

CHƯƠNG 1

- 1. Chế độ đặc quyền là gì? Cho một ví dụ của lệnh đặc quyền.**
 - Chế độ đặc quyền là có quyền truy cập đầy đủ k hạn chế tài nguyên của máy tính. VD: truy cập vào đĩa từ
- 2. Hệ thống đa chương là gì (multiprogramming system)**
 - Hệ thống đa chương cho phép nhiều chương trình tải cùng một thời điểm, hoạt động cùng một lúc, hệ thống có thể chia sẻ thời gian, chia sẻ cpu..
- 3. Hệ thống được sử dụng trong các thiết bị chuyên dụng điều khiển máy móc là hệ thống nào (Đa chương, đa xử lý, thời gian thực, phân tán)**
 - là hệ điều hành thời gian thực
- 4. Hệ điều hành nào hỗ trợ tính toán hiệu năng cao (chia sẻ thời gian, xử lý song song, xử lý theo lô, thời gian thực)**
 - Xử lý song song
- 5. Hệ thống có khả năng tương tác hiệu quả với người dùng là hệ thống nào? (Đơn chương, đa chương, chia sẻ thời gian)**
 - là hệ thống chia sẻ thời gian
- 6. Hệ điều hành phát triển được chi phối bởi yếu tố gì?**
 - chi phối bởi chỉ số H/P
- 7. Hệ điều hành trên máy tính MainFrame có các đặc tính gì?**
 - hoạt động trên máy tính có nhiều cpu, có bộ nhớ lớn có khả năng chia sẻ, có tính tin cậy và xử lý cao...
- 8. Hệ thống chia sẻ thời gian là gì? (Timesharing System)**
 - hệ thống chia sẻ thời gian là hệ thống đa chương, đa nhiệm thực hiện nhiều công việc thông qua chuyển đổi cpu thời gian diễn ra rất nhanh.
- 9. Ba mục đích chính của một hệ điều hành là gì?**
 1. giúp người dùng sử dụng một cách dễ dàng hơn
 2. tối ưu hóa sử dụng phần cứng, khai thác tối đa
 3. làm hệ thống máy tính sử dụng dễ dàng hơn
- 10. Hệ điều hành có thể phân loại theo những tiêu chí nào? Mỗi cách phân loại có những loại hệ điều hành nào?**
 - Có thể phân loại theo: dựa trên quy mô hmtt(sever, hđh thiết bị dd)
 - Dựa trên góc độ người sd(đơn đa người sử dụng), theo chương trình (đơn đa nhiệm, phân tán,)
- 11. Hệ điều hành thời gian thực có những loại gì? Đặc tính của từng loại?**
 - Có 2 loại hard và soft
- 12. Trong một môi trường đa chương và chia sẻ thời gian, nhiều người dùng chia sẻ hệ thống đồng thời, điều này có thể mang đến những vấn đề gì? Chúng ta có thể đảm bảo mức độ bảo mật giống như trên các máy riêng lẻ không?**
 - Không đảm bảo tính an toàn của hệ thống, không thể đảm bảo
- 13. Hệ điều hành đã trải qua các giai đoạn phát triển nào? Thách thức và vấn đề của từng giai đoạn là gì?**

Trải qua 4 giai đoạn dựa vào chỉ số H/P :

 - Gdd1: hệ thống hoạt động hết công suất
 - gd2: đảm bảo dễ dàng cho người sử dụng và nhiều người tương tác
 - gd3: giá thành rẻ và máy cấu hình thấp
 - gd4: kết nối mt với nhau và đảm bảo an toàn mới

14.Đưa ra 2 lý do tại sao cache hữu ích. Nêu những vấn đề cache giải quyết. Nêu những vấn đề do cache gây ra?

- Lợi ích :giúp giảm thời gian truy cập và tăng hiệu suất của cpu giải quyết vấn đề chen lệch về tốc độ giao tiếp giữa các phần với nhau .
- Vấn đề gây ra hiện tượng mất bộ nhớ ,tăng giá thành của hệ thống

15.Lập lịch ưu tiên là gì? (preemptive) hay lập lịch không đặc quyền

- Lập lịch ưu tiên là cho phép chương trình kết thúc việc thực hiện một chương trình khác hay 1 công việc tiến trình có thể ngắt để thực hiện chương trình khác

16.Sự khác biệt giữa bất và ngắt là gì? Chức năng của chúng là gì?

- Ngắt tạo ra ở phần mềm ,bất phần cứng thực hiện một công việc nào đó.tạo ra cho hdd có thể nhận biết dc những sự kiện.

17.Hệ điều hành phải làm gì để khắc phục vấn đề về phần cứng, màn hình nhỏ... trên thiết bị di động

- Thiết kế giao diện phù hợp ,sử dụng phần cứng ,tài nguyên và năng lượng được hiệu quả....

18.Chế độ giám sát và chế độ người sử dụng giúp bảo vệ hệ thống như thế nào?

- Chế độ giám sát những cái hoạt động quan trọng ảnh hưởng hdd dưới chế độ giám sát bảo vệ dc tài nguyên quan trọng của hệ thống

CHƯƠNG 2-TIẾN TRÌNH

1. Kể tên 1 hệ thống chỉ cho phép một tiến trình thực thi tại một thời điểm thời gian? 1 hệ thống cho phép nhiều tiến trình thực thi đồng thời?

- Hệ thống có k sd pipeline
- Hệ thống có sử dụng pipeline, linux

2. Trong Unix, lời gọi hệ thống nào được sử dụng để tạo tiến trình. Fork ()

3. Một tiến trình sẽ bị kết thúc trong các trường hợp nào?

- Con đã vượt quá tài nguyên được cấp
- Nhiệm vụ được giao cho tt không còn cần thiết
- Tt cha đang thoát ra

4. Trạng thái Sẵn sàng của tiến trình là gì?

- Tiến trình đang đợi để dc gán cho một quá trình xử lý

5. Các tiến trình liên lạc với nhau để làm gì?

- Chia sẻ thông tin
- Hợp tác hoàn thành tác vụ

6. Khi mở một tab mới trên trình duyệt chrome, trình duyệt này có tạo ra tiến trình mới không? Có

7. Khi mở một tab mới trên trình duyệt chrome, trình duyệt này có tạo ra tiến trình mới không? Có

8. Sự khác biệt, mối quan hệ giữa tiến trình và tiểu trình ?

- Tiến trình có không gian địa chỉ và chỉ có một dòng xử lý. Các tiểu trình thì chia sẻ một không gian địa chỉ,
- Một tiến trình có thể có nhiều tiểu trình chạy đồng thời

9. Tại sao các hệ điều hành hiện đại hỗ trợ môi trường đa nhiệm ?

- Như vậy sẽ tăng hiệu suất sd CPU và tăng tốc độ xử lý

10.Sự khác biệt, mối quan hệ giữa tiến trình và tiểu trình ?

- Tiến trình có không gian địa chỉ và chỉ có 1 dòng xử lý. Tiểu trình chia sẻ 1 không gian địa chỉ và các dòng này hđ song song như các tiến trình phân biệt.
- Một tiến trình có thể có nh tiểu trình chạy đồng thời

11. Tiến trình và luồng khác nhau gì?

- Tiến trình là sự thực thi của chương trình (tức là những hoạt động của chương trình đó). Luồng là một sự thực thi đơn lẻ có tuần tự bên trong tiến trình (tức là những nhiệm vụ mà tiến trình đó phải thực hiện). Một tiến trình có thể chứa nhiều luồng cũng như một chương trình có thể chứa nhiều tiến trình. Vậy nên có thể nói luồng là đơn vị nhỏ nhất trong một chương trình và đôi khi một luồng được xem như là một tiến trình nhỏ (tiểu trình) bên trong một tiến trình lớn. Hiểu được sự khác biệt này chúng ta có thể phân bố tốt chương trình để có thể giải quyết chương trình đó một cách tối ưu nhất.

12. Phân biệt sự khác nhau trong cách tiếp cận để ưu tiên cho tiến trình ngắn trong các thuật toán điều phối sau: FCFS, RR, và điều phối với độ ưu tiên đa cấp

- FCFS: Trong thuật toán này, độ ưu tiên phục vụ tiến trình căn cứ vào thời điểm hình thành tiến trình. Hàng đợi các tiến trình được tổ chức theo kiểu FIFO. Mọi tiến trình đều được phục vụ theo trình tự xuất hiện cho đến khi kết thúc hoặc bị ngắt
- RR: Tương tự như định thời FIFO nhưng sự trưng dụng CPU được thêm vào để chuyển CPU giữa các quá trình. Hàng đợi sẵn sàng được xem như một hàng đợi vòng. Bộ định thời CPU di chuyển vòng quanh hàng đợi sẵn sàng, cấp phát CPU tới mỗi quá trình có khoảng thời gian tối đa bằng một định mức thời gian.

CHƯƠNG 3-BỘ NHỚ

1. Giải thích sự khác biệt giữa địa chỉ logic và địa chỉ physic?

Địa chỉ logic	Địa chỉ Physic
Là địa chỉ sinh bởi CPU, tham chiếu đến một vị trí độc lập với cấu trúc, tổ chức vật lý của bộ nhớ. Mọi tham chiếu bộ nhớ trong mã lệnh chương trình được compiler tạo ra đều là địa chỉ logic. VD: các tiến trình biên dịch tạo ra mã lệnh chương trình mà trong đó tham chiếu đến bộ nhớ đều là địa chỉ logic	Là địa chỉ sinh bởi đơn vị quản lý bộ nhớ MMU, là một địa chỉ thực trong bộ nhớ chính, địa chỉ Physic được chuyển từ địa chỉ logic khi trình biên dịch thực hiện quá trình linking/loading

2. Giải thích sự khác biệt giữa hiện tượng phân mảnh nội vi và ngoại vi?

Phân mảnh nội vi	Phân mảnh ngoại vi
<ul style="list-style-type: none">- Là hiện tượng sẽ có vùng nhớ dư thừa khi ta cấp phát một vùng nhớ hơi lớn hơn kích thước yêu cầu. VD: Cấp một khoảng trống 24,484 bytes cho một process yêu cầu 24,482 bytes.- Hiện tượng phân mảnh nội vi thường xảy ra khi bộ nhớ thực được chia thành các khối kích thước xảy ra khi bộ nhớ thực chia thành các khối kích thước cố định và các process được cấp phát theo đơn vị khối	<ul style="list-style-type: none">- Là hiện tượng khi kích thước không gian nhớ còn trống đủ thỏa mãn yêu cầu cấp phát nhưng không gian nhớ này lại không liên tục.- Hiện tượng phân mảnh ngoại vi 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.

3. Xét một hệ thống trong đó một chương trình khi được nạp vào bộ nhớ sẽ phân biệt hoàn toàn phân đoạn code và phân đoạn data. Giả sử CPU sẽ xác định được khi nào cần truy xuất lệnh hay dữ liệu, và phải truy xuất ở đâu. Khi đó mỗi chương trình sẽ được cung cấp 2 bộ thanh ghi base-limit: một cho phân đoạn code, và một cho phân

đoạn data. Bộ thanh ghi base-limit của phân đoạn code tự động được đặt thuộc tính readonly. Thảo luận các ưu và khuyết điểm của hệ thống này.

- *Ưu điểm:*

+ Do bộ thanh ghi base limit của phân đoạn code tự động được đặt thuộc tính readonly, nên chương trình đó có thể được chia sẻ cho nhiều người dùng khác nhau.

+ Kỹ thuật này có thuận lợi trước hết là vì nó là một cơ chế có hiệu quả trong việc chia sẻ lệnh(code) và dữ liệu(data). Một thí dụ là, chỉ duy nhất có một bản sao của editor(trình soạn thảo) hay một compiler(trình biên dịch) cần thiết được giữ trong bộ nhớ, và đoạn code này được chia sẻ bởi tất cả các tiến trình cần truy xuất đến editor code hay compiler code(code của trình soạn thảo hay trình biên dịch).

+ Một ưu điểm nữa của kỹ thuật này là bảo vệ phần code của chương trình đối với việc sửa đổi có thể gây ra lỗi cho chương trình.

- *Khuyết điểm:* Kỹ thuật này gây ra một khuyết điểm là code và dữ liệu bị phân ra thành 2 phần khác nhau, vì trong thực tế 2 thành phần này thường được kết hợp với nhau trong khi phát sinh trình biên dịch cho chương trình.

4. Tại sao kích thước trang luôn là lũy thừa của 2 ?

Kích thước trang luôn là lũy thừa của 2 vì: để thực hiện việc dịch địa chỉ luận lý thành số trang và độ dời trang rất dễ dàng.

5. Tại sao trong hệ thống sử dụng kỹ thuật phân trang, một tiến trình không thể truy xuất đến vùng nhớ không được cấp cho nó ? Làm cách nào hệ điều hành có thể cho phép sự truy xuất này xảy ra ? Hệ điều hành có nên cho phép điều đó không ? Tại sao?

- Trong hệ thống sử dụng kỹ thuật phân trang, một tiến trình không thể truy xuất đến vùng nhớ không được cấp cho nó vì: sự phân chia rõ ràng giữa tầm nhìn bộ nhớ của người dùng và bộ nhớ vật lý thực sự.
- Cách để hệ điều hành có thể cho phép sự truy xuất này xảy ra: Sự thật, chương trình người dùng được phân bổ khắp bộ nhớ vật lý và nó cũng quản lý các quá trình khác. Sự khác nhau giữa tầm nhìn bộ nhớ của người dùng và bộ nhớ vật lý thực sự được làm cho tương thích bởi phần cứng dịch địa chỉ. Địa chỉ luận lý được dịch thành địa chỉ vật lý.
- Hệ điều hành có nên cho phép điều đó. Vì bằng cách này, việc có thêm vùng nhớ giúp người dùng có được quyền hành chi tiết hơn đối với hệ thống bộ nhớ và các phương thức truy cập.

6. Nếu cho phép hai phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang trong bộ nhớ thì sẽ có hiệu quả gì ? Giải thích làm cách nào hiệu quả này có thể được sử dụng để giảm thời gian cần khi sao chép một khối lượng lớn vùng nhớ từ vị trí này sang vị trí khác. Khi đó nếu sửa nội dung một trang thì sẽ tác động đến trang còn lại thế nào?

- Nếu cho phép hai phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang trong bộ nhớ thì sẽ có hiệu quả: giảm thời gian cần khi sao chép một khối lượng lớn vùng nhớ từ vị trí này sang vị trí khác.
- Cách làm hiệu quả này có thể được sử dụng để giảm thời gian cần khi sao chép một khối lượng lớn vùng nhớ từ vị trí này sang vị trí khác bằng việc thay đổi số hiệu khung trang.
- Khi đó nếu sửa nội dung một trang thì sẽ tác động đến trang còn lại: sẽ được ánh xạ toàn bộ trên trang còn lại (thực tế không phải là trang còn lại mà là chính nó).

7. Vì sao đôi lúc người ta kết hợp hai kỹ thuật phân trang và phân đoạn ?

- Đôi lúc người ta kết hợp hai kỹ thuật phân trang và phân đoạn vì: hai kỹ thuật này được kết hợp để có thể hỗ trợ lẫn nhau. Phân đoạn các trang thì có ích trong trường hợp các bảng trang trở nên rất lớn. Một vùng liên tục của bảng trang mà không được sử dụng có thể chia vào trong một mục trong bảng phân đoạn với địa chỉ trang là 0. Phân trang các đoạn xử lý trường hợp có nhiều đoạn dài mà cần nhiều thời gian để cấp phát. Bằng việc phân trang các đoạn, chúng ta giảm thiểu lãng phí bộ nhớ do phân mảnh ngoại cũng như đơn giản quá trình cấp phát.

8. Mô tả cơ chế cho phép một phân đoạn có thể thuộc về không gian địa chỉ của hai tiến trình.

- Cơ chế cho phép một phân đoạn có thể thuộc về không gian địa chỉ của hai tiến trình: do bảng phân đoạn là tập hợp các thanh ghi nền và thanh ghi giới hạn(base-limit register), một phân đoạn nào đó có thể được chia sẻ khi một mục trong bảng phân đoạn của 2 tiến trình khác nhau ánh xạ vào cùng một địa chỉ vật lý duy nhất. Các bảng phân đoạn của 2 tiến trình này cần có con trỏ nền cho mỗi bảng phân đoạn, và số hiệu đoạn cần chia sẻ cần giống nhau ở 2 tiến trình

9. Giải thích vì sao chia sẻ một module trong kỹ thuật phân đoạn lại dễ hơn trong kỹ thuật phân trang?

- Chia sẻ một module trong kỹ thuật phân đoạn lại dễ hơn trong kỹ thuật phân trang vì: phân đoạn thì dựa trên sự phân chia logic bộ nhớ chứ không phải phân chia vật lý, do đó những đoạn có kích thước bất kỳ có thể được chia sẻ với chỉ một mục trong bảng phân đoạn của tiến trình. Trong phân trang thì bắt buộc phải có một mục chung trong bảng phân trang cho mỗi trang nhớ được chia sẻ.

10. Xét bảng phân đoạn sau đây :

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Cho biết địa chỉ vật lý tương ứng với các địa chỉ logic sau đây :

a. 0,430	a) Địa chỉ vật lý tương ứng = $219 + 430 = 649$ (hợp lệ)
b. 1,10	b) Địa chỉ vật lý tương ứng = $2300 + 10 = 2310$ (hợp lệ)
c. 2,500	c) Địa chỉ vật lý tương ứng = $90 + 500 = 590$ (không hợp lệ vì địa chỉ logic=500 > Length=100)
d. 3,400	d) Địa chỉ vật lý tương ứng = $1327 + 400 = 1727$ (hợp lệ)
e. 4,112	

e) Địa chỉ vật lý tương ứng = 1952 + 112 (không hợp lệ vì địa chỉ logic=112>Length=96)
--

CHƯƠNG 5-FILE

1. Tập tin là gì? Thư mục là gì? Tại sao phải quản lý tập tin và thư mục?

- 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 là: để lưu trữ 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.
- Phải quản lý tập tin và thư mục vì: để dễ dàng tìm kiếm hoặc kiểm tra khi cần thiết, vì số lượng tập tin và thư mục sẽ sinh sôi và nảy nở theo thời gian (VD: số lượng câu hỏi của Google là tập tin, số nhóm của Google là thư mục nhỏ, ... Trong một ngày có biết bao nhiêu câu hỏi và trả lời nếu không đặt ngăn nắp, đúng thư mục thì sau này sẽ không biết đâu mà tìm).

2. Tập tin có những đặc tính gì? Những đặc tính nào là quan trọng? Tại sao?

- Tập tin có những đặc tính:
 - + Các tiến trình có thể đọc
 - + Các tiến trình có thể 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
- Đặc tính quan trọng của tập tin là: Thông tin trên tập tin là vững bền không bị ảnh hưởng bởi các xử lý tạo hay kết thúc tiến trình. Vì tập tin là đơn vị lưu trữ thông tin của bộ nhớ ngoài dùng để lưu trữ dữ liệu nếu bị mất khi tạo hoặc kết thúc tiến trình thì nó sẽ mất đi chức năng quan trọng nhất của tập tin là lưu trữ dữ liệu cho người dùng sử dụng hoặc cho các tiến trình.

3. Giải thích mục đích của các hàm open() và close() đối với các tập tin?

- Mục đích của các hàm open() đối với các tập tin: là cho phép hệ thống thiết lập một số thuộc tính và địa chỉ đĩa trong bộ nhớ để tăng tốc độ truy xuất.
- Mục đích của các hàm close() đối với các tập tin: là khi chấm dứt truy xuất, thuộc tính và địa chỉ trên đĩa không cần dùng nữa, tập tin được đóng lại để giải phóng vùng nhớ.

4. Một số hệ thống tự động xóa toàn bộ file của người dùng khi người dùng đăng xuất hoặc tắt máy. Một số hệ thống thì giữ lại toàn bộ file của người dùng ? Phân tích ưu nhược điểm của từng hệ thống ?

	Hệ thống tự động xóa toàn bộ file của người dùng khi người dùng đăng xuất hoặc tắt máy.	Hệ thống thì giữ lại toàn bộ file của người dùng
Ưu điểm	- Tăng tính bảo mật thông tin	- Dễ dàng sửa đổi dữ liệu
Nhược điểm	- Khó sửa đổi dữ liệu	- Bảo mật thông tin kém hơn

5. Nếu sử dụng lệnh chmod 752 thì quyền truy cập của file đó sẽ như thế nào? Quyền truy cập của file sẽ được hiểu: trị số 752 được sinh ra từ:

- Owner: 4 (read) + 2 (write) + 1 (execute) = 7
 - Group: 4 (read) + 1 (execute) = 5
 - Others: 2 (write) = 2
- ⇒ Nếu sử dụng lệnh chmod 752 thì quyền truy cập của file đó sẽ: Cấp quyền truy cập đầy đủ cho Owner, chỉ cho phép Group dùng đọc và chạy các file trong thư mục, và chỉ cho phép Others dùng đọc.

6. Có 3 File có thông tin về phân quyền như sau:

File 1: -rw-rw-r-

File 2: rwxrw----

File 3 : rw-r-----

a) Hãy giải thích chế độ phân quyền của cả 3 file

b) Sử dụng lệnh chmod với tham số thế nào để được phân quyền như 3 file trên

a) Chế độ phân quyền của 3 file là:

- File 1: Các user thuộc Owner được phép đọc và sửa file, các user thuộc Group có thể đọc và sửa file, còn những người còn lại là chỉ được đọc file.
- File 2: Cấp quyền truy cập đầy đủ cho Owner, chỉ cho phép Group có thể đọc và sửa file, nhưng không phân quyền truy cập cho những người dùng khác.
- File 3: Các user thuộc Owner được phép đọc và sửa file, các user thuộc Group có thể đọc file, nhưng không phân quyền truy cập cho những người dùng khác.

b) Để được phân quyền như 3 file trên ta sử dụng lệnh chmod với tham số:

File 1: 664

File 2: 760

File 3: 640

#	Permission	rwX
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x

CHƯƠNG 6: IO SYSTEMS

4. Nhiệm vụ của Device Driver là gì ?

- Device drivers phát ra các chỉ thị và kiểm tra xem chỉ thị đó có được thực hiện chính xác không.
- Chức năng của device drivers là nhận những yêu cầu trừu tượng từ phần mềm nhập/xuất độc lập thiết bị ở lớp trên, và giám sát yêu cầu này thực hiện. Nếu driver đang rảnh, nó sẽ thực hiện ngay yêu cầu, ngược lại, yêu cầu đó sẽ được đưa vào hàng đợi.

5. Có bao nhiêu loại thiết bị I/O ? Với mỗi loại cho ví dụ cụ thể.

- Thiết bị tương tác được với con người : dùng để giao tiếp giữa người và máy. Ví dụ: màn hình, bàn phím, chuột, máy in ...

- Thiết bị tương tác trong hệ thống máy tính là các thiết bị giao tiếp với nhau Ví dụ: đĩa, băng từ, card giao tiếp...
- Thiết bị truyền thông : như modem...

6. So sánh Pooling, Interrupts và DMA ?

- **Pooling:** Thời gian phản ứng lại sự kiện phụ thuộc rất lớn vào vị trí mà chương trình đang thực hiện trong vòng lặp.
- Thời gian phản ứng cũng là một hàm của số lượng sự kiện được kích hoạt tại một thời điểm và sau đó là thời gian thực hiện các dịch vụ trong một lần quét vòng lặp của chương trình.
- Tất cả các sự kiện được chương trình đối xử một cách bình đẳng và không có sự ưu tiên nào cả.
- Khi có một đặc tính mới, do đó là dịch vụ mới, được thêm vào chương trình, thời gian phản ứng lại dài thêm ra.
- **Interrupt:** Rất hữu dụng và cũng gây không ít khó khăn cho người lập trình
- Các chế độ ngắt và xử lý ngắt cho phép thực hiện các quá trình tốc độ cao, phản ứng kịp thời với các sự kiện ở bên trong và bên ngoài Chương trình ngắt hoạt động gần giống với chương trình con: vòng quét chương trình chính tạm thời ngừng, chuyển sang thực hiện vòng quét của chương trình ngắt Chương trình con được kích hoạt bằng lệnh gọi chương trình con, chương trình ngắt được kích hoạt bằng tín hiệu ngắt
- Ý tưởng của ngắt: sự xuất hiện của một sự kiện có thể “ngắt” tiến trình thực hiện của chương trình, “nhồi” thêm và thực hiện một tiến trình khác vào như hình 2. Khi tiến trình nhồi thêm được thực hiện xong, chương trình chính lại được thực hiện tiếp từ thời điểm bị ngắt. Tiến trình của sự kiện ngắt được thực hiện ngay lập tức mà không phải quan tâm đến chương trình chính. Những tiến trình như thế người ta gọi là “chương trình con dịch vụ ngắt”
- **DMA:** thường thì hoạt động chuyển thông tin giữa thiết bị I/O với bộ nhớ máy tính được thực hiện thông qua CPU theo cơ chế tuần tự: CPU ra lệnh "import" để nhập 1 đơn vị thông tin (byte/word) từ thiết bị I/O vào thanh ghi CPU rồi ra lệnh ghi thông tin từ thanh ghi CPU vào ô nhớ RAM nào đó. Quy trình này rất chậm vì CPU phải tốn nhiều chu kỳ máy để thực hiện các lệnh di chuyển thông tin, hơn nữa thông tin còn phải đi vòng qua CPU trước khi đến được vị trí cuối cùng. Để tăng tốc độ chuyển dữ liệu giữa I/O và RAM, người ta đã thiết kế mạch cứng DMA (Direct Memory Access) cho phép chuyển dữ liệu trực tiếp giữa I/O và RAM mà không cần đi ngang qua CPU nữa. Như vậy, nếu driver thiết bị I/O hỗ trợ tốt chế độ hoạt động DMA thì bạn nên cho phép chế độ này hoạt động.
- Điểm lợi của DMA: Với những chức năng giao tiếp ngoài có tính liên tục như xử lý âm thanh, kết nối mạng với bên ngoài thì rất cần thiết. CPU không thể xử lý đa nhiệm cùng lúc nhiều chức năng nên buộc phải sử dụng DMA. Nếu gói dữ liệu chưa thể ghi hết vào DMA thì CPU có thể đợi (do CPU có tốc độ lớn hơn nhiều với giao tiếp ngoài) cho đến khi nào cờ ngắt báo đã ghi xong.

7. Cơ chế hoạt động DMA như thế nào ?

- Khi sử dụng, CPU gửi cho bộ điều khiển một số các thông số như địa chỉ trên đĩa của khối, địa chỉ trong bộ nhớ nơi định vị khối, số lượng byte dữ liệu để chuyển.
- Sau khi bộ điều khiển đã đọc toàn bộ dữ liệu từ thiết bị vào buffer của nó và kiểm tra checksum, bộ điều khiển chuyển byte đầu tiên vào bộ nhớ chính tại địa chỉ được mô tả bởi địa chỉ bộ nhớ DMA.
- Sau đó nó tăng địa chỉ DMA và giảm số bytes phải chuyển. Quá trình này lặp cho tới khi số bytes phải chuyển bằng 0, và bộ điều khiển tạo một ngắt. Như vậy không cần phải copy khối vào trong bộ nhớ, nó đã hiện hữu trong bộ nhớ.

8. Có thể thực hiện I/O mà không sử dụng điều khiển thiết bị không? Tại sao?

- Không. Do chức năng của device drivers là nhận những yêu cầu trừu tượng từ phần mềm nhập/xuất độc lập thiết bị ở lớp trên, và giám sát yêu cầu này thực hiện

9. Tác dụng của buffer, caching trong IO Systems

- Buffers là vùng đệm mục đích để tổ chức và giữ dữ liệu trong một vùng nhớ nào đó cho đến khi dữ liệu này được chuyển qua nơi khác.
- Mục đích của cached là tạo ra một vùng nhớ tốc độ cao nhằm tăng tốc quá trình đọc/ghi file ra đĩa

BÀI TẬP TRÊN LỚP

1. Thế nào là tiến trình IO bound, CPU-bound? Cho ví dụ?

- I/O bound có nghĩa là tốc độ tiến trình bị giới hạn bởi tốc độ của hệ thống con I/O.
VD: một tác vụ xử lý dữ liệu từ đĩa, tính số lượng các dòng trong một tệp có thể bị ràng buộc I/O.
- CPU- bound có nghĩa là tốc độ tiến trình được giới hạn bởi bộ nhớ số lượng có sẵn và tốc độ truy cập bộ nhớ đó. Một nhiệm vụ xử lý một lượng lớn dữ liệu bộ nhớ.
VD: nhân các ma trận lớn, có khả năng là Memory Bound.

2. Phân biệt chương trình và tiến trình

- Chương trình hay còn gọi là phần mềm (software) là một tập hợp những câu lệnh được viết bằng một hoặc nhiều ngôn ngữ lập trình theo một trật tự xác định nhằm tự động thực hiện một số chức năng hoặc giải quyết một bài toán nào đó. Khi thực hiện các chỉ thị này, chương trình sẽ chuyển thành các tiến trình kèm theo các tài nguyên phục vụ cho hoạt động của process.
- Tiến trình là một thực thể điều khiển đoạn mã lệnh có riêng một không gian địa chỉ, có ngăn xếp (stack) riêng rẽ, có bảng chứa các số miêu tả file (file descriptor) được mở cùng tiến trình và đặc biệt là có một định danh PID (process identifier) duy nhất trong toàn bộ hệ thống vào thời điểm tiến trình đang chạy. Để hoàn thành công việc của 1 tiến trình cần các nguồn tài nguyên như: CPU, bộ nhớ chính, các tệp tin và các thiết bị nhập xuất.

3. Tiến trình có thể tạo ra một môi trường để chạy các chương trình hay không?

- Tiến trình CÓ thể tạo ra môi trường để chạy các chương trình

4. Kể tên 5 trạng thái của tiến trình. Khi nào thì tiến trình chuyển sang các trạng thái đó.

- **5 trạng thái của tiến trình:**

- + *Mới tạo(new)*: tiến trình đang được tạo lập.
- + *Running*: các chỉ thị của tiến trình đang được xử lý.

- + **Blocked:** tiến trình chờ được cấp phát tài nguyên, hay chờ một sự kiện xảy ra.
- + **Ready(ready):** tiến trình chờ được cấp phát CPU để xử lý.
- + **Waiting (đợi):** tiến trình phải dừng vì thiếu tài nguyên hoặc chờ một sự kiện nào đó.
- Tiến trình chuyển sang các trạng thái đó khi: tiến trình thay đổi qua những trạng thái đó là do nhiều nguyên nhân như: phải chờ đợi một sự kiện nào đó xảy ra, hay tạo một thao tác nhập xuất hoàn tất, buộc phải dừng hoạt động khi hết thời gian xử lý.

5. Khối quản lý tiến trình (PCB) là gì? PCB bao gồm những gì?

- Khối quản lý tiến trình (PCB) là: là một cấu trúc dữ liệu trong nhân hệ điều hành chứa thông tin cần thiết để quản lý một tiến trình nhất định.
- PCB bao gồm:
 - + Định danh của tiến trình (process identifier hay PID)
 - + Giá trị các thanh ghi của tiến trình, trong đó đáng chú ý là con trỏ chương trình và con trỏ stack
 - + Không gian địa chỉ của tiến trình
 - + Độ ưu tiên (trong đó tiến trình có giá trị cao hơn được ưu tiên trước, ví dụ nice trong các hệ điều hành Unix)
 - + Thông tin kế toán tiến trình, ví dụ như thời điểm thực thi gần nhất, bao nhiêu thời gian CPU đã sử dụng...
 - + Con trỏ tới PCB tiếp theo, nghĩa là con trỏ tới tiến trình tiếp theo được chạy
 - + Thông tin V/R (ví dụ các thiết bị V/R được cấp phát cho tiến trình, danh sách các tệp đang mở...)

6. Phân biệt lập lịch dài hạn, ngắn hạn, trung hạn. Khi nào thì hệ điều hành cần sử dụng tới các kỹ thuật lập lịch này.

- Phân biệt lập lịch dài hạn, ngắn hạn:
 - + Lập lịch dài hạn: Quyết định lựa chọn tác vụ nào được đưa vào hệ thống và nạp những tiến trình của tác vụ đó vào bộ nhớ chính để thực hiện, Chức năng điều phối tác vụ quyết định mức độ đa chương của hệ thống, Được kích hoạt khi: hệ thống tạo lập một tiến trình và có một tiến trình kết thúc xử lý, Điều phối tác vụ có tần suất hoạt động thấp
 - + Lập lịch ngắn hạn: áp dụng với những tiến trình mà mã nguồn của nó đã được đẩy vào bộ nhớ trong (tiến trình đã sẵn sàng nhận giờ phân bổ của CPU), Có tần suất hoạt động cao, sau mỗi lần xảy ra ngắt
 - + Lập lịch trung hạn: một số môi trường mới có cơ chế đưa tiến trình ra ngoài ready queue sao cho đảm bảo I/O bound và CPU bound xấp xỉ nhau, lập lịch trung hạn là kết hợp ngắn hạn và dài hạn
- Hệ điều hành cần sử dụng tới các kỹ thuật lập lịch này khi: ta cần lên lịch

7. Hàm/lệnh fork() thực hiện công việc gì? Hàm/lệnh này trả về gì?

- Hàm lệnh Fork(): cho phép một process (gọi là process cha) tạo ra một process mới (process con). Process con được tạo ra gần như là một bản sao của process cha: lấy từ process cha stack, data, heap và text segment.

- Hàm lệnh fork() sinh ra một process con từ tiến trình đang chạy. Hàm này trả về giá trị là PID của process con về cho chương trình đang chạy và trả về 0 cho process con.

8. Hàm/lệnh wait() thực hiện công việc gì?

- Hàm lệnh wait() sẽ ngừng việc thực thi của process cha lại cho đến khi một trong những process con kết thúc. Thứ hai, khi process kết thúc thì trạng thái kết thúc sẽ được trả về vào tham số của wait().

9. Tiến trình xác sống (zombie) là gì? Khi nào thì xuất hiện tiến trình zombie.

- Tiến trình xác sống (Zombie) là: tiến trình con chạy nhưng không còn hoạt động nữa, sống như đã chết. Tuy nhiên, các tiến trình này có thể gây ra tải cao cho server hoặc chết server của server.
- Xuất hiện tiến trình zombie khi: trong Linux, khi một tiến trình kết thúc, hệ điều hành sẽ không xóa nó khỏi bộ nhớ ngay lập tức. Thay vào đó, Linux vẫn giữ lại mô tả tiến trình (process descriptor) trong bộ nhớ (Mô tả tiến trình chỉ chiếm một lượng nhỏ bộ nhớ). Lúc này, trạng thái của tiến trình sẽ là EXIT_ZOMBIE và “cha mẹ” của tiến trình đó được thông báo rằng tiến trình con đã “chết” với tín hiệu tên là SIGCHLD. Tiến trình cha mẹ sau đó có nghĩa vụ thực thi chức năng wait() với nhiệm vụ đọc trạng thái và thông tin của tiến trình đã chết đó. Sau khi chức năng wait() được gọi, tiến trình Zombie lúc này sẽ được xóa hoàn toàn khỏi bộ nhớ.

10. Tiến trình mồ côi (orphan) là gì? Khi nào thì xuất hiện tiến trình orphans. Tiến trình orphans sẽ nhận tiến trình nào là tiến trình cha (parent)

- Tiến trình mồ côi (orphan) là: Trạng thái của tiến trình con khi tiến trình cha kết thúc trong khi tiến trình con vẫn tồn tại.
- Tiến trình orphans xuất hiện khi: tiến trình cha thực hiện câu lệnh mà không có lệnh wait thì tiến trình con sẽ thành tiến trình mồ côi.
- Tiến trình orphans sẽ nhận tiến trình gốc làm tiến trình cha

11. Kể ra các lợi ích khi sử dụng tiểu trình (Thread)

- Chia sẻ tài nguyên, khả năng mở rộng.
- Chia sẻ được một không gian địa chỉ chung.
- Có thể truy xuất đến cả STACK của những tiểu trình khác trong cùng một tiến trình.

12. Phân biệt lập lịch độc quyền và không độc quyền

- Độc quyền: một tiến trình có thể sử dụng CPU cho đến khi tiến trình kết thúc
- Không độc quyền: sẽ ngừng hoạt động tiến trình hiện hành, trong khi độc quyền sẽ cho phép tiến trình tiếp tục xử lý.

13. Kể tên 2 mô hình/phương thức truyền thông giữa các tiến trình. Nêu ưu nhược điểm của từng mô hình/phương thức.

- 2 mô hình/phương thức truyền thông giữa các tiến trình là: thông qua và bộ nhớ chia sẻ
- Ưu nhược điểm của từng mô hình/phương thức:
 - + Thông qua:
 - Ưu điểm: không bị xung đột.
 - Nhược điểm: không trao đổi được thông tin với dữ liệu lớn.
 - + Bộ chia nhớ chia sẻ:

- Ưu điểm: chia sẻ nhiều loại dữ liệu và với tốc độ cao.
- Nhược điểm: có thể bị xung đột.

14. Thread pool là gì tại sao nên sử dụng Thread Pool?

- Thread pool (một chủ đề) là: tạo ra nhiều luồng tại lúc bắt đầu tiến trình và đặt chúng vào một pool-nơi chúng “ngồi” và đợi việc.
- Phải dùng Thread pool vì:
 - + Các luồng thuộc một tiến trình có thể chia sẻ tài nguyên của tiến trình.
 - + Nhưng trong một số trường hợp mỗi luồng cần dữ liệu riêng.
 - + Dùng luồng để tồn tại sẽ nhanh hơn so với chờ đợi để tạo luồng.
 - + Giới hạn số lượng tồn tại ở một thời điểm. Điều này đặc biệt quan trọng trên các hệ thống không thể hỗ trợ số lượng lớn các luồng cùng lúc.

15. Phân biệt 2 giải thuật lập lịch SJF và SRJF, giải thuật nào thường có thời gian chờ đợi trung bình (WT) ngắn hơn?

- SJF: Khi CPU được cấp, nó sẽ đưa vào tiến trình có chiều dài burst time nhỏ nhất để xử lý. Là một dạng độ ưu tiên đặc biệt với độ ưu tiên $p_i = \text{thời_gian_còn_lại}$ (Processi) Thuật toán shortest-job-first (SJF) Có thể được cài đặt độc quyền hoặc không độc quyền Thuật toán SJF đã được chứng minh là sẽ cho ra thời gian chờ trung bình cho các tiến trình cần xử lý với một số con tối thiểu nhất. Bằng cách chuyển cách tiến trình ngắn hơn lên trước các tiến trình tốn thời gian dài, nó đã giảm một cách đáng kể thời gian chờ cho các tiến trình ngắn, kéo theo giảm rõ rệt thời gian chờ trung bình. SJF thường được dùng trong các bài toán cần xử lý dài hạn nhiều hơn.
- SRJF: Hiểu đơn giản nghĩa của thuật toán này là: đến trước, phục vụ trước. một tiến trình sẽ chiếm giữ CPU cho đến khi nào nó trả CPU, thông qua phương thức ngắt hay yêu cầu nhập/xuất xuất hiện Với thuật toán FCFS thì thời gian đợi trung bình sẽ không phải lúc nào cũng tối ưu nhất cho ta mà còn tùy thuộc vào thứ tự vào của các tiến trình. Phiên hà khi sử dụng trong những hệ thống chia sẻ thời gian. Vì mỗi lần thực hiện, nó yêu cầu toàn quyền sử dụng CPU, và điều này sẽ dễ gây nên sự đổ vỡ cho hệ thống nếu có một người dùng nạp quá nhiều tiến trình mà các tiến trình đó có burst time không hề nhỏ trong phiên sử dụng của mình. * Thời gian chờ đợi trung bình (WT) của Giải thuật lập lịch SRJF ngắn hơn.
- Giải thuật SRJF thường có thời gian chờ đợi trung bình (WT) ngắn hơn.

16. Phương pháp nào để dự đoán thời gian sử dụng CPU của một tiến trình?

- Dùng SRTF để làm cơ sở đánh giá các phương pháp khác (vì là phương pháp tối ưu) về thời gian phản hồi trung bình.
- Adaptive (thích ứng): Dự đoán tương lai bằng cách quan sát quá khứ.
- Nếu trong quá khứ tiến trình (chương trình) thường dùng CPU nhiều (CPU - bound) thì có thể trong tương lai nó sẽ sử dụng nhiều.

- Nếu trong quá khứ tiến trình (chương trình) thường thao tác I/O (I/O bound) thì có thể trong tương lai nó sẽ sử dụng CPU ít.

17. Lập lịch theo độ ưu tiên (Priority) có ưu điểm gì? Hạn chế của giải thuật này là gì? Làm sao để khắc phục hạn chế này?

- Lập lịch theo độ ưu tiên (Priority) có ưu điểm: có thể lựa chọn được processes được thực thi trước nhờ vào độ ưu tiên mà CPU sẽ cấp phát cho process đó thực hiện trước.
- Hạn chế: trì hoãn vô hạn định – processes có độ ưu tiên thấp có thể không bao giờ được thực thi.
- Giải pháp: làm mới (aging) – độ ưu tiên của processes sẽ giảm theo thời gian.

18. Multilevel Feedback Queue Scheduling là giải thuật lập lịch gì? Ưu nhược điểm của giải thuật này?

- Ưu điểm:
 - + Tiến trình có thể di chuyển giữa các queue
 - + Tiến trình sử dụng nhiều CPU burst có thể di chuyển từ hàng đợi có mức ưu tiên cao xuống hàng đợi có mức ưu tiên thấp hơn.
- Hạn chế:
 - + Không linh động, một tiến trình không thể di chuyển giữa các hàng đợi.
 - + Dễ xảy ra trường hợp “đói” CPU.

19. Với giải thuật RR (Round Robin). Nên làm gì để chọn ra thời gian quantum phù hợp?

- Mỗi tiến trình được gán cho một độ ưu tiên tương ứng, tiến trình có độ ưu tiên cao nhất sẽ được chọn để cấp phát CPU đầu tiên. Độ ưu tiên có thể được định nghĩa nội tại hay nhờ vào các yếu tố bên ngoài. Độ ưu tiên nội tại sử dụng các đại lượng có thể đo lường để tính toán độ ưu tiên của tiến trình, ví dụ các giới hạn thời gian, nhu cầu bộ nhớ... Độ ưu tiên cũng có thể được gán từ bên ngoài dựa vào các tiêu chuẩn do hệ điều hành như tầm quan trọng của tiến trình, loại người sử dụng sở hữu tiến trình...
- Giải thuật điều phối với độ ưu tiên có thể theo nguyên tắc độc quyền hay không độc quyền. Khi một tiến trình được đưa vào danh sách các tiến trình sẵn sàng, độ ưu tiên của nó được so sánh với độ ưu tiên của tiến trình hiện hành đang xử lý. Giải thuật điều phối với độ ưu tiên và không độc quyền sẽ thu hồi CPU từ tiến trình hiện hành để cấp phát cho tiến trình mới nếu độ ưu tiên của tiến trình này cao hơn tiến trình hiện hành. Một giải thuật độc quyền sẽ chỉ đơn giản chèn tiến trình mới vào danh sách sẵn sàng, và tiến trình hiện hành vẫn tiếp tục xử lý hết thời gian dành cho nó.

20. Thay trang

	FIFO	LRU	OPT
--	------	-----	-----

Cơ chế	Thay trang theo hàng đợi, vào trước thì thay trước	Thay vào trang mà dùng trong quá khứ xa nhất	Thay vào trang mà dùng trong tương lai xa nhất
Ưu	Đơn giản	Tối ưu	Tối ưu
Nhược	Không tối ưu, thay trang nhiều lần	Tốn bộ nhớ	Phải dự báo trước

21. Tính toán thời gian truy cập trung bình

- Công thức:

$$\text{Access Latency} = \text{Average access time} = \text{average seek time} + \text{average latency}$$

$$\text{Average I/O time} = \text{average access time} + (\text{amount to transfer} / \text{transfer rate}) + \text{controller overhead}$$

- Đĩa từ:

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

Ví dụ: Chuyển khối 4kB vào đĩa 7200RPM với thời gian tìm kiếm trung bình 5ms, tốc độ truyền 1Gb/s, với controller overhead 0,1 ms

Ta có: spindle=7200RPM => Average latency=4,17 ms

$$\text{Average access time} = \text{average seek time} + \text{average latency} = 5 + 4,17 = 9,17 \text{ ms}$$

$$\text{Average I/O time} =$$

$$= \text{average access time} + (\text{amount to transfer} / \text{transfer rate}) + \text{controller overhead}$$

$$= 9,17 + (4 \times 8) / (1 \times 2^{10} \times 2^{10}) \times 10^3 + 0,1$$

$$= 9,17 + 0,031 + 0,1 = 9,301 \text{ ms}$$

22. I/O

Có 100 khối	Liên tục	Liên kết	Chỉ mục
Thêm 1 khối ở đầu	200(đọc và ghi 100 khối, dịch về sau)+ 1(ghi khối mới)	1(ghi khối mới)	1
Thêm sau khối 50	100(dịch 50 khối sau khối 50)+1	49(đọc 49 khối đầu để tìm khối 50) + 2(sửa khối 50) + 1(ghi khối sau khối 50)	1
Thêm 1 khối vào cuối	1	2(sửa khối 100)+ 1(ghi khối sau khối 100)	1

