạng thái Ready là trạng thái của một tiến trình trong hệ thống đang chờ được cấp processor để bắt đầu thực hiện.

- Trạng thái running: là trạng thái mà tiến trình đang được sở hữu processor để hoạt động, hay nói cách khác là các chỉ thị của tiến trình đang được thực hiên, xử lý bởi processor.

- Trạng tháo Blocked (khóa): là trạng thái mà tiến trình đang chờ để được cấp phát thêm tài nguyên, để một sự kiện nào đó xảy ra, hya một quá trình vào/ra kết thúc.

- Kết thúc: tiến trình hoàn tất xử lý.

# Câu 16: Giải thích vai trò của PCB (process control block).

Trả lời:

Khối kiểm soát tiến trình (PCB) là một cấu trúc dữ liệu được hệ điều hành duy trì cho mỗi tiến trình dùng để chứa các thông tin cần thiết về tiến trình đó. PCB chứa bản sao công việc của các tiến trình. Nó lưu lại trạng thái của tiến trình tại thời điểm đang xét. Trạng thái này bao gồm các địa chỉ liên kết của tiến trình và dữ liệu, số hiệu tiến trình, bộ đếm lệnh, nội dung các thành ghi, giới hạn bộ nhớ của tiến trình,...

- Nó có tác dung lưu lại trạng thái của tiến trình tại thời điểm đang xét (thường xảy ra khi luân chuyển CPU) – trạng thái của tiến trình này bao gồm: các địa chỉ liên kết của tiến trình và dữ liệu, số hiệu tiến trình, bộ đếm lệnh, nội dung các thanh ghi, giới hạn bộ nhớ của tiến trình, danh sách các file đang mở, các thông tin về thời gian thống kê tiến trình hiện tại.

# Câu 17: Giới thiệu và giải thích ngắn gọn các thành phần (components) của Hệ điều hành.

Trả lời:

Các thành phần của hệ điều hành:

a. Hệ thống quản lý tiến trình:

- Tạo lập, hủy bỏ 1 tiến trình.

- Tạm dừng, tái kích hoạt một tiến trình.

- Cung cấp các cơ chế trao đổi thong tin giữa các tiến trình.

- Cung cấp cơ chế đồng bộ hóa các tiến trình.

b. Hệ thống quản lý bộ nhớ:

- Cấp phát và thu hồi vùng nhớ cho các tiến trình khi cần thiết.

- Ghi nhận tình trạng bộ nhớ chính: vùng đã cấp phát, vùng còn có thể sử dụng.

- Quyết định tiến trình nào được nạp vào bộ nhớ chính khi có một vùng nhớ trống.

c. Hệ thống quản lý nhập/xuất:

- gửi các lệnh điều khiển đến các thiết bị.

- Tiếp nhận các ngắt.

- Xử lý lỗi.

d. Hệ thống quản lý tệp tin:

- Tạo lập, hủy bỏ một tập tin hoặc một thư mục.

- Cung cấp các thao tác xử lý tập tin và thư mục.

- Tạo lập quan hệ tương ứng giữa tập tin và bộ nhớ phụ chứa nó.

e. Hệ thống bảo vệ:

xây dựng cơ chế bảo vệ thích hợp (trong trường hợp nhiều người cùng sử dụng đông thời các tiến trình).

f. Hệ thống dịch lệnh (shell)

- Đóng vai trò giao diện giữa người sử dụng và hệ điều hành

- Các lệnh được chuyển đến hệ điều hành dưới dạng chỉ thị điều khiển

- Shell nhận lệnh và thông dịch lệnh để hệ điều hành có xử lý tương ứng.

g. Quản lý mạng:

- Một hệ thống phân bố nhiều bộ xử lý với các bộ nhớ độc lập.

- Các tiến trình trong hệ thống có thể kết nối với nhau qua mạng truyền thông

- Việc truy xuất đến tài nguyên mạng thông qua các trình điều khiển giao tiếp mạng.

# Câu 18: Nêu vắn tắt các kĩ thuật cấp phát bộ nhớ (nạp chương trình vào bộ nhớ).

Trả lời:

a. Kỹ thuật phân vùng cố định (fixed partitioning)

Bộ nhớ được chia thành n phần, kích thước các phần không nhất thiết phải bằng nhau, mỗi phần sử dụng một bộ nhớ độc lập. Mỗi phần có thể nạp một chương trình và tổ chức thực hiện đồng thời. Vì mỗi phần được coi là một bộ nhớ độc lập, nên các chương trình sẽ có một danh sách quản lý không gian nhớ tự do riêng. Chương trình được nạp vào phần nào thì sẽ ở đó cho đến khi kết thúc.

- Ưu điểm: đơn giản, dễ tổ chức, giảm thời gian tìm kiếm.

- Nhược điểm: lãng phí bộ nhớ và xảy ra hiện tượng phân mảnh nội vi.

b. Kỹ thuật phân vùng động (dynamic partitioning)

- Bộ nhớ có một bảng quản lý không gian nhớ tự do thống nhất. Khi thực hiện chương trình, hệ thống dựa vào kích thước chương trình để phân bố không gian nhớ thích hợp tạo thành một vùng nhớ độc lập và tạo bảng quản lý riêng. Khi các chương trình kết thúc, bộ nhớ dành cho chương trình sẽ bị thu hồi.

-Ưu điểm: tận dụng không gian nhớ tự do.

- Nhược điểm: không gây ra hiện tượng phân mảnh nội vi nhưng lại xuất hiện phân mảnh ngoại vi .

c. Kỹ thuật phân trang:

- Phân bộ nhớ vật lý thành các khối (block) có kích thước cố định bằng nhau, gọi là khung trang. Không gian địa chỉ cũng được chia thành các khối có cùng kích thước với khung trang và được gọi là trang. Khi cần nạp một tiến trình để xử lý các trang của tiến trình sẽ được nạp vào những khung trang còn trống. Một tiến trình có kích thước N trang sẽ yêu cầu N khung trang tự do.

- Cơ chế MMU (memory management unit) là cơ chế phần cứng hỗ trợ thực hiện chuyển đổi địa chỉ trong cơ chế phân trang gọi là bảng trang. Mỗi phần tử trong bảng trang cho biết địa chỉ bắt đầu của vị trí lưu trữ trang tương ứng trong bộ nhớ vật lý.

- Mỗi địa chỉ phát sinh bới CPU chia làm 2 phần:

+ Số hiệu trang (p)

+ Địa chỉ tương ứng trang trang (d)

+ Địa chỉ vật lý của trang = d+ địa chỉ bắt đầu của trang

# Câu 19. Trong kỹ thuật cấp phát bộ nhớ phân vùng động, khi nạp tiến trình mới vào bộ nhớ cần lựa chọn một vùng còn “ rỗng” nào đó. Nêu các phương pháp/thuật toán lựa chọn.

Trả lời:

Khi có một tiến trình cần được nạp vào bộ nhớ mà trong bộ nhớ có nhiều hơn một khối nhớ trống (free block) có kích thước lớn hơn kích thước của tiến trình đó, thì hệ đièu hành phải quyết định chọn một khối nhớ trống có kích thước phù hợp để nạp tiến trình sao cho việc lựa chọn này dẫn đến việc sử dụng bộ nhớ chính hiệu quả nhất. Có 3 thuật toán được sử dụng: best-fit, first-fit và worst-fit. cả 3 thuật toán này phải chọn 1 khối nhớ trống có kích thước bằng hoặc lớn hơn kích thước của tiến trình cần nạp vào nhưng nó có điểm khác nhau cơ bản sau:

- Best-fit: chọn khối nhớ có kích thước nhỏ nhất trong các khối nhớ rỗng có kích thước lớn hơn kích thước tiến trình cần nạp vào

- First-fit: chọn khối nhớ đầu tiên có kích thước phù hợp với tiến trình cần nạp vào.

- Worst-fit: chọn khối nhớ lớn nhất trong số các khối nhớ phù hợp với kích thước tiến trình cần nạp

# Câu 20: Phân tích hai khái niệm Page và Paging (trang và phân trang).

Trả lời

Page là một đơn vị trong không gian địa chỉ ảo. Hệ điều hành phải lưu trữ một bảng các trang để ánh xạ từ các trang tương ứng với các khung trong bộ nhớ vật lý.

Paging là việc phân trang, tức là phân chia không gian địa chỉ ảo thành các trang. Không gian địa chỉ ảo sẽ được chia thành các trang có kích thước bằng nhau.

# Câu 21: Giải thích cơ chế phân đoạn (segmentation) trong quản lý bộ nhớ của hệ điều hành.

Trả lời

Dùng cơ chế phân đoạn để quản lý bộ nhớ có hỗ trợ user view.

- Không gian địa chỉ ảo là 1 tập các đoạn, mỗi đoạn có tên và kích thước riêng.

- Một địa chỉ luận lý được định vị bằng tên đoạn và độ dời (offset) bên trong đoạn đó (so sánh với phân trang)

Hiện thực phân đoạn:

Địa chỉ luận lý là 1 cặp giá trị (segment, offset)

Bảng phân đoạn (segment table): gồm nhiều mục, mỗi mục chưa:

- Base: chứa địa chỉ khởi đầu của segment trong bộ nhớ.

- limit: xác định kích thước segment

Segment table base register (STBR): trỏ đến vị trí bảng phân đoạn trong bộ nhớ.

Segment table length register (STLR): số lượng segment của chương trình.

Một chỉ số segment s là hợp lệ nếu s<STLR

# Câu 22: Mô tả ngắn gọn “Bộ nhớ ảo” – Virtual memory. Bộ nhớ ảo có lợi và thiệt hại gì đối với việc tối ưu hoá sử dụng CPU.

Trả lời

Nhiều năm trước đây người ta thường phải đối mặt với nhiều chương trình lớn vượt quá khả năng bộ nhớ. Giải pháp đưa ra là sử dụng bộ nhớ ảo. Ý tưởng là kích thước bao gồm chương trình, dữ liệu, và ngăn xếp có thể vượt quá khả năng của bộ nhớ chính, do đó hệ điều hành phải giữ lại các phần trên của chương trình hiện thời trong bộ nhớ chính, phần còn lại sẽ được để lại trên đĩa.

Khi xử lý một tiến trình lớn ta phải tải cả tiến trình vào bộ nhớ nhưng nếu sử dụng bộ nhớ ảo thì chỉ phải tải những phần cần thiết cho nên sẽ giảm được các thao tác vào ra, tăng tốc độ xử lý của CPU, đáp ứng nhanh hơn.

Nhưng khi sử dụng bộ nhớ ảo thì CPU phải thực hiện việc ánh xạ từ địa chỉ logic sang địa chỉ vật lý.

# Câu 23: Trong kỹ thuật bộ nhớ ảo thường sử dụng Phân trang theo yêu cầu (demand paging). Hãy giới thiệu cơ chế này.

Trả lời

Một hệ thống phân trang theo yêu cầu là hệ thống sử dụng kỹ thuật phân trang kết hợp với kỹ thuật swapping. Một tiến trình được xem như một tập các trang, thường trú trên bộ nhớ phụ (thường là đĩa). Khi cần xử lý, tiến trình sẽ được nạp vào bộ nhớ chính. Nhưng thay vì nạp toàn bộ chương trình, chỉ những trang cần thiết trong thời điểm hiện tại mới được nạp vào bộ nhớ. Như vậy một trang chỉ được nạp vào bộ nhớ chính khi có yêu cầu.

Với mô hình này, cần cung cấp một cơ chế phần cứng giúp phân biệt các trang đang ở trong bộ nhớ chính và các trang trên đĩa. Có thể sử dụng lại bit*valid-invalid*nhưng với ngữ nghĩa mới:

*valid*: trang tương ứng là hợp lệ và đang ở trong bộ nhớ chính .

*Invalid*: hoặc trang bất hợp lệ (không thuộc về không gian địa chỉ của tiến trình) hoặc trang hợp lệ nhưng đang được lưu trên bộ nhớ phụ.

Một phần tử trong bảng trang mộ tả cho một trang không nằm trong bộ nhớ chính, sẽ được đánh dấu*invalid* và chứa địa chỉ của trang trên bộ nhớ phụ.

# Câu 24: Trong kỹ thuật sử dụng Bộ nhớ ảo Hệ điều hành cần có bộ phận quản lý việc hoán chuyển các trang/đoạn giữa bộ nhớ thực và bộ nhớ ảo và có thể xảy ra “Lỗi trang” (page fault). Hệ điều hành giải quyết vấn đề đó như thế nào và mục tiêu cần đạt được?

Trả lời

Khi nhận được tín hiệu lỗi trang, hệ điều hành phải tạm dừng tiến trình hiện tại để tiến hành việc xử lý lỗi trang. Khi xử lý lỗi trang hệ điều hành có thể gặp một trong hai tình huống sau:

- Hệ thống còn frame trống : Hệ điều hành sẽ thực hiện các bước sau:

+ Tìm vị trí của page cần truy xuất trên đĩa.

+ Nạp page vừa tìm thấy vào bộ nhớ chính.

+ Cập nhật lại bảng trang (PCT) tiến trình.

+ Tái kích hoạt tiến trình để tiến trình tiếp tục hoạt động.

- Hệ thống không còn frame trống :

+ Tìm vị trí của page cần truy xuất trên đĩa.

+ Tìm một page không hoạt động hoặc không thực sự cần thiết tại thời điểm hiện tại để swap out nó ra đĩa, lấy frame trống đó để nạp page mà hệ thống vừa cần truy xuất. Page bị swap out sẽ được hệ điều hành swap in trở lại bộ nhớ tại một thời điểm thích hợp sau này.

+ Cập nhật PCT của tiến trình có page vừa bị swap out.

+ Nạp trang vừa tìm thấy ở trên (bước 1) vào frame trống ở trên (bước 2).

+ Cập nhật lại bảng trang (PCT) của tiến trình.

+ Tái kích hoạt tiến trình để tiến trình tiếp tục hoạt động.

- Khi xử lý lỗi trang, trong trường hợp hệ thống không còn frame trống hệ điều hành phải chú ý đến các vấn đề sau:

+ Nên chọn page nào trong số các page trên bộ nhớ chính để swap out: “Neo” một số page

+ Phải tránh được trường hợp hệ thống xảy ra hiện tượng “trì trệ hệ thống”

+ Đánh dấu các trang bị thay đổi

# Câu 25: Đánh giá các thuật toán thay trang (Page Replacement) trong kỹ thuật sử dụng Bộ nhớ ảo.

Trả lời

Khi không còn các trang trống để cấp phát cho tiến trình yêu cầu thì hệ điều hành phải chọn một trang để swap out ra ngoài, và chỗ trống đó sẽ được cấp cho tiến trình mới. Việc chọn trang nào để swap out sử dụng một trong các thuật toán sau đây.

Giống nhau :

+ Các thuật toán thay trang đều phải mất số lỗi trang ban đầu là như nhau. Trong ví dụ trên là số lỗi trang ban đầu là 3.

Khác nhau:

1> Thuật toán thay thế trong FIFO

Ý tưởng của thuật toán : trang được chọn là trang được sử dụng lâu nhất

Ưu điểm:

Thuật toán này đơn giản, dễ hiểu, dễ cài đặt :

Nhược điểm:

Khi thực hiện có trường hợp không cho kết quả tốt, vì : trang được chọn để thay thế có thể là trang chứa nhiều dữ liệu cần thiết, thường xuyên được sử dụng nên được nạp sớm, do vậy, khi bị chuyển ra khỏi bộ nhớ chính sẽ nhanh chóng gây ra lỗi trang

Số lượng lỗi trang xảy ra sẽ tang lên khi số lượng khung trang sử dụng tăng. Hiện tượng này gọi là nghịch lý Belady.

2>Thuật toán tối ưu :

Tư tưởng: thay thế trang mà nó không được dùng cho một khoảng thời gian lâu nhấ

+ Ưu điểm : Số lỗi trang là nhỏ nhất trong tất cảc các thuật toán

+ Khuyết điểm : Khó thực thi,khó trong việc cài đặt, chủ yếu dùng để nghiên cứu so sánh.

3>Thuật toán LRU :(least-recently-used – ít được sử dụng gần đây nhất)

+ tư tuởng thuật toán:

LRU dùng thời điểm cuối cùng trang được truy xuất – dùng quá khứ gần để dự đoán tương lai. Tức là sẽ thay thế trang mà nó không được dùng trong khoảng thời gian lâu nhất.

+| ưu điểm:

Ít xảy ra lỗi trang hơn fifo.

Thay thế tối ưu hoá và LRU không gặp phải sự nghịch lý của Belady.

+hạn chế:

+Đòi hỏi cả phần cứng hỗ trợ.

4>thuật toán xấp xỉ LRU. Và counting – based như ở dưới.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| thuật toán | FIFO | Optimal | LRU | Sấp xỉ LRU | Counting-based |
| Cách thức | Trang được chọn là trang đã sử dụng lâu nhất | Trang được chọn là trang sẽ không được sử dụng trong thời gian dài nhất | Trang được chọn là trang không được sử dụng trong khoảng thời gian dài nhất | Trang được chọn là một trang cũ và không được sử dụng | Duy trì một bộ đếm.LFU chọn các trang có số đếm nhỏ nhất.MFU sử dụng tham số dựa trên việc các trang có số đếm nhỏ nhất |
| Ưu điểm | Đơn giản, dễ hiểu, dễ lập trình | Giảm số lỗi trang xuống nhỏ nhất | Khó thực thi. | Có triển vọng | Việc thực thi là rất tốn kém. |
| Hạn chế | Hiệu quả không cao vì trang được chọn có thể là trang chứa nhiều dữ liệu cần thiết thường xuyên được sử dụng nên thường xuyên được nạp sớm khi , làm tăng số lỗi trang | Khó thực thi, chỉ dùng để nghiên cứu, giảng dạy | Có hiệu quả | Thuật toán luôn kết thúc | Hiệu quả không cao |

# Câu 26: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán FIFO.

Trả lời

- Thuật toán FIFO là thuật toán đơn giản và dễ cài đặt nhất. Với thuật toán này thì trang mà hệ điều hành chọn để swap out là trang được đưa vào bộ nhớ sớm nhất, hay ở trong bộ nhớ lâu nhất

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 |
| 7 | 7 | 7 | ***2*** | | | 2 | 2 | ***4*** | 4 | 4 | ***0*** | | | | | 0 | 0 | | | | |
|  | 0 | 0 | 0 | | | ***3*** | 3 | 3 | ***2*** | 2 | 2 | | | | | ***1*** | 1 | | | | |
|  |  | 1 | 1 | | | 1 | ***0*** | 0 | 0 | ***3*** | 3 | | | | | 3 | ***2*** | | | | |

Số lỗi trang: 12

# Câu 27: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán Tối ưu OPT.

Trả lời

- Thay thế trang sẽ không được sử dụng lâu nhất

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 |
| 7 | 7 | 7 | ***2*** | | | 2 | | | 2 | | | | | 2 | | | | | 2 | | | | | | |
|  | 0 | 0 | 0 | | | 0 | | | ***4*** | | | | | ***0*** | | | | | 0 | | | | | | |
|  |  | 1 | 1 | | | ***3*** | | | 3 | | | | | 3 | | | | | ***1*** | | | | | | |

Số lỗi trang: 8

# Câu 28: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán LRU.

Trả lời:

- Theo thuật toán này thì trang được hệ điều hành chọn để thay thế là trang có khoảng thời gian từ lúc nó được truy xuất gần đây nhất đến thời điểm hiện tại là dài nhất, so với các trang đang ở trên bộ nhớ chính, tức là thay thế trang tồn tại lâu nhất trong bộ nhớ chính tính đến thời điểm đang xét. Như vậy trong trường hợp này hệ điều hành phải ghi nhận thời điểm cuối cùng trang được truy xuất.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 0 | 1 | 2 | 0 | 3 | 0 | 4 | 2 | 3 | 0 | 3 | 2 | 1 | 2 | 0 | 1 |
| 7 | 7 | 7 | ***2*** | | | 2 | | | ***4*** | 4 | 4 | ***0*** | | | | | ***1*** | | | 1 | | |
|  | 0 | 0 | 0 | | | 0 | | | 3 | ***2*** | 2 | 2 | | | | | 2 | | | 2 | | |
|  |  | 1 | 1 | | | ***3*** | | | 0 | 0 | ***3*** | 3 | | | | | 3 | | | ***0*** | | |

Số lỗi trang: 11

# Câu 29: Phân biệt hai hiện tượng phân mảnh nội (internal fragmentation) và phân mảnh ngoài (external fragmentation), chúng xuất hiện khi nào và tại sao?

Trả lời

Hiện tượng phân mảnh ngoại là hiện tượng không gian bộ nhớ cung cấp cho một tiến trình là không liên tục. Hiện tượng này xảy ra trong việc phân vùng, phân đoạn. Việc phân vùng đặc biệt trong phân vùng tĩnh, sẽ tạo ra các lỗ trống ở rải rác trong bộ nhớ. Có thể giảm sự phân mảnh ngoại bằng việc sắp xếp lại toàn bộ các vùng nhớ để gộp các lỗ trống rải rác đó thành một khối lớn hơn. Vì thế toàn bộ không gian nhớ trong một yêu cầu nào đó sẽ nằm rải rác. Trong phân đoạn thì một chương trình được chia thành nhiều đoạn như : chương trình chính, đoạn ngăn xếp, đoạn thủ tục…sau một thời gian chạy thì bộ nhớ sẽ chứa các đoạn và các lỗ hổng.

Hiện tượng phân mảnh nội là hiện tượng không gian bộ nhớ được cấp phát lớn hơn kích thước mà nó yêu cầu. Phần kích thước sai khác đó có thể coi là một phân vùng nhưng không được sử dụng. Xảy ra trong phân vùng động, phân trang.

# Câu 30: Giải thích khái niệm “hoán chuyển” (swapping); Ý nghĩa, ứng dựng của việc áp dụng kĩ thuật này.

Trả lời:

Một process có thể tạm thời bị swap ra khỏi bộ nhớ chính và lưu trên một hệ thống lưu trữ phụ. Sau đó, process có thể được nạp lại vào bộ nhớ để tiếp tục quá trình thực thi. Cơ chế này được gọi là hoán vị ( swapping).

Swapping policy: 2 ví dụ:

- Round-robin: swap out P1 (vừa tiêu thụ hết quantum của nó), swap in P2, thực thi P3,...

- Roll out, Roll in: dùng trong cơ chế định thời theo độ ưu tiên (priority-based scheduling)

Process có độ ưu tiên thấp hơn sẽ bị swap out nhương chỗ cho process có độ ưu tiên cao hơn mới đến được nạp vào bộ nhớ để thực thi.

# Câu 31: Hãy giải thích vì sao DMA thường được “ưu ái” sử dụng như là phương thức thực thi trao đổi với ngoại vi? DMA có lợi điểm gì không đối với input thông qua bàn phím? Giải thích tại sao hoặc tại sao không?

Trả lời

Hệ điều hành trao đổi với ngoại vi bằng ba cách : polling (hỏi vòng), interrupt (ngắt), DMA (truy cập trực tiếp bộ nhớ). Trong đó DMA (direct memory access) thường được “ưu ái” sử dụng như là phương pháp thực thi trao đổi với ngoại vi vì đây là phương pháp mà các thiết bị ngoại vi trao đổi trực tiếp với main memory mà không cần sự can thiệp của CPU, cho nên tiêu tốn ít thời gian hơn, nhanh hơn rất nhiều so với hai phương pháp trên.

Nếu không sử dụng DMA thì khi có một yêu cầu đọc dữ liệu từ disk vào bộ nhớ, quá trình xử lý diễn ra như sau: disk controller sẽ đọc từng khối dữ liệu (block) từ disk rồi chuyển vào vùng đệm của disk controller. Tại đây dữ liệu sẽ được kiểm tra để chắc rằng quá trình đọc không gây ra lỗi. Sau đó khi CPU nhận được ngắt yêu cầu, nó sẽ sao chép từng block từ vùng đệm của disk controller vào main memory.

Nếu sử dụng DMA ,tức là hệ thống phải có DMA controller. DMA controller truy cập trực tiếp vào bus hệ thống độc lập với CPU. Khi ngoại vi yêu cầu sử dụng DMA thì DMA controller sẽ độc chiếm bus hệ thống (bus hệ thống ở trạng thái bị treo), nó sẽ gửi yêu cầu tới disk controller yêu cầu disk controller chuyển dữ liệu từ disk vào vùng đệm của disk controller, sau khi dữ liệu được kiểm tra tại đây, nó sẽ được chuyển ngay tới main memory mà không qua sự can thiệp của CPU. Sau khi toàn bộ yêu cầu được phục vụ xong thì DMA controller mới trao trả quyền kiểm soát bus hệ thống cho CPU.

Đối với việc input từ bàn phím thì DMA không có lợi điểm gì đáng kể bởi vì việc nhập liệu từ bàn phím không yêu cầu cao về mặt tốc độ và bộ đệm của bàn phím không lớn.

# Câu 33: Khi học môn “Tin học đại cương” chúng ta đã được biết đến một cách chia các thiết bị ngoại vi thành 2 hoặc 3 loại (nhóm) là “VÀO’, “RA” và/hoặc vừa “VÀO vừa RA”; Dưới góc nhìn người thiết kế Hệ điều hành chúng ta có thể chia thiết bị ngoại vi thành 2 loại. Đó là gì và cách chia này dựa trên căn cứ nào? Hãy nêu ví dụ một số ngọai vi thuộc từng loại trên.

Trả lời

Dưới góc nhìn của người thiết kế Hệ điều hành chúng ta có thể chia thiết bị ngoại vi thành 2 loại: block devices (thiết bị khối) và character devices (thiết bị ký tự). Việc chia như thế này là căn cứ vào việc lưu trữ, thao tác đối với dữ liệu.

- Block devices là các thiết bị lưu trữ thông tin dưới dạng các khối dữ liệu, các khối này thường có kích thước từ 512 đến 32,768 byte. Việc truy cập dữ liệu như đọc ghi block là độc lập đối với từng block. Mỗi block có một địa chỉ nhất định. Các câu lệnh liên quan là :readblock (đọc khối), writeblock(ghi khối), seek (định vị khối). Các thiết bị trong nhóm này thường là các thiết bị lưu trữ như hard disk, CD, DVD, MO ...

- Character devices là các thiết bị như bàn phím, chuột, các cổng nối tiếp…,thao tác trên từng ký tự, sử dụng các lệnh như get, put để lấy dữ liệu, sự dụng các phương pháp mã hóa để mã hóa chuỗi bit thành ký tự…

# Câu 34: Giải thích sự khác biệt giữa “polled I/O” và “interrupt-driven I/O”. Nêu những ưu điểm của “interrup-driven I/O” so với “polled I/O”.

Trả lời

“pooled ” (hỏi vòng) và “interrupt”(ngắt) đều là hai phương pháp trao đổi giữa ngoại vi và hệ điều hành, nhưng chúng có sự khác biệt rất lớn.

Phương pháp hỏi vòng là phương pháp đơn giản, dễ lập trình. Các device controller đều có các thanh ghi điều khiển, trên đó thường có một hay một vài bit cờ (status bit). Cpu sẽ thực hiện vòng lặp kiểm tra lần lượt các bit trạng thái này để xác định xem thiết bị nào yêu cầu được phục vụ. Tại một thời điểm nào đó có một thiết bị yêu cầu được phục vụ thì CPU sẽ đáp ứng yêu cầu đó, khi hoàn thành xong thì CPU mới tiếp tục vòng lặp hỏi tới thiết bị tiếp theo. Phương pháp này chỉ áp dụng cho hệ thống có ngoại vi ngang hàng, hoặc các hệ thống đơn giản ít ngoại vi như các hệ thống thu tin, nhận số liệu từ bên ngoài. Phương pháp này có nhược điểm là thời gian đợi lâu, xảy ra hiện tượng độc chiếm CPU hoặc gây lãng phí.

Trong phương pháp sử dụng ngắt thì các ngoại vi nếu yêu cầu được phục vụ sẽ gửi tới CPU tín hiệu ngắt. CPU sẽ kiểm tra, nếu là ngắt không che được thì sẽ gọi chương trình con phục vụ ngắt. Nếu là ngắt bị che thì CPU sẽ bỏ qua yêu cầu đó. Như vậy, trong hệ thống sự dụng ngắt này thì mỗi một lớp thiết bị I/O sẽ kết hợp với một cấu trúc dữ liệu trong bảng mô tả ngắt (interrupt descriptor table), có đầu là các vector ngắt\_ bao gồm các địa chỉ của các chương trình con phục vụ ngắt. Phương pháp này đã xét tới sự ưu tiên, giảm thời gian chờ, tăng hiệu quả CPU, hạn chế được hiện tượng độc chiếm CPU.

Sự khác biệt trên là do sự phát triển của hệ điều hành, với mục đích sử dụng CPU một cách hiệu quả hơn.

# Câu 35: Một cách tương đối, ổ đĩa cứng chậm hơn rất nhiều so với CPU. Nêu các yếu tố đánh giá tốc độ truy xuất dữ liệu trên đĩa. Hãy nêu ngắn gọn các kĩ thuật mà Hệ điều hành sử dụng để nâng cao thông lượng (throughput) trung bình của đĩa.

Trả lời

Tốc độ vào/ra của đĩa phị thuộc vào các yếu tố:

* Thời gian dịch chuyển đầu từ (seek). Thời gian hoàn tất gọi là seek time
* Độ trễ quay vòng (latency). Đến đúng track , chờ cho đến khi khối cần thiết đến dưới đầu đọc. thời gian chờ này gọi là latency time
* Tốc độ truyền dữ liệu. Vận chuyển dữ liệu giữa đĩa và bộ nhớ chính gọi là transfer time

Tổng thời gian cho dịch vụ đĩa chính là tổng của 3 khoảng thời gian trên. Trong đó seek time và latency time là mất nhiều thời gian nhất, do đó để giảm thiểu thời gian truy xuất hệ điều hành đưa ra các thuật toán:

1. FCFS (First Come First Search):

Với thuật toán này, trình tự dịch chuyển đầu từ như là đối với trình tự truy cập tới các cylinder của ổ đĩa cứng. Do vậy, không kết hợp tốt được thời gian dịch chuyển, thuật toán này thực hiện tốt với các tải nhẹ, nhưng khi tải tăng lên, thời gian phục vụ sẽ kéo dài, có trường hợp đầu từ phải đổi hướng liên tục sẽ đẫn tới đầu từ nhanh bị hư hỏng, nhưng bù lại thuật toán đơn giản, dễ lập trình.

1. SSTF (Shortest Seek Time First):

Đầu từ sẽ dịch chuyển tới cylinder gần với nó nhất. Thuật toán này sẽ đảm bảo thời gian dịch chuyển của đầu từ ngắn, làm việc tốt với mức tải vừa phải nhưng lại có vấn đề khi xác định tải dưới mức, có trường hợp đầu từ phải đổi hướng liên tục sẽ đẫn tới đầu từ nhanh bị hư hỏng.

1. SCAN (Elevator):

Đầu từ dịch chuyển theo một chiều nào đó, khi gặp dữ liệu thì đọc luôn, khi đi tới biên (0)thì lại quay ngược lại để đọc dữ liệu. Hoạt động của đầu từ lúc này như hoạt động của thang máy. Cứ đi theo một chiều nào đó, tới cylinder cần đọc dữ liệu thì thực hiện đọc, khi đi tới cylinder có số cylinder cực trị thì quay lại theo chiều ngược lại và tiếp tục đọc dữ liệu như lúc đi. Thuật toán này cũng khá đơn giản, đầu từ ít bị đổi hướng, khắc phục được sự trì hoãn vô định nhưng thời gian phục vụ chưa tốt với trọng tải lớn.

1. C-SCAN:

Đầu từ dịch chuyển chỉ theo một chiều nhất định (chẳng hạn theo chiều tăng của số cylinder), thì sau khi dịch chuyển đến cylinder có số cylinder max sẽ ngay lập tức dịch chuyển đầu từ về 0 và lại bắt đầu đọc theo chiều tăng của số cylinder. Và thực hiện ngược lại nếu đầu từ quay theo chiều giảm của số cylinder. Thuật toán này, chỉ cần dịch chuyển theo chiều tăng của cylinder, nếu thấy có yêu cầu đọc dữ liệu của cylinder nào thì thực hiên đọc luôn. Thuật toán này thực hiện khá đơn giản, nhưng thời gian dịch chuyển của đầu từ khá lớn (chắc chắn lớn hơn thuật toán C-LOOL).

1. C-LOOK:

Thuật toán này tương tự như với C-SCAN, chỉ khác, khi dịch chuyển đến cylinder có số cylinder cao nhất(ko phải là max) thì nó chuyển về cylinder có số cylinder thấp nhất. Với thuật toán này thì thời gian dịch chuyển đầu từ chắc chắn sẽ ngắn hơn thuật toán C-SCAN, nhưng lại cần một bước kiểm tra số cylinder thấp nhất .

Chọn 1 thuật toán phân tích kĩ và lấy VD cụ thể

Kết luận: Với những thuật to¸n lập lịch, vấn đề là phải lựa chọn thuật to¸n nào cho hệ thống. Thuật to¸n SSTF th× rất th«ng thường. Thuật to¸n SCAN và C-SCAN thÝch hợp cho những hệ thống phải truy xuất dữ liệu khối lượng lớn. Với bất kỳ thuật to¸n lập lịch nào, điều quan trọng là khối lượng về số và kiểu khối cần truy xuất. VÝ dụ , nếu số khối cần truy xuất là liªn tục th× FCFS là thuật to¸n tốt.

# Câu 36, 37, 38, 39, 40: Hãy giải thích thuật toán Lập lịch cho đầu từ FCFS, SSTF, SCAN, C-SCan, C-LOOK là gì, cho ví dụ để mô tả để tính được seek time.

Trả lời

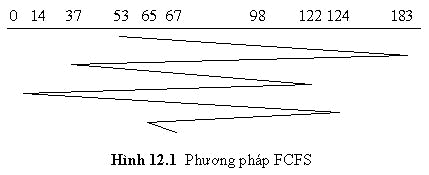
Thời gian dịch chuyển đầu từ từ cylinder này sang cylinder kế tiếp là 5 mili giây. Tổng số thời gian tốn để dịch chuyển đầu từ là bao nhiêu cho mỗi phương pháp dưới đây?

**36. Lập lịch FCFS :**

Phương pháp lập lịch đơn giản nhất là FCFS(first-come,first-served). Thuật toán này rất dể lập trình nhưng không cung cấp được một dịch vụ tốt. Ví dụ : cần phải đọc các khối theo thứ tự như sau :

98, 183, 37, 122, 14, 124, 65, và 67

Giả sử hiện tại đầu đọc đang ở vị trí 53. Như vậy đầu đọc lần lượt đi qua các khối 53, 98, 183, 37, 122, 14, 124, 65, và 67 như hình sau :

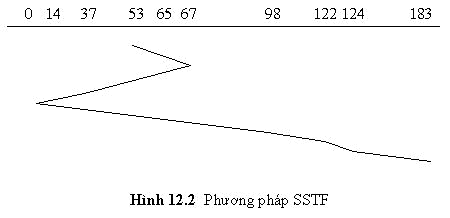


**37. Lập lịch SSTF (shortest-seek-time-first)**

Thuật toán này sẽ di chuyển đầu đọc đến các khối cần thiết theo vị trí lần lượt gần với vị trí hiện hành của đầu đọc nhất. Ví dụ : cần đọc các khối như sau

98, 183, 37, 122, 14, 124, 65, và 67

Giả sử hiện tại đầu đọc đang ở vị trí 53. Như vậy đầu đọc lần lượt đi qua các khối 53, 65, 67, 37, 14, 98, 122, 124 và 183 như hình sau :



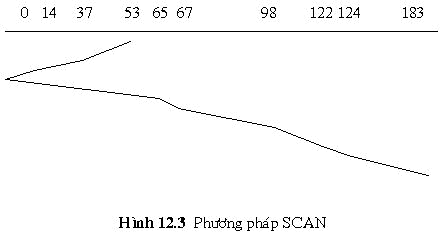
Với ví dụ này, thuật toán SSTF làm giảm số khối mà đầu đọc phải di chuyển là 208 khối.

**38. Lập lịch SCAN**

Theo thuật toán này, đầu đọc sẽ di chuyển về một phía của đĩa và từ đó di chuyển qua phía kia. Ví dụ : cần đọc các khối như sau :

98, 183, 37, 122, 14, 124, 65, và 67

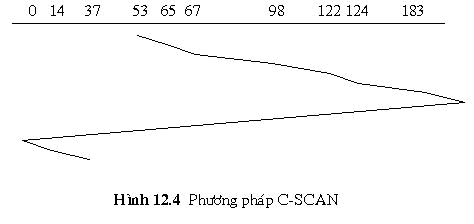
Giả sử hiện tại đầu đọc đang ở vị trí 53. Như vậy đầu đọc lần lượt đi qua các khối 53, 37, 14, 0 , 65, 67, 98, 122, 124 và 183 như hình sau :



Thuật toán này còn được gọi là thuật toán thang máy.Hình ảnh thuật toán giống như hình ảnh của một người quét tuyết, hay quét lá.

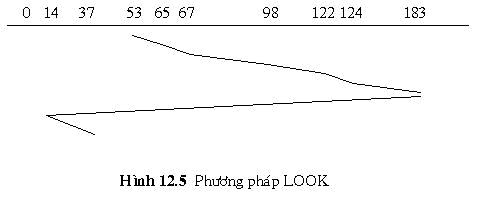
**39. Lập lịch C-SCAN**

Thuật toán này tương tự như thuật toán SCAN, chỉ khác là khi nó di chuyển đến một đầu nào đó của đĩa, nó sẽ lập tức trở về đầu bắt đầu của đĩa. Lấy lại ví dụ trên, khi đó thứ tự truy xuất các khối sẽ là : 53, 65, 67, 98, 122, 124, 183, 199, 0, 14, 37 như hình sau :



**40. Lập lịch LOOK:**

Nhận xét rằng cả hai thuật toán lập lịch SCAN và C-SCAN luôn luôn chuyển đầu đọc của đĩa từ đầu này sang đầu kia. Nhưng thông thường thì đầu đọc chỉ chuyển đến khối xa nhất ở mỗi hướng chứ không đến cuối. Do đó SCAN và C-SCAN được chỉnh theo thực tế và gọi là lập lịch LOOK. Như hình sau :



Lựa chọn thuật toán lập lịch :

Với những thuật toán lập lịch, vấn đề là phải lựa chọn thuật toán nào cho hệ thống.Thuật toán SSTF thì rất thông thường.Thuật toán SCAN và C-SCAN thích hợp cho những hệ thống phải truy xuất dữ liệu khối lượng lớn.Với bất kỳ thuật toán lập lịch nào, điều quan trọng là khối lượng về số và kiểu khối cần truy xuất. Ví dụ , nếu số khối cần truy xuất là liên tục thì FCFS là thuật toán tốt.

1. FCFS

0 2 6 10 20 38 45 92 199

Từ sơ đồ trên ta tính được:

Tổng thời gian của phương pháp này là:

T = ( 10 + 82 + 90 + 43 + 39 + 32 ) \* 0.005= 1.48 (giây )

1. SSTF

0 2 6 10 20 38 45 92 199

Tổng thời gian của phương pháp này:

T =( 18 + 90 ) \* 0.005 = 0.54 ( giây )

1. SCAN

0 2 6 10 20 38 45 92 199

Tổng thời gian trong trường hợp này là:

T = (179+197 ) \* 0.005 = 1.88 ( giây )

1. C-SCAN

0 2 6 10 20 38 45 92 199

Tổng thời gân trong trường hợp này là

T = (20 + 179 + 199 ) \* 0.005 = 1.99 ( giây )

1. C-LOOK

0 2 6 10 20 38 45 92 199

Đối với tất cả các trường hợp trên, đầu từ bắt đầu từ cylinder số 20, và đối với SCAN và C-SCAN đầu từ đang dịch chuyển theo chiều tăng của số cylinder (dịch vào phía tâm của đĩa).

# Câu 42: Tắc nghẽn (deadlock) và các điều kiện để xảy ra tắc nghẽn.

Trả lời

Tắc nghẽn :

Tất cả hiện tượng tắc nghẽn đều bắt nguồn từ sự xung đột về tài nguyên của hai

hoặc nhiều tiến trình đang hoạt động đồng thời trên hệ thống. Tài nguyên ở đây có

thể là một ổ đĩa, một record trong cơ sở dữ liệu, hay một không gian địa chỉ trên bộ nhớ chính.

Điều kiện xảy ra tắc nghẽn :

Năm 1971, Coffman đã đưa ra và chứng tỏ được rằng, nếu hệ thống tồn tại đồng

thời bốn điều kiện sau đây thì hệ thống sẽ xảy ra tắt nghẽn:

1. Loại trừ lẫn nhau (mutual excution) hay độc quyền sử dụng: Đối với các

tài nguyên không phân chia được thì tại mỗi thời điểm chỉ có một tiến

trình sử dụng được tài nguyên.

2. Giữ và đợi (hold and wait): Một tiến trình hiện tại đang chiếm giữ tài

nguyên, lại xin cấp phát thêm tài nguyên mới.

3. Không ưu tiên (No preemption): Không có tài nguyên nào có thể được

giải phóng từ một tiến trình đang chiếm giữ nó.

Sự tắc nghẽn có thể tồn tại với ba điều kiện trên, nhưng cũng có thể không

xảy ra chỉ với 3 điều kiện đó. Để chắc chắn tắc nghẽn xảy ra cần phải có điều kiện

thư tư

4. Đợi vòng tròn (Circular wait): Đây là trường hợp của ví dụ 1 mà chúng

ta đã nêu ở trên. Tức là, mỗi tiến trình đang chiếm giữ tài nguyên mà

tiến trình khác đang cần.

* Ba điều kiện đầu là điều kiện cần chứ không phải là điều kiện đủ để xảy ra

tắc nghẽn. Điều kiện thứ tư là kết quả tất yếu từ ba điều kiện đầu.

# Câu 43: Nguyên tắc/ phương thức chung để nâng cao hiệu năng của hệ thống.

Trả lời:

Nguyên tắc/ phương thức chung để nâng cao hiệu năng của hệ thống:

- Khắc phục hạn chế về lãng phí thời gian của processor trong hệ điều hành;

- Khai thác tối đa thời gian xử lý của processor;

-Tiết kiệm không gian bộ nhớ chính hệ điều hành;

-Chia sẻ thời gian xử lý processor; kết hợp chia sẻ không gian bộ nhớ chính.

# Câu 44: Đánh giá ngắn gọn các loại thiết bị lưu trữ; sự khác nhau căn bản giữa Đĩa cứng và Băng từ; Nguyên tắc tổ chức lưu trữ tệp tin trên 2 loại thiết bị lưu trữ này.

Trả lời:

Thiết bị lưu trữ :là thiết bị có chức năng lưu trữ toàn bộ các thông tin như OS, software, data,... Thiết bị lưu trữ còn được gọi là bộ nhớ ngoài, thuộc loại bộ nhớ bất biến. Dữ liệ trên thiết bị lưu trữ không được truy cập trực tiếp bởi CPU mà sử dụng các kênh vào/ra để đưa dữ liệu vào vùng nhớ trung gian của bộ nhớ chính rồi mới đưa vào CPU để xử lý.

1 số loại thiết bị lưu trữ: đĩa mềm, đĩa cứng, đĩa quang, đĩa từ, ...

Khác nhau:

- Việc thâm nhập vào đĩa từ là ngẫu nhiên, còn việc truy xuất băng từ là tuần tự, như vậy, việc truy xuất dữ liệu trên băng từ mất nhiều thời gian hơn trên đĩa từ.

- Đĩa từ có dung lượng hạn chế, còn băng từ có nhiều cuộn băng có thể lấy ra khỏi máy đọc băng nên dung lượng của băng từ là rất lớn.

- Dữ liệu ghi trên băng từ có cấu trúc gồm 1 số các rãnh song song theo chiều dọc của băng,.

- Dữ liệu ghi trên đĩa cứng gồm nhiều rãnh là các vòng tròn đồng tâm (track), mỗi rãnh gồm nhiều cung (sector) là đơn vị nhỏ nhất lưu trữ dữ liệu. Chuỗi thông tin ghi trên cung gồm có: số thứ tự của cung, 1 khoảng trống, số liệu của cung đó bao gồm cả mã sửa lỗi, 1 khoảng trống, sô thứ tự của cung tiếp theo.

# Câu 45: Các thuộc tính chính và các trạng thái của File các hệ điều hành khác nhau thường có?

- Thuộc tính tập tin :  
Ngoài tên và dữ liệu, hệ điều hành cung cấp thêm một số thông tin cho tập tin gọi là thuộc tính.  
Các thuộc tính thông dụng trong một số hệ thống tập tin :  
 + Tên thuộc tính  
 + Ý nghĩa  
 + Bảo vệ :Ai có thể truy xuất được và bằng cách nào  
 + Mật khẩu :Mật khẩu cần thiết để truy xuất tập tin  
 + Người tạo :Id của người tạo tập tin  
 + Người sở hữu :Người sở hữu hiện tại  
 + Chỉ đọc :0 là đọc ghi, 1 là chỉ đọc  
 + Ẩn :0 là bình thường, 1 là không hiển thị khi liệt kê  
 + Hệ thống :0 là bình thường, 1 là tập tin hệ thống  
 + Lưu trữ :0 đã đuợc backup, 1 cần backup  
 + ASCII/binary :0 là tập tin văn bản, 1 là tập tin nhị phân  
 + Truy xuất ngẫu nhiên :0 truy xuất tuần tự, 1 là truy xuất ngẫu nhiên  
 + Temp: 0 là bình thường, 1 là bị xóa khi tiến trình kết thúc  
 + Khóa :0 là không khóa, khác 0 là khóa  
 + Độ dài của record :Số byte trong một record  
 + Vị trí khóa :Offset của khóa trong mỗi record  
 + Giờ tạo :Ngày và giờ tạo tập tin  
 + Thời gian truy cập cuối cùng :Ngày và giờ truy xuất tập tin gần nhất  
 + Thời gian thay đổi cuối cùng :Ngày và giờ thay đổi tập tin gần nhất  
 + Kích thước hiện thời :Số byte của tập tin  
 + Kích thước tối đa. :Số byte tối đa của tập tin

# Câu 46: Giới thiệu cấu trúc thư mục trong việc tổ chức lưu trữ tệp tin trên đĩa.

Tập tin: là đơn vị lưu trữ thông tin của bộ nhớ ngoài. Các tiến trình có thể đọc hay tạo mới tập tin nếu cần thiết. Thông tin trên tập tin là vững bền không bị ảnh hưởng bởi các xử lý tạo hay kết thúc các tiến trình, chỉ mất đi khi user thật sự muốn xóa. Tập tin được quản lý bởi hệ điều hành.  
  
Thư mục: để lưu trữ dãy các tập tin, hệ thống quản lý tập tin cung cấp thư mục, mà trong nhiều hệ thống có thể coi như là tập tin.  
  
Hệ thống quản lý tập tin: các tập tin được quản lý bởi hệ điều hành với cơ chế gọi là hệ thống quản lý tập tin. Bao gồm : cách hiển thị, các yếu tố cấu thành tập tin, cách đặt tên, cách truy xuất, cách sử dụng và bảo vệ tập tin, các thao tác trên tập tin. Cách tổ chức thư mục, các đặc tánh và các thao tác trên thư mục.

\* MÔ HÌNH TỔ CHỨC VÀ QUẢN LÝ CÁC TẬP TIN

a. Mô hình Tập tin Thư mục Tập tin :

- Tên tập tin :  
Tập tin là một cơ chế trừu tượng và để quản lí mỗi đối tượng phải có một tên. Khi tiến trình tạo một tập tin, nó sẽ đặt một tên, khi tiến trình kết thúc tập tin vẫn tồn tại và có thể được truy xuất bởi các tiến trình khác với tên tập tin đó.  
Cách đặt tên tập tin của mỗi hệ điều hành là khác nhau, đa số các hệ điều hành cho phép sử dụng 8 chữ cái để đặt tên tập tin như ctdl, caycb, tamhghau v.v, thường thường thì các ký tự số và ký tự đặc biệt cũng được sử dụng như baitap2,  
Hệ thống tập tin có thể có hay không phân biệt chữ thường và chữ hoa. Ví dụ : UNIX phân biệt chữ thường và hoa còn MS-DOS thì không phân biệt.  
Nhiều hệ thống tập tin hỗ trợ tên tập tin gồm 2 phần được phân cách bởi dấu "." mà phần sau được gọi là phần mở rộng. Ví dụ : vidu.txt. Trong MS-DOS tên tập tin có từ 1 đến 8 kí tư, phần mở rộng có từ 1 đến 3 kí tự. Trong UNIX có thể có nhiều phần cadch như prog.c.Z. Một số kiểu mở rộng thông thường là :.bak, .bas, .bin, .c, .dat, .doc, .ftn, .hlp, .lib, .obj, .pas, .tex, .txt.  
  
- Cấu trúc của tập tin :

Gồm 3 loại :  
 + Dãy tuần tự các byte không cấu trúc : hệ điều hành không biết nội dung của tập tin:MS-DOS và UNIX sử dụng loại này.  
 + Dãy các record có chiều dài cố định.  
 + Cấu trúc cây : gồm cây của những record, không cần thiết có cùng độ dài, mỗi record có một trường khóa giúp cho việc tìm kiếm nhanh hơn.

- Kiểu tập tin :  
Nếu hệ điều hành nhận biết được loại tập tin, nó có thể thao tác một cách hợp lý trên tập tin đó. Các hệ điều hành hỗ trợ cho nhiều loại tập tin khác nhau bao gồm các kiểu như : tập tin thường, thư mục, tập tin có ký tự đặc biệt, tập tin khối.  
 + Tập tin thường : là tập tin text hay tập tin nhị phân chứa thông tin của người sử dụng.  
 + Thư mục : là những tập tin hệ thống dùng để lưu giữ cấu trúc của hệ thống tập tin.  
 + Tập tin có ký tự đặc biệt : liên quan đến nhập xuất thông qua các thiết bị nhập xuất tuần tự như màn hình, máy in, mạng.  
 + Tập tin khối : dùng để truy xuất trên thiết bị đĩa.  
 + Tập tin thường được chia làm hai loại là tập tin văn bản và tập tin nhị phân.  
. Tập tin văn bản chứa các dòng văn bản cuối dòng có ký hiệu enter. Mỗi dòng có độ dài có thể khác nhau. Ưu điểm của kiểu tập tin này là nó có thể hiển thị, in hay soạn thảo với một editor thông thường.Đa số các chương trình dùng tập tin văn bản để nhập xuất, nó cũng dễ dàng làm đầu vào và đầu ra cho cơ chếpipeline.  
. Tập tin nhị phân : có cấu trúc khác tập tin văn bản. Mặc dù về mặt kỹ thuật , tập tin nhị phân gồm dãy các byte , nhưng hệ điều hành chỉ thực thi tập tin đó nếu nó có cấu trúc động. Ví dụ một một tập tin nhị phân thi hành được của UNIX. Thường thường nó bao gồm năm thành phần : header, text, data, relocation bits,symbol table. Header bắt đầu bởi byte nhận diện cho biết đó là tập tin thi hành. Sau đó là 16 bit cho biết kích thước các thành phần của tập tin, địa chỉ bắt đầu thực hiện và một số bit cờ. Sau header là dữ liệu và text của tập tin. Nó được nạp vào bộ nhớ và định vị lại bởi những bit relocation. Bảng symbol được dùng để debug.

- Truy xuất tập tin :  
Tập tin lưu trữ các thông tin. Khi tập tin được sử dụng, các thông tin này được đưa vào bộ nhớ của máy tính. Có nhiều cách để truy xuất chúng. Một số hệ thống cung cấp chỉ một phương pháp truy xuất, một số hệ thống khác, như IBM chẳng hạn cho phép nhiều cách truy xuất.  
 + Kiểu truy xuất tập tin đơn giản nhất là truy xuất tuần tự . Tiến trình đọc tất cả các byte trong tập tin theo thứ tự từ đầu. Các trình soạn thảo hay trình biên dịch cũng truy xuất tập tin theo cách này. Hai thao tác chủ yếu trên tập tin là đọc và ghi. Thao tác đọc sẽ đọc một mẫu tin tiếp theo trên tập tin và tự động tăng con trỏ tập tin. Thao tác ghi cũng tương tự như vậy. Tập tin có thể tự khởi động lại từ vị trí đầu tiên và trong một số hệ thống tập tin cho phép di chuyển con trỏ tập tin đi tới hoặc đi lui n mẫu tin.  
Truy xuất kiểu này thuận lợi cho các loại băng từ và cũng là cách truy xuất khá thông dụng. Truy xuất tuần tự cần thiết cho nhiều ứng dụng. Có hai cách truy xuất. Cách truy xuất thứ nhất thao tác đọc bắt đầu ở vị trí đầu tập tin, cách thứ hai có một thao tác đặc biệt gọi là SEEK cung cấp vị trí hiện thời làm vị trí bắt đầu. Sau đó tập tin được đọc tuần tự từ vị trí bắt đầu.  
+ Một kiểu truy xuất khác là truy xuất trực tiếp. Một tập tin có cấu trúc là các mẫu tin logic có kích thước bằng nhau, nó cho phép chương trình đọc hoặc ghi nhanh chóng mà không cần theo thứ tự. Kiểu truy xuất này dựa trên mô hình của đĩa. Đĩa cho phép truy xuất ngẫu nhiên bất kỳ khối dữ liệu nào của tập tin. Truy xuất trực tiếp được sử dụng trong trường hợp phải truy xuất một khối lượng thông tin lớn như trong cơ sở dữ liệu chẳng hạn. Ngoài ra còn có một số cách truy xuất khác dự trên kiểu truy xuất này như truy xuất theo chỉ mục ...

- Thuộc tính tập tin :  
Ngoài tên và dữ liệu, hệ điều hành cung cấp thêm một số thông tin cho tập tin gọi là thuộc tính.  
Các thuộc tính thông dụng trong một số hệ thống tập tin :  
 + Tên thuộc tính  
 + Ý nghĩa  
 + Bảo vệ :Ai có thể truy xuất được và bằng cách nào  
 + Mật khẩu :Mật khẩu cần thiết để truy xuất tập tin  
 + Người tạo :Id của người tạo tập tin  
 + Người sở hữu :Người sở hữu hiện tại  
 + Chỉ đọc :0 là đọc ghi, 1 là chỉ đọc  
 + Aån :0 là bình thường, 1 là không hiển thị khi liệt kê  
 + Hệ thống :0 là bình thường, 1 là tập tin hệ thống  
 + Lưu trữ :0 đã đuợc backup, 1 cần backup  
 + ASCII/binary :0 là tập tin văn bản, 1 là tập tin nhị phân  
 + Truy xuất ngẫu nhiên :0 truy xuất tuần tự, 1 là truy xuất ngẫu nhiên  
 + Temp: 0 là bình thường, 1 là bị xóa khi tiến trình kết thúc  
 + Khóa :0 là không khóa, khác 0 là khóa  
 + Độ dài của record :Số byte trong một record  
 + Vị trí khóa :Offset của khóa trong mỗi record  
 + Giờ tạo :Ngày và giờ tạo tập tin  
 + Thời gian truy cập cuối cùng :Ngày và giờ truy xuất tập tin gần nhất  
 + Thời gian thay đổi cuối cùng :Ngày và giờ thay đổi tập tin gần nhất  
 + Kích thước hiện thời :Số byte của tập tin  
 + Kích thước tối đa. :Số byte tối đa của tập tin

\* HỆ THỐNG THƯ MỤC THEO CẤP BẬC :  
  
Có 2 cách thiết kế hệ thống thu mục:

- Thiết kế đơn giản nhất là hệ thống chỉ có thư mục đơn(còn gọi là thư mục một cấp), chứa tất cả các tập tin của tất cả người dùng, cách này dễ tổ chức và khai thác nhưng cũng dễ gây ra khó khăn khi có nhiều người sử dụng vì sẽ có nhiều tập tin trùng tên. Ngay cả trong trường hợp chỉ có một người sử dụng, nếu có nhiều tập tin thì việc đặt tên cho một tập tin mới không trùng lắp là một vấn đề khó.  
- Cách thứ hai là có một thư mục gốc và trong đó có nhiều thư mục con, trong mỗi thư mục con chứa tập tin của người sử dụng (còn gọi là thư mục hai cấp), cách này tránh được trường hợp xung đột tên nhưng cũng còn khó khăn với người dùng có nhiều tập tin. Người sử dụng luôn muốn nhóm các ứng dụng lại một cách logic.  
Từ đó, hệ thống thư mục theo cấp bậc (còn gọi là cây thư mục) được hình thành với mô hình một thư mục có thể chứa tập tin hoặc một thư mục con và cứ tiếp tục như vậy hình thành cây thư mục như trong các hệ điều hành DOS, Windows, v. v...  
Ngoài ra, trong một số hệ điều hành nhiều người dùng, hệ thống còn xây dựng các hình thức khác của cấu trúc thư mục như cấu trúc thư mục theo đồ thị có chu trình và cấu trúc thư mục theo đồ thị tổng quát. Các cấu trúc này cho phép các người dùng trong hệ thống có thể liên kết với nhau thông qua các thư mục chia sẻ.  
  
ĐƯỜNG DẪN :  
  
Khi một hệ thống tập tin được tổ chức thành một cây thư mục, có hai cách để xác định một tên tập tin.

- Cách thứ nhất là đường dẫn tuyệt đối, mỗi tập tin được gán một đường dẫn từ thư mục gốc đến tập tin. Ví dụ : /usr/ast/mailbox.  
- Dạng thứ hai là đường dẫn tương đối, dạng này có liên quan đến một khái niệm là thư mục hiện hành hay thư mục làm việc. Người sử dụng có thể quy định một thư mục là thư mục hiện hành. Khi đó đường dẫn không bắt đầu từ thư mục gốc mà liên quan đến thư mục hiện hành. Ví dụ, nếu thư mục hiện hành là /usr/ast thì tập tin với đường dẫn tuyệt đối /usr/ast/mailbox có thể được dùng đơn giản là mailbox.  
Hầu hết các hệ điều hành đều hỗ trợ hệ thống thư mục theo cấp bậc với hai entry đặc biệt cho mỗi thư mục là "." và "..". "." chỉ thư mục hiện hành, ".." chỉ thư mục cha.

# 

# Câu 48: Tại sao ta có thể nói “Hệ điều hành thực hiện Vào/Ra”. Hãy giải thích cơ chế Chương trình ứng dụng thực hiện I/O như thế nào?

Trả lời:

Vì đó là một trong các chức năng chính của hệ điều hành. Hệ điều hành là quản lý tất cả những thiết bị nhập/xuất của máy tính. Hệ điều hành phải ra các chỉ thị điều khiển thiết bị, kiểm soát các ngắt và lỗi. Hệ điều hành phải cung cấp một cách gioao tiếp đơn giản cà tiện dụng giữa các thiết bị và phần còn lại của hệ thống và giao tiếp này phải được lắp với thiết bị.

Cơ chế chương trình ứng dụng thực hiện I/O: trong một chương trình ứng dụng, người dùng muốn đọc một khối từ một tập tin, hệ điều hành được kích hoạt để thực hiện yêu cầu này. Phần mềm độc lập thiết bị tìm kiếm trong cache, nếu khối cần đọc không có sẵn, nó sẽ gọi chương trình điều khiển thiết bị gửi yêu cầu đến phần cứng. Tiến trình bị ngưng lại cho đến khi thao tác kiểm soát đĩa hoàn tất. Khi thao tác này hoàn tất, phần cứng phát sinh một ngắt. Bộ phận kiểm soát ngắt kiểm tra biến cố này, ghi nhận trạng thái cảu thiết bị và đánh thức tiến trình bị ngưng để chám dứt yêu cầu I/O và cho tiến trình của người sử dung tiếp tục được thực hiện.

# Câu 49: Phân biệt Device, Driver và Device Controller.

Trả lời:

- Device controler: điều khiển công việc phần cơ của thiết bi, thuộc về phần cứng. Làm việc đồng thời song song và độc lập với CPU.

Phải nói về device controler như một mạch điện tử (phần cứng).

- Device driver: hỗ trợ tương tác người dùng với thiết bị, thuộc về phần mềm.

+ Driver: là các đoạn mã lệnh mà qua đó máy tính hiểu được mà điều khiển thiết bị đó nhằm phục vụ mục đích của mình.

Mỗi thiết bị có một Driver riêng biệt, thông qua driver, người dùng có thể thao tác và cấu hình các thiết bị đó theo từng mục đích cá nhân riêng.

Trình điều khiển thiết bị là phần mềm máy tính được viết riêng cho một thiết bị phần cứng cụ thể, cho phép một chương trình, thường là hệ điều hành hoặc ứng dụng phần mềm, tương tác với thiết bị đó. Trình điều khiển thiết bị cung cấp việc xử lí các ngắt cần thiết mà các giao tiếp phần cứng phụ thuộc thời gian bất đối xứng đòi hỏi.

Device: phần loại device: Block, character, virtual

Các divece được phân chia thành 2 loại tùy thuộc vào cách thức đọc/ ghi dữ liệu trên nó:

Block device như: HDD, CDROM, floppy disk, USB stick … tức là các thiết bị dung để lưu trữ dữ liệu lâu dài (thường được gọi là storage device). Dữ liệu trên các device này mỗi lần được đọc ghi theo từng block (khối) byte chẳng hạn 512 byte đối với FDD/HDD, chứ không phải theo từng bit một. Các khối dữ liệu trên các device này được đánh địa chỉ và có thể truy cập đến một địa chỉ bất kì.

Character device như: mouse, modem, printer, … Dữ liệu trên các device này được đọc/ghi theo từng bit một (chứ không phải từng khối byte). Các device này, chả hạn như mouse làm nhiệm vụ chuyển thao tác di chuột thành dữ liệu về vị trí của con trỏ trên màn hình, luôn có dư liệu thay đổi liên tục theo thời gian và không được lưu trữ để sau này dùng lại.

Ngoài ra Linux (cũng như các hệ điều hành khác) còn tạo thêm các “pseudo device” (device giả)