**Đề cương môn hệ điều hành\_hvktqs**

***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 (lời gọi hệ thống) cung cấp 1 giao tiếp giữa tiến trình và hệ điều hành . Lời gọi này cũng như 1 lệnh hợp ngữ . Một số hệ thống cho phép lời gọi hệ thống được thực hiện từ các cấp lập trình bậc cao như các hàm và lời gọi hàm nó có thể phát sinh lời gọi từ các thủ tục hay gọi trực tiếp trong dòng.

2/ ý nghĩa của System call:

 - Tiết kiệm thời gian cho người sử dụng . Người dùng có thể gõ các lệnh hệ thống để cần 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,và 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

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

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.

***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:

     1/ 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.

 2/ 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

 3/ 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 cach 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, phần mềm, 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 mà các chương trình có thể làm việc hữu hiệu 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 phân phối chúng cho các chương trình và người sử dụng khi cần thiết . Do có rất nhiều yêu cầu, hệ điều hành phải giải quyết vấn đề tranh chấp tài nguyên và  quyết định việc phải cấp phát tài nguyên cho những yêu cầu theo thứ tự nào để hoạt động của máy tính là hiệu quả nhất kiểm xoát việc sử dụng máy tính đặc biệt là các thiết bị xuất nhập

   -  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

   Sự khác nhau giữa các hình thức Multi-tasking :

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

     - Trong mô hình đa nhiệm ưu tiên(Preemptive Multitasking), thì việc chuyển CPU 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 CPU. Bộ phận lập lịch thường dựa vào độ ưu tiên của tiến trình để quyết định việc cấp CPU cho nó.

***Câu 5: Sự giống và khác nhau giữa Chương trình , Tiến trình và Luồng*** 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 1 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: 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:  a/ Cơ chế: 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:

Chú thích: Save state into PC0: cất giữ trạng thái trong PCB0

       Reload state from PC1: Khôi phục trạng thái từ PCB1

       Idle: Nghỉ excuting: Hoạt động

  Từ 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.

 b/ 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***

*Trả 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.

 - Mục đích của việc lập lịch cho CPU là để:

-          Tối ưu hóa CPU.

-          Số lượng tiến trình được phục vụ là nhiều nhất.

-          Khoảng thời gian quay vòng là ít nhất.

-          Giảm thời gian chờ trong hàng đợi.

-          Cực tiểu hóa thời gian hoàn tất các tác vụ xử lý theo lô.

-          Cực đại hóa số công việc được xử lý trong một đơn vị thời gian.

 - Nguyên tắc của việc lập lịch: Tổ chức dòng xếp hàng cho các tiến trình , sau đó xác lập 1 chính sách phục vụ cho dòng xếp hàng đó.

***Câu 9: Giải thích tại sao phần địa chỉ cao(top half), địa chỉ thấp(bottom-half) HĐH không phải là tiến trình.***

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 10: Các cách lựa chọn tiến trình từ “ready to run” để thực hiện.***

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

 1/ 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) nên 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.

 2/ 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 và kích hoạt)

   - Hạn chế:

   +Phải mất thời gian cho việc ước lượng về thời gian để 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).

 3/ Round – robin scheduling (RR - Định thời luân phiên )

 - Ưu điểm: + Đây là giải thuật đơn giản, công bằng và phổ dụng.

         + Tránh bị rơi 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 gian chia cho các tiến trình không hợp lý sẽ lãng phí thời gian.

 4/ Prioity scheduling ( Lập lịch ưu tiên):

  -  Đặc điểm: Mỗi tiến trình được gán một độ ưu tiên và 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 ưu tiên của nó sẽ giảm đi 1 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 do xác định được tiến trình nào quan trọng cho thực hiện trước.

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

***Câu 11: Trong kĩ thuật cấp phát bộ nhớ phân vùng động,khi nạp tiến trình vào bộ nhớ cần lựa chọn vùng còn rỗng nào đó. Nêu các thuật toán hệ điều hành có thể sử dụng trong trường hợp cần lựa chọn này.***

Trả lời:Đối với hệ điều hành sử dụng phân vùng động,để cấp phát bộ nhớ cho một tiến trình ta sử dụng một số thuật toán như First – Fit, Best – fit, Worst – fit, Quick – fit. Mỗi một phương pháp có những đặc điểm riêng ưu điểm và hạn chế nhất định

  - First – fit (chọn lỗ trống đầu tiên đủ lớn đối với kích thước của tiến trình): quản lí quá trình sẽ quét dọc theo danh sách các đoạn cho đến khi tìm được một lỗ trống có kích thước đủ lớn. Như vậy trong quá trình đó tiến trình sẽ được cấp phát lỗ trống đầu tiên có kích thước lớn hơn hoặc bằng nó. Như vậy lỗ trống được cấp phát có thể chia làm 2 phần: phần cấp phát cho tiến trình và phần không sử dụng đến (tức là nó còn rảnh).

- Best – fit (chọn lỗ trống có kích thước nhỏ nhất mà đủ lớn cho kích thước của tiến trình): một tiến trình sẽ được cấp phát lỗ trống mà có kích thước vừa khít với nó nhất trong danh sách lỗ trống rảnh, tức là phần không sử dụng đến trong khối là nhỏ nhất. Yêu cầu phải duyệt hết toàn bộ các lỗ trống trừ khi các lỗ trống đã được sắp xếp theo kích thước.

- Worst – fit (chọn lỗ trống có kích thước lớn nhất để cung cấp cho tiến trình): một tiến trình sẽ được cấp phát phần bộ nhớ thuộc khối rảnh có kích thước lớn nhất trong danh sách các khối rảnh. Cũng phải duyệt hết toàn bộ danh sách

Trong đó: first – fit là thuật toán đơn giản nhất với thời gian tìm kiếm là nhanh nhất do bắt đầu gặp khối có kích thước đủ cho tiến trình là nó nhận,

best fit tốt hơn worst fit trong giới hạn của tốc độ và khả năng lưu trữ.

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

Trả lời: - Các trạng thái của tiến trình: Khi một tiến trình thực thi, nó thay đổi trạng thái.Trạng thái của tiến trình tại 1 thời điểm được xác định bởi các hoạt động hiện hành của nó tại thời điểm đó.Mỗi tiến trình có thể ở 1 trong các trạng thái sau:

 + Mới (new): tiến trình đang được tạo ra.

 + Đang chạy (running): các chỉ thị đang được thực thi.

 + Chờ (waiting): tiến trình đang chờ sự kiện xảy ra (VD: hoàn thành việc nhập/xuất hay nhận dạng tín hiệu).

 + Sẵn sàng (ready): tiến trình đang chờ được cấp phát CPU để xử lý.

 + Kết thúc (terminated): tiến trình hoàn thành việc thực thi.

Tại một thời điểm xác định, chỉ có duy nhất 1 tiến trình ở trạng thái *running* nhưng nhiều tiến trình có thể ở trạng thái *waiting* hay *ready*.

 - Các mô hình tiến trình hệ thống:

   + Mô hình simple monitor system:

 + Mô hình hệ thống lô (batch system)

 +Mô hình hệ thống lô với trạng thái cấm (a batch system with blocked state)

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

Trả lời: Mỗi quá trình được hiện diện trong hệ điều hành bởi Khối điều khiển quá trình(Process Control Block-PCB) – cũng được gọi khối điều khiển tác vụ.  Nó chứa nhiều phần thông tin được gắn liền với một quá trình xác định, gồm:

**• Trạng thái quá trình** (process state): trạng thái có thể là mới, sẳn sàng, đang chạy, chờ đợi, kết thúc, …

• **Bộ đếm chương trình** (program counter): bộ đếm hiển thị địa chỉ của chỉ thị kế tiếp được thực thi cho quá trình này.

• **Các thanh ghi** (registers) CPU: các thanh ghi khác nhau về số lượng và loại, phụ thuộc vào kiến trúc máy tính. Chúng gồm các bộ tổng (accumulators), các thanh ghi chỉ mục, các con trỏ ngăn xếp, và các thanh ghi đa năng (generalpurpose registers), cùng với thông tin mã điều kiện (condition-code information). Cùng với bộ đếm chương trình, thông tin trạng thái này phải được lưu khi một ngắt xảy ra, cho phép quá trình được tiếp tục một cách phù hợp sau đó (Hình III.3).

• **Thông tin lập thời biểu CPU** (CPU-scheduling information): thông tin gồm

độ ưu tiên của quá trình, các con trỏ chỉ tới các hàng đợi lập thời biểu, và bất

kỳ tham số lập thời biểu khác.

• **Thông tin quản lý bộ nhớ** (Memory-management information): thông tin này có thể gồm những thông tin như giá trị của các thanh ghi nền và thanh ghi giới hạn, các bảng trang hay các bảng phân đoạn, phụ thuộc hệ thống bộ nhớ được dùng bởi hệ điều hành.

• **Thông tin tính toán** (accounting information): thông tin này gồm lượng CPU và thời gian thực được dùng, công việc hay số quá trình,…

• **Thông tin trạng thái nhập/xuất** (I/O status information): thông tin này gồm danh sách của thiết bị nhập/xuất được cấp phát quá trình này, một danh sách các tập tin đang mở,.. PCB đơn giản phục vụ như kho chứa cho bất cứ thông tin khác nhau từ quá trình này tới quá trình khác.

***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:

1/ Quản lý tiến trình: Tiến trình là 1 chương trình đang hoạt động.Khi thực hiện, tiến trình đòi hỏi 1 số tài nguyên nhất định như :CPU, bộ nhớ, các thiết bị nhập xuất… Các tài nguyên này sẽ được cấp phát cho tiến trình vào những thời điểm cần thiết và được thu hồi khi tiến trình kết thúc. Ngoài ra, khi tiến trình hoạt động trong hệ thống có thể phát sinh các tiến trình con. Như vậy, nhiệm vụ của hệ điều hành trong quản lý tiến trình là:

-          Tạo lập và hủy bỏ các tiến trình

-          Tạm dừng và khôi phục lại các tiến trình

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

-          Cung cấp các cơ chế để giao tiếp giữa các tiến trình

2/ Quản lý bộ nhớ chính: Bộ nhớ chính là 1 mảng kiểu byte hay kiểu word có kích thước lớn. Mỗi phần tử có địa chỉ riêng. Đó là 1 kho chứa dữ liệu có khả năng truy xuất nhanh được chia sẻ bởi CPU và các thiết bị nhập xuất. Bộ nhớ chính là 1 thiết bị lưu trữ không ổn định.Nội dung của nó bị mất khi hệ thống bị lỗi.Trong quản lý bộ nhớ hệ điều hành có các nhiệm vụ sau:

  - Lưu giữ thông tin về các vị trí trong bộ nhớ đã được sử dụng và ai sử dụng.

  - Quyết định tiến trình nào được nạp vào bộ nhớ chính, khi bộ nhớ đã có thể dùng được.

   - Cấp phát và thu hồi bộ nhớ khi cần thiết.

3/ Quản lý bộ nhớ phụ: Vì bộ nhớ chính bất ổn định và quá nhỏ để lưu tất cả chương trình và dữ liệu nên hệ thống máy tính phải cung cấp việc lưu trữ phụ để lưu dự phòng bộ nhớ chính. Hầu như tất cả chương trình : chương trình dịch, hợp ngữ, thủ tục, trình soạn thảo, định dạng... đều được lưu trữ trên đĩa cho tới khi nó được thực hiện, nạp vào trong bộ nhớ chính và cũng sử dụng đĩa để chứa dữ liệu và kết quả xử lý. Vì vậy một bộ quản lý hệ thống đĩa rất quan trọng cho hệ thống máy tính. Vai trò của hệ điều hành trong việc quản lý đĩa : Quản lý vùng trống trên đĩa, định vị lưu trữ và lập lịch cho đĩa.

4/ Quản lý hệ thống nhập/ xuất: Một trong những mục tiêu của hệ điều hành là *che dấu* những đặc thù của các thiết bị phần cứng đối với người sử dụng thay vào đó là một lớp thân thiện hơn, người sử dụng dể thao tác hơn. Một hệ thống nhập/xuất bao gồm :

  - Một hệ thống vùng nhớ đệm

  - Giao diện trình điều khiển thiết bị chung.

  - Điền khiển các thiết bị phần cứng xác định.

5/ Quản lý tập tin: Một tập tin là một tập hợp những thông tin lien quan với nhau do người tạo ra nó xác định. Thông thường một tập tin đại diện cho một chương trình và dữ liệu. Dữ liệu của tập tin có thể là số, là ký tự, hay ký số. Tập tin thường có dạng tự do, như tập tin văn bản, nhị phân...(là tập tin chứa dãy các bit). Vai trò của hệ điều hành trong việc quản lý tập tin :

Tạo và xoá một tập tin.

Tạo và xoá một thư mục.

Hỗ trợ các hàm nguyên thủy để thao tác trên tập tin và thư mục.

Ánh xạ tập tin trên hệ thống lưu trữ phụ.

Sao lưu dự phòng tập tin trên các thiết bị lưu trữ ổn định.

6/ Hệ thống bảo vệ: ***Hệ thống bảo vệ*** là một cơ chế kiểm soát quá trình truy xuất của chương trình, tiến trình, hoặc người sử dụng với tài nguyên của hệ thống và của người dùng. Cơ chế này cũng cung cấp cách thức để mô tả lại mức độ kiểm soát:

  - Phân biệt giữa người sử dụng được phép hay không được phép truy cập.

  - Xác định các điều khiển được áp dụng.

  - Cung cấp phương tiện để thực thi.

7/ Mạng(distributed system):Hệ phân tán là tập hợp các bộ xử lý, chúng không chia sẻ bộ nhớ, các thiết bị ngoại vi hay đồng hồ.Thay vào đó mỗi bộ vi xử lý có bộ nhớ, đồng hồ riêng.Các bộ vi xử lý trong hệ thống được kết nối với nhau thông qua mạng truyền thông.Hệ thống phân tán cung cấp cho người dùng truy xuất tới các tài nguyên khác nhau mà hệ thống duy trì. Truy xuất tới các tài nguyên chia sẻ cho phép tăng tốc độ tính toán,tăng khả năng sẵn dùng của dữ liệu, tăng mức độ tin cậy.

8/ Thành phần thông dịch lệnh: Nhiều lệnh được cung cấp tới hệ điều hành bởi các lệnh điều khiển để giải quyết việc tạo và quẩn lý tiến trình,quẩn lý nhập xuất, quản lý việc lưu trữ phụ, quản lý bộ nhớ chính, truy xuất hệ thống tập tin, bảo vệ và mạng. Trong các hệ thống chia xẻ thời gian một chương trình có thể đọc và thông dịch các lệnh điều khiển được thực hiện một cách tự động. Chương trình này thường được gọi là bộ thông dịch điều khiển card, cơ chế dòng lệnh hoặc Shell. Chức năng của nó rất đơn giản đó là lấy lệnh kế tiếp và thi hành.

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

1.      Kỹ thuật phân vùng động (Dynamic Partitioning)

-    Số lượng các phân vùng trên bộ nhớ và kích thước của mỗi phân vùng là có thể thay đổi.

-Khi có một tiến trình được nạp vào bộ nhớ nó được hệ điều hành cấp cho nó không gian vừa đủ để chứa tiến trình, phần còn lại được trả về tập hợp các vùng trống. Nếu vùng trống mới nằm kề với các cùng trống khác, các vùng trống nằm kề này được gom lại để tạo thành 1 vùng trống lớn hơn.

-Khi một tiến trình kết thúc nó được đưa ra ngoài và phần không gian bộ nhớ mà tiến trình này trả lại cho hệ điều hành sẽ được hệ điều hành cấp cho tiến trình khác, cả khi tiến trình này có kích thước nhỏ hơn kích thước của không gian nhớ trống đó.

-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 điều hành phải quyết định chọn một khối nhớ trống phù hợp nào để nạp tiến trình sao cho việc lựa chọn này dẫn đến việc sử dụng bộ nhớ chính là hiệu quả nhất.

-Có 3 thuật toán mà hệ điều hành sử dụng trong trường hợp này, đó là: Fist fit, Best Fit, Worst Fit

2. Kỹ thuật phân vùng cố định (Fixed Partitioning)

-Quản lý bộ nhớ với những phân đọan cố định .Hệ điều hành chia bộ nhớ thành n vùng nhớ cố định( có thể không bằng nhau)  
-Việc phân chia này được thực hiện vào lúc khởi động hệ thống và không thay đổi suốt quá trình chạy.Với tổ chức như vậy cần duy trì một hàng đợi duy nhất để lưu trữ những tiến trình chưa được cấp phát bộ nhớ

-Tất cả tiến trình được đặt trong một hàng đợi duy nhất. Khi có một phân vùng tự do , tiến trình đầu tiên trong hàng đợi có kích thước phù hợp sẽ được đặt vào phân vùng này và cho xử lý.  
-Nếu kích thước của tiến trình không vừa đúng bằng kích thước phân vùng chứa nó, phần bộ nhớ không sử dụng đến trong phân vùng sẽ bị lãng phí xảy ra hiện tượng phân mảnh nội vi.  
-Mức độ đa chương của hệ thống bị giới hạn bởi số lượng phân vùng.Vấn đề bảo vệ giữa các phân vùng: Để bảo vệ cần tổ chức hai thanh ghi : thanh ghi nền và thanh ghi giới hạn.  
-Khi tiến trình được tạo lập, nạp vào thanh ghi nền địa chỉ bắt đầu của phân vùng được cấp phát cho tiến trìnhvà nạp vào thanh ghi giới hạn kích thước của tiến trình  
- Sau đó mỗi đị chỉ bộ nhớ được phát sinh sẽ tự động được cộng với địa chỉ chứa trong thanh ghi nền để cho ra địa chỉ tuyệt đối trong bộ nhớ và các địa chỉ được đối chiếu với thanh ghi giới hạn để bảo đảm tiến trình không truy xuất ngoài phạm vi được cấp phát cho nó.

3. Swapping : hoán đổi tạm thời một tiến trình khỏi bộ nhớ ra thiết bị lưu trữ ngoài (swap out ) (backing store = vùng bộ nhớ nhanh nhất, tốt nhất để chứa bản sao của tất cả các hình ảnh bộ nhớ cho all user, phải cung cấp truy cập trực tiếp cho hình ảnh bộ nhớ đó), giải phóng bộ nhớ, sau đó có thể được đưa trở lại bộ nhớ để tiếp tục thực hiện.(swap in). Thời gian swap chính là time chuyển tiến trình từ bộ nhớ chính đến bộ nhớ phụ, một tiến trình cần phải có time xử lý đủ lớn.

4. Phân trang : Phân bộ nhớ vật lý thành các khối (block) có kích thước cố định và bằng nhau, gọi là khung trang (page frame). Không gian địa chỉ cũng được chia thành các khối có cùng kích thước với khung trang, và được gọi là trang (page). Khi cần nạp một tiến trình để xử lý, các trang của tiến trình sẽ được nạp vào những khung trang còn trống. Lưu trạng thái của các khung trang. Một tiến trình kích thước N trang sẽ yêu cầu N khung trang tự do. Tạo một bảng trang để chuyển đổi địa chỉ logic sang địa chỉ vật lý. Phương pháp này cấp phát không liền nhau nên có hiện tượng phân mảnh nội= kích thước phân vùng cố định.

5. Phân đoạn : chương trình thường được chia thành nhiều phần : dữ liệu, lệnh, ngăn xếp. Chương trình được chia thành các đoạn theo cấu trúc logic, mỗi đoạn được phân vào một vùng bộ nhớ có kích thước bằng nhau, mỗi đoạn tương ứng với khôn gian địa chỉ riêng được phân biệt bởi tên và độ dài của chúng. Các vùng nhớ thuộc các đoạn khác nhau có thể nằm ở các vị trí khác nhau, do vậy chương trình có thể chiếm nhiều hơn một đoạn và không cần liên tiếp trong bộ nhớ. Tránh được phân mảnh trong nhưng lại có hiện tượng phân mảnh ngoài. Dễ sắp xếp bộ nhớ, dễ chia sẻ các đoạn giữa các tiến trình khác nhau, kích thước mỗi đoạn có thể thay đổi mà không ảnh hưởng đến đoạn khác.

5. Phân đoạn với phân trang : Tức là phân trang trong mỗi phân đoạn, bộ nhớ = nhiều phân đoạn và phân đoạn= nhiều trang. Mỗi phần tử của bảng phân đoạn gồm hai thành phần : Thanh ghi giới hạn (limit chứa kích thước của phân đoạn) và thanh ghi cơ sở (base chứa địa chỉ của bảng )

**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.**

-    First fit : Cấp phát bằng cách tìm khối nhớ trống nào có độ lớn đủ lớn đầu tiên để cấp (có nhiều hole nhỏ nhưng không đủ cấp cho một tiến trình mới nào cả thì được gọi là phân mảnh ngoài, ‘ngoài’ vì nó không thuộc một tiến trình nào- fregmentation)

-         Best fit : Phân mảnh thì làm lãng phí bộ nhớ. Vậy có phương pháp này, tức là tìm hole nào có độ lớn đủ lớn nhỏ nhất để cấp phát. Phải tím kiếm trên toàn bộ danh sách vùng trống.

-         Worst fit : tìm hole nào có độ lớn đủ lớn nhưng lớn nhất để cấp phát cho tiến trình mới. Phải tìm kiếm trên toàn bộ danh sách vùng trống

-         BF và FF tốt hơn về tốc độ và tối ưu hóa việc sử dụng bộ nhớ

-

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

-         Trang được hình thành từ việc chia bộ nhớ logic thành các khối có cùng kích thước

-         Phân trang là khi tiến trình cần xử lý thì nạp trang của tiến trình vào các khung  trang nào còn trống .Phải nắm được trang nào thuộc tiến trình nào thì mới quản lý và cấp phát được. Muốn vậy cần phải bảo vệ các tiến trình truy xuất vào các trang k phải của mình.  Việc bảo vệ vùng nhớ được thực hiện bằng cách liên kết một khung với một bit gọi là bit kiểm tra hợp lệ (valid-invalid bit). Valid chỉ ra rằng trang đi kèm là nằm trong không gian địa chỉ logic của tiến trình, vì vậy truy xuất trang này là hợp lệ. Ngược lại là invalid.

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

-          Chương trình được chia thành các đoạn(segments) theo cấu trúc logic bao gồm chương trình chính, thủ tục, hàm, biến cục bộ, biến toàn cục, commonblock, ngăn xếp, các mảng, bảng symbol, mỗi đoạn được phân vào một vùng bộ nhớ có kích thước bằng nhau, mỗi đoạn tương ứng với khôn gian địa chỉ riêng được phân biệt bởi tên và độ dài của chúng. Các vùng nhớ thuộc các đoạn khác nhau có thể nằm ở các vị trí khác nhau, do vậy chương trình có thể chiếm nhiều hơn một đoạn và không cần liên tiếp trong bộ nhớ. Tránh được phân mảnh trong nhưng lại có hiện tượng phân mảnh ngoài. Dễ sắp xếp bộ nhớ, dễ chia sẻ các đoạn giữa các tiến trình khác nhau, kích thước mỗi đoạn có thể thay đổi mà không ảnh hưởng đến đoạn khác.

**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.**

-         Bộ nhớ ảo là một kỹ thuật cho phép xử lý một tiến trình không được nạp toàn bộ vào bộ nhớ vật lý. Bộ nhớ ảo mô hình hoá bộ nhớ như một bảng lưu trữ rất lớn và đồng nhất, tách biệt hẳn khái niệm không gian địa chỉ và không gian vật lý. Người sử dụng chỉ nhìn thấy và làm việc trong không gian địa chỉ ảo, việc chuyển đổi sang không gian vật lý do hệ điều hành thực hiện với sự trợ giúp của các cơ chế phần cứng cụ thể.

-         Bộ nhớ ảo có thể được thực hiện thông qua phân trang và phân đoạn theo yêu cầu. các tiến trình có thể được phân ra thành các trang hoặc các đoạn và không nhất thiết phải nằm liên tục với nhau. Mọi sự truy xuất vùng nhớ của tiến trình đều được chuyển đổi địa chỉ lúc thi hành do đó có thể swap in và swap out. Vì 2 lý do trên nên các trang hoặc các phân đoạn không nhất thiết phải nằm trong bộ nhớ chính lúc thi hành.

-         Ưu điểm : Có nhiều tiến trình trong bộ nhớ hơn, giải thuật lập lịch sẽ tối ưu hơn, nâng cao mức độ đa chương, một tiến trinh có thể lớn hơn kích thước của bộ nhớ chính. Chương trình không còn bị ràng buộc bởi không gian địa chỉ vật lý sẵn có.Người dùng có thể viết chương trình có không gian địa chỉ ảo rất lớn, đơn giản hóa tác vụ lập trình Cho phép các tiến trình sử dụng chung không gian địa chỉ và dễ dàng chia sẻ tài nguyên, cung cấp cơ chế hữu hiệu cho việc tạo quá trình.

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

-         Phân trang theo yêu cầu =phân trang +swapping

-         Một trang chỉ được nạp vào bộ nhớ chính khi có yêu cầu

+ giảm các thao tác vào ra

+ tiết kiệm bộ nhớ

+ đáp ứng nhanh

+ tăng được số người sử dụng (tiến trình)

-         Khi cần một trang à tham chiếu đến nó

+ tham chiếu lỗi à hủy bỏ

+ không nằm trong bộ nhớ  trongà đưa trang vào bộ nhớ từ thiết bị lưu trữ ngoài (Ram) vào bộ nhớ trong. Lúc này có 2 TH sảy ra : nếu còn frame thì load trang vào frame tương ứng. Nếu không còn frame thì phải swap out 1 page nào đó ra và nạp trang mới vào. Chỉ swap 1 page vào một frame, cùng một trang có thể được nạp vào bộ nhớ trong vài lần.

-         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

-         Cơ chế phần cứng hỗ trợ kỹ thuật phân trang theo yêu cầu là sự kết hợp của cơ chế hỗ trợ kỹ thuật phân trang và kỹ thuật swapping:

-         Bảng trang: Cấu trúc bảng trang phải cho phép phản ánh tình trạng của một trang là đang nằm trong bộ nhớ chính hay bộ nhớ phụ.

-         Bộ nhớ phụ: Bộ nhớ phụ lưu trữ những trang không được nạp vào bộ nhớ chính. Bộ nhớ phụ thường được sử dụng là đĩa, và vùng không gian đĩa dùng để lưu trữ tạm các trang trong kỹ thuật swapping được gọi là không gian swapping.

**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 gải quyết vấn đề đó như thế nào và mục tiêu cần đạt được?**

Truy xuất đến một trang được đánh dấu bất hợp lệ sẽ làm phát sinh một lỗi trang (page fault). Khi dò tìm trong bảng trang để lấy các thông tin cần thiết cho việc chuyển đổi địa chỉ, nếu nhận thấy trang đang được yêu cầu truy xuất là bất hợp lệ, cơ chế phần cứng sẽ phát sinh một ngắt để báo cho hệ điều hành. Hệ điều hành sẽ xử lý lỗi trang như sau :

Kiểm tra truy xuất đến bộ nhớ là hợp lệ hay bất hợp lệ

Nếu truy xuất bất hợp lệ : kết thúc tiến trình

Ngược lại : đến bước 3

Tìm vị trí chứa trang muốn truy xuất trên đĩa.

Tìm một khung trang trống trong bộ nhớ chính :

Nếu tìm thấy : đến bước 5

Nếu không còn khung trang trống, chọn một khung trang « nạn nhân » và chuyển trang « nạn nhân » ra bộ nhớ phụ (lưu nội dung của trang đang chiếm giữ khung trang này lên đĩa), cập nhật bảng trang tương ứng rồi đến bước 5

Chuyển trang muốn truy xuất từ bộ nhớ phụ vào bộ nhớ chính : nạp trang cần truy xuất vào khung trang trống đã chọn (hay vừa mới làm trống ) ; cập nhật nội dung bảng trang, bảng khung trang tương ứng.

Tái kích hoạt tiến trình người sử dụng.

(Câu hỏi thêm ai là người quyết định thứ tự của các trang được thực hiện à User vì họ chạy lệnh nào thì nó sẽ thực hiện trang đó, ví dụ vừa soạn thảo, sau đó phân khung….)

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

Vấn đề chính khi thay thế trang là chọn lựa một trang « nạn nhân » để chuyển ra bộ nhớ phụ. Có nhiều thuật toán thay thế trang khác nhau, nhưng tất cả cùng chung một mục tiêu chọn trang « nạn nhân » là trang mà sau khi thay thế sẽ gây ra ít lỗi trang nhất.

Có thể đánh giá hiệu qủa của một thuật toán bằng cách xử lý trên một chuỗi các địa chỉ cần truy xuất và tính toán số lượng lỗi trang phát sinh.

-         FIFO : chọn các trang đã được nạp vào bộ nhớ trong lâu nhất để thay thế à thuật toán này dễ hiểu, dễ cài đặt . Tuy nhiên khi thực hiện không phải lúc nào cũng có kết qủa tốt : 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 bộ nhớ phụ sẽ nhanh chóng gây ra lỗi trang cho những lần truy xuất sau. Số lượng lỗi trang xảy ra sẽ tăng lên khi số lượng khung trang sử dụng tăng. Hiện tượng này gọi là nghịch lý Belady.

-         OPT : thay thế trang sẽ lâu được sử dụng nhất trong tương laià thuật toán này hoàn hảo về mặt ý tưởng nhưng không khả thi về mặt thực tế. Thuật toán này bảo đảm số lượng lỗi trang phát sinh là thấp nhất , nó cũng không gánh chịu nghịch lý Belady, tuy nhiên, đây là một thuật toán không khả thi trong thực tế, vì không thể biết trước chuỗi truy xuất của tiến trình trong tương lai!

-         LRU : thay thế trang lâu nhất trong bộ nhớ chưa được sử dụng à dùng quá khứ gần để đoán tương lai, FIFO để biết thời điểm nạp vào, tối ưu thời điểm sẽ truy cập, LRU đòi hỏi phần cứng phải hỗ trợ khá nhiều : biến bộ đếm, stack.

**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.**

-         Như trên

-         Ví dụ đại loại thế này

-

**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.**

-         Như trên

-         Ví dụ đại loại như sau :

**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.**

-         Như trên

-         Ví dụ

**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?**

**- Phân mảnh ngoại (external fragmentation): là hiện tượng khi kích thước không gian nhớ còn trống đủ để thỏa mãn yêu cầu cấp phát nhưng không gian nhớ này lại không liên tục**. Hiện tượng phân mảnh ngoại xảy ra khi bạn thường xuyên cấp phát vùng nhớ mới, sau đó xóa đi những phần vùng nhớ đã cấp phát một cách không thứ tự.  
**- Phân mảnh nội (internal fragmentation): là hiện tượng sẽ có vùng nhớ dư thừa khi ta cấp phát một vùng nhớ hơi lớn hơn kích thước yêu cầu.**  
**Ví dụ: Cấp một khoảng trống 18,464 bytes cho một process yêu cầu** 18,462 bytes  
Hiện tượng phân mảnh nội thường xảy ra khi bộ nhớ thực được chia thành các khối kích thước cố định (fixed-sized block) và các process được cấp phát theo đơn vị khối.

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

  Một tiến trình có thể được tạm thời hoán đổi ra khỏi bộ nhớ tới backing store, và rồi được đưa trở lại bộ nhớ để thực hiện tiếp.

  Backing store – thiết bị nhớ thứ cấp đủ lớn( đĩa từ) để cung cấp bản sao của tất cả hình ảnh bộ nhớ cho tất cả người sử dụng; phải cung cấp sự truy nhập trực tiếp tới các hình ảnh bộ nhớ này.

  Roll out, roll in – biến thể hoán đổi được sử dụng cho thuật giải lập lịch dựa trên mức ưu tiên (priority-based scheduling); tiến trình có mức ưu tiên thấp hơn bị thay ra để tiến trình có mức ưu tiên cao hơn có thể được nạp và thực hiện.

  Phần lớn thời gian hoán đổi là thời gian chuyển dữ liệu; tổng thời gian chuyển tỷ lệ thuận với dung lượng bộ nhớ hoán đổi.

  Swap out: chọn tiến trình để đưa ra backing store

  Swap in: chọn tiến trình từ backing store để đưa vào bộ nhớ trong

  Trong các hệ điều hành sử dụng swapping, tồn tại module hệ thống swapper có chức năng: chọn tiến trình swap out, chọn tiến trình swap in, định vị & quản lý không gian chuyển

**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?**

-         DMA là bộ nhớ truy xuất trực tiếp không thông qua bộ vi xử lý (cho phép chuyển dữ liệu trực tiếp giữa I/O và RAM mà không cần đi ngang qua CPU nữa), được sử dụng để truyền tải dữ liệu giữa bộ nhớ và các thiết bị ngoại vi như ổ đĩa cứng, CD ROM,DVD… được nhanh hơn. Điều này sẽ thấy rõ ràng khi ghi đĩa, tốc độ đọc và ghi đĩa sẽ tăng lên rất nhiều. Với DMA thì CPU không phải truyền thông tin từ ngoại vi sang controller nữa, cho phép CPU chia tải (offload) các ngăn xếp bộ nhớ ngày càng chật chội sang các thành phần khác, cách thức này giải phóng CPU khỏi các công việc lặt vặt và cung cấp nhiều chu trình hơn để giải quyết các công việc phức tạp hơn, thường dùng truyền tải dữ liệu với dung lượng lớn.

-         DMA sẽ không có lợi gì đối với input thông qua bàn phím bởi vì input bởi bàn phím thì rất chậm, bàn phím chỉ chuyển được 1 kí tự/1 lần, còn DMA thì chuyển theo block cho nên nếu làm như vậy thì sẽ rất mất thời gian, gây lãng phí.

**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.**

-         Đa nhiệm: tức là có thể thực hiện đồng thời nhiều chương trình, điều này được thực hiện bằng cách chuyền từ chương trình này sang chương trình khác một cách tương đối nhanh chóng để tạo cho người dùng cảm giác tất cả các chương trình đang được xử lý đồng thời. Đa nhiệm nhiều người dùng: cho phép nhiều người đồng thời đăng nhập vào hệ thống. Việc này được quản lí thông qua tài khoản người dùng và mật khẩu tương ứng(VD: các phiên bản mới HĐH Windows như Win 2000,XP,7,8,Mac OS X, **iPhone OS 4.0, WebOS, Windows Phone 7, Android, Unix- đa trình**...).

**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.**

Để phân loại các thiết bị vào/ra người ta dựa vào khuôn dạng dữ liệu mà thiết bị thực hiện vào/ra với hệ thống. Các thiết bị vào/ra có thể chia thành hai loại chủ yếu: Các thiết bị khối và các thiết bị ký tự.

Các thiết bị khối:  Các thiết bị khối là các thiết bị thực hiện vào/ra hệ thống theo từng khối dữ liệu, mỗi khối có kích thước cố định, có địa chỉ. Kích thước các khối như nhau nếu cùng một thiết bị, ngược lại các thiết bị khác nhau thì chúng có thể thực hiện vào/ra theo khối dữ liệu có kích thước khác nhau. Nói chung mỗi khối thường có kích thước từ 128-1024 bytes. Các thiết bị khối đọc/ghi dữ liệu theo khối vì vậy trong quá trình xử lý có thể thực hiện tìm kiếm, lựa chọn. Ví dụ: đĩa từ, card mạng (NIC)...

Các thiết bị ký tự: Các thiết bị là các thiết bị thực hiện vào/ra với hệ thống theo từng dãy ký tự gối nhau. Các dãy ký tự không được địa chỉ hóa vì vậy khi xử lý không thể tìm kiếm hoặc lựa chọn. Ví dụ: Màn hình, máy in...

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

Ø  Cơ chế thăm dò (polling)

CPU thường xuyên kiểm tra port của thiết bị:

•      Thiết bị nhập: Có tín hiệu nhập không? nếu có đọc dữ liệu

•      Thiết bị xuất: Có trong trạng thái sẵn sàng không? Nếu có xuất dữ liệu.

à Hao phí CPU để kiểm tra thiết bị nhập à cơ chế Interrupt.

Ø  Cơ chế ngắt (intterupt)

            Mỗi thiết bị có một đường tín hiệu riêng gọi là đường intterupt

            Khi thiết bị nhập có dữ liệu:

a)     Thiết bị gởi tín hiệu ngắt về CPU

b)     CPU tạm dừng chương trình đang thi hành

c)     CPU thi hành trình xử lý ngắt để đọc dữ liệu tại port

d)     Sau khi hoàn tất, CPU tiếp tục chương trình bị dừng

**Câu 36: Hãy giải thích thuật toán Lập lịch cho đầu từ FCFS là gì, cho ví dụ để mô tả để tính được seek time. Lập lịch FCFS :**

-         Phương pháp lập lịch đơn giản nhất là FCFS(first-come,first-served), đọc dữ liệu theo đúng trình tự của blog thông tin. Thuật toán này rất dễ  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 :

-

**Câu 37: Hãy giải thích thuật toán Lập lịch cho đầu từ SSTF là gì, cho ví dụ để mô tả để tính được seek time.**

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 khố

**Câu 38: Hãy giải thích thuật toán Lập lịch cho đầu từ SCAN là gì, cho ví dụ để mô tả để tính được seek time.**

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.

**Câu 39: Hãy giải thích thuật toán Lập lịch cho đầu từ C-SCAN là gì, cho ví dụ để mô tả để tính được seek time**

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 :

**Câu 40: Hãy giải thích thuật toán Lập lịch cho đầu từ C-LOOK là gì, cho ví dụ để mô tả để tính được seek time.**

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.(túm lại là khi đến cylinder chứa thông tin cần đọc cuối cùng nó không quay lại 0 nữa mà quay ra lun cái còn lại) 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.

**Câu 42: Tắc nghẽn (deadlock)  và các điều kiện để xảy ra 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

Hiện tượng tắc nghẽn xảy ra khi và chỉ khi hệ thống tồn tại 4 điều kiện sau:

- Có thể sử dụng tài nguyên không thể chia sẻ: Mỗi thời điểm, một tài nguyên chỉ được cấp phát cho một tiến trình. Khi tiến trình sử dụng xong, hệ thông mới thu hồi và cấp phát tài nguyên này cho tiến trình khác.

- Có hiện tượng giữ và đợi tài nguyên: Các tiến trình chiếm giữ các tài nguyên đã cấp phát cho nó và chờ được cấp phát thêm tài nguyên mới

- Không có hệ thống phân phối lại tài nguyên: Tài nguyên không thể được được thu hồi từ tiến trình đang chiếm giữ chúng trước khi tiến trình này sử dụng chúng xong

- Có hiện tượng chờ đội vòng tròn: Có ít nhất 2 tiến trình chờ đợi lẫn nhau: tiến trình này chờ được cấp phát tài nguyên mà tiến trình kia đang chiếm giữ và ngược lại

**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.**

-         Thiết bị lưu trữ là thiết bị có chức năng chính là lưu trữ toàn bộ các thông tin như OS, phần mềm, dữ liệu, nó còn được gọi là bộ nhớ phụ hay bộ nhớ ngoài, thuộc loại bộ nhớ bất biến.

**Câu 47: Các phương pháp được sử dụng để quản lí, cấp phát các block đĩa cho các File.**

- HĐH chia không gian đĩa thành những phần có kích thước bằng nhau gọi là Block. Nội dung của 1 file được lưu trữ trong khối này

               1. Quản lý các Block tự do, HĐH dùng phương pháp Bitmap Mỗi khối tự do tương đương với bit 0. Dùng bảng bitmap, mỗi phần tử trong bảng cho biết trạng thái của Block trên đĩa. Như vậy khi cấp phát 1 file có n khối tự do thì HĐH sẽ tìm n bít = 0 trên bảng Bitmap này để cấp phát cho file đó.

               2. Cấp phát theo danh sách liên kết: Chiến lược này được thể hiện 1 danh sách liên kết các Block để chứa nội dung của 1 file. Từ đầu tiên của khối đầu tiên sẽ được sử dụng một con trỏ để trỏ đến khối tiếp theo và cho đến hết

               3. Cấp phát theo danh sách liên kết sử dụng chỉ mục: Lưu các con trỏ trên vào một bảng chỉ mục, nạp bảng chỉ mục này vào bộ nhớ khi HĐH đọc nội dung của file trên đĩa.

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

**Divice controler** : Điều khiển công việc phần cơ của thiết bị , thuộc về phần cứng . Làm việc đồng thời song song và độc lập với CPU

**Divice driver** : hổ trợ tương tác người dùng với thiết bị , thuộc về phần mềm .