TRẮC NGHIỆM MÔN HỆ ĐIỀU HÀNH

# CHƯƠNG 4: TẮC NGHẼN (DEADLOCK)



1. Deadlock là gì:
   1. là hiện tượng một tiến trình luôn trong trạng thái chờ
   2. là hiện tượng một tiến trình bị dừng khi đang hoạt động bình thường
   3. là hiện tượng hệ điều hành tự động tắt máy tính
   4. là hiện tượng các chương trình không cho phép người dùng truy cập vào tài nguyên máy

tính

1. Nguyên nhân nào dưới đây có thể gây ra hiện tượng Deadlock (tắc nghẽn)?
   1. Hệ thống bị thiếu bộ nhớ.
   2. Nhiều tiến trình chờ một tài nguyên.
   3. Tiến trình chờ tài nguyên lâu vô tận.
   4. Số tài nguyên nhỏ hơn số tiến trình.
2. Các bước mà một tiến trình thực hiện để sử dụng tài nguyên:
   1. Yêu cầu, sử dụng, giải phóng
   2. Tìm kiếm, yêu cầu, sử dụng, giải phóng
   3. Sử dụng, chuyển tiếp, giải phóng
   4. Sử dụng, giải phóng
3. Trong đồ thị cấp tài nguyên RAG các cạnh biểu thị mối tương quan giữa tiến trình (P) và

tài nguyên (R). Hãy cho biết RAG có những loại cạnh nào?

* 1. Cạnh yêu cầu và cạnh độc lập.
  2. Cạnh yêu cầu và cạnh đã cấp phát.
  3. Cạnh lấy và cạnh trả lại.
  4. Cạnh đòi hỏi và cạnh trả lại.

1. Cạnh yêu cầu trong đồ thị cấp tài nguyên có dạng:
   1. nét liền từ tiến trình tới tài nguyên.
   2. nét đứt đoạn từ tài nguyên tới tiến trình.
   3. nét liền từ tài nguyên tới tiến trình.
   4. nét đứt đoạn từ tiến trình tới tài nguyên.
2. Cho biết ý nghĩa nào đúng cho đồ thị cấp tài nguyên RAG biểu thị như hình vẽ:



* 1. Tiến trình P1 yêu cầu tài nguyên R1.
  2. Tài nguyên R1 đã cấp phát cho tiến trình P1.
  3. Tiến trình P1 được chuyển vào tài nguyên R1.
  4. Tiến trình P1 gởi thông báo cho tài nguyên R1.

1. Cho biết ý nghĩa nào đúng cho đồ thị cấp tài nguyên RAG biểu thị như hình vẽ:



* 1. Tiến trình P1 yêu cầu tài nguyên R1.
  2. Tài nguyên R1 đã cấp phát cho tiến trình P1.
  3. Tiến trình P1 được chuyển vào tài nguyên R1.
  4. Tiến trình P1 gởi thông báo cho tài nguyên R1.

1. Cho biết đồ thị cấp tài nguyên RAG như hình vẽ dưới đây biểu thị cho trường hợp nào?



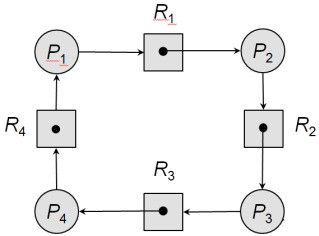
* 1. Loại trừ hỗ tương.
  2. Giữ và chờ.
  3. Không cho trưng dụng tài nguyên.
  4. Tồn tại chu trình.

1. Cho biết đồ thị cấp tài nguyên RAG như hình vẽ dưới đây biểu thị cho trường hợp nào?



* 1. Loại trừ hỗ tương.
  2. Giữ và chờ.
  3. Không cho trưng dụng tài nguyên.
  4. Tồn tại chu trình.

1. Cho biết đồ thị cấp tài nguyên RAG như hình vẽ dưới đây biểu thị cho trường hợp nào?



* 1. Loại trừ hỗ tương.
  2. Giữ và chờ.
  3. Không cho trưng dụng tài nguyên.
  4. Tồn tại chu trình.

1. Trong quá thực thi tiến trình, những trường hợp nào là điều kiện xảy ra Deadlock:
   1. Loại trừ hỗ tương & Giữ và chờ cấp thêm tài nguyên
   2. Giữ và chờ cấp thêm tài nguyên & Không cho trưng dụng tài nguyên.
   3. Loại trừ hỗ tương & Tồn tại chu trình trong đồ thị cấp phát tài nguyên
   4. Loại trừ hỗ tương & Giữ và chờ cấp thêm tài nguyên & Tồn tại chu trình trong đồ thị cấp

phát tài nguyên & Không cho trưng dụng tài nguyên.

1. Các trường hợp sau có thể diễn ra trong quá thực thi tiến trình:
2. Loại trừ tương hỗ,
3. Giữ và chờ cấp thêm tài nguyên,
4. Không cho trưng dụng tài nguyên,
5. Đợi xoay vòng.



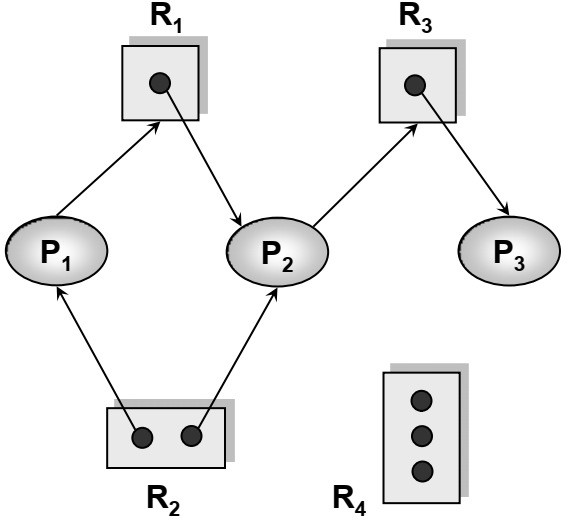
Khả năng xảy ra Deadlock (tắc nghẽn) hệ thống do những trường hợp nào gây ra?

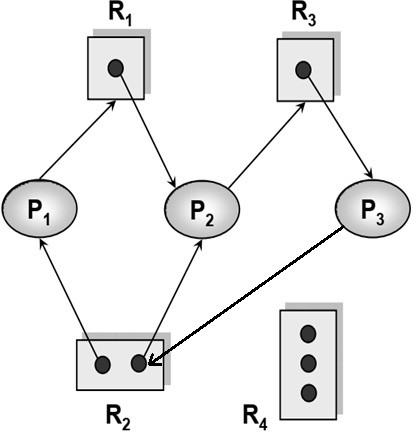
* 1. Trường hợp (1) và (2).
  2. Trường hợp (1), (2) và (3).
  3. Trường hợp (1) và (4).
  4. Tất cả 4 trường hợp trên.

1. Nếu đồ thị cấp phát tài nguyên RAG không chứa chu trình thì hệ thống:
   1. chắc chắn không xảy ra Deadlock (tắc nghẽn).
   2. chắc chắn xảy ra Deadlock (tắc nghẽn).

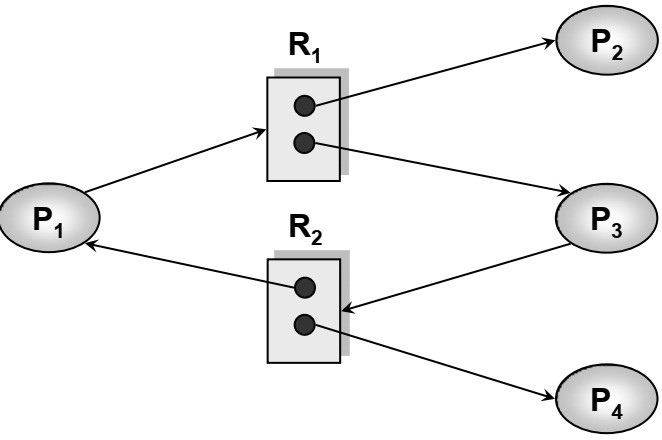
**B**. có thể xảy ra Deadlock (tắc nghẽn).

**D**. có thể không xảy ra Deadlock (tắc nghẽn).

1. Tại thời điểm t, hệ thống có đồ thị cấp phát tài nguyên RAG như hình dưới. Hãy chọn đánh giá đúng:
   1. chắc chắn không xảy ra Deadlock (tắc nghẽn).
   2. chắc chắn xảy ra Deadlock (tắc nghẽn).
   3. sẽ xảy ra Deadlock nếu P1 không trả tài nguyên R2.
   4. sẽ xảy ra Deadlock nếu P2 không trả tài nguyên R1.
2. Tại thời điểm t, hệ thống có đồ thị cấp phát tài nguyên RAG như hình dưới. Hãy chọn đánh giá đúng:



* 1. không xảy ra Deadlock (tắc nghẽn).
  2. chắc chắn xảy ra Deadlock (tắc nghẽn).
  3. sẽ xảy ra Deadlock nếu P1 không trả tài nguyên R2.
  4. sẽ xảy ra Deadlock nếu P2 không trả tài nguyên R1.

1. Tại thời điểm t, hệ thống có đồ thị cấp phát tài nguyên RAG như hình dưới (P3 yêu cầu instance phía dưới của R2). Hãy chọn đánh giá đúng:
   1. chắc chắn không xảy ra Deadlock (tắc nghẽn).
   2. chắc chắn xảy ra Deadlock (tắc nghẽn).
   3. sẽ xảy ra Deadlock nếu P4 không trả instance trong R2.
   4. không xảy ra Deadlock nếu P2 không trả tài nguyên R1.
2. Trong hệ thống tài nguyên có nhiều thực thể (instance), nếu đồ thị cấp phát tài nguyên

RAG chứa chu trình thì hệ thống sẽ thế nào?

* 1. chắc chắn không xảy ra Deadlock (tắc nghẽn).
  2. có thể có hoặc không xảy ra Deadlock (tắc nghẽn).
  3. chắc chắn xảy ra Deadlock (tắc nghẽn).
  4. Deadlock (tắc nghẽn) ngay lập tức.

1. Hệ điều hành KHÔNG dùng phương pháp nào dưới đây cho việc giải quyết Deadlock?
   1. Phương pháp Ngăn chặn (preventing)
   2. Phương pháp Phòng tránh (avoiding)
   3. Phương pháp Phát hiện và Khôi phục
   4. Phương pháp Định thời (Scheduling)
2. Các trường hợp sau có thể diễn ra trong quá thực thi tiến trình:
3. Loại trừ tương hỗ,
4. Giữ và chờ cấp thêm tài nguyên,
5. Không cho trưng dụng tài nguyên,
6. Đợi xoay vòng.

Để NGĂN CHẶN Deadlock, Hệ điều hành cần phải ngăn chặn không cho xảy ra trường hợp nào?

* 1. Ngăn chặn trường hợp (1) và (3).
  2. Ngăn chặn trường hợp (2) và (4).
  3. Ngăn chặn 1 trong 4 trường hợp.
  4. Ngăn chặn cả 4 trường hợp.



1. Hệ điều hành ngăn chặn deadlock bằng cách đặt ra quy định: “Nếu tiến trình đang giữ tài nguyên thì phải hoàn trả trước khi yêu cầu tài nguyên khác”. Quy định này là:
   1. Ngăn chặn điều kiện “loại trừ tương hỗ” (Mutual exclusion)
   2. Ngăn chặn điều kiện “giữ và chờ tài nguyên” (Hold and Wait)
   3. Ngăn chặn điều kiện “không trưng dụng” (Non-preemption)
   4. Ngăn chặn điều kiện “chờ xoay vòng” (Circular wait)
2. Hệ điều hành ngăn chặn deadlock bằng cách đặt ra quy định:

* Nếu: tiến trình A đang giữ tài nguyên R1 và A yêu cầu thêm tài nguyên R2 thì:
* Thu hồi mọi tài nguyên R1 mà A đang giữ.
* Cấp lại tài nguyên R1 và R2 cho A nếu cả 2 tài nguyên này không có tranh chấp. Quy định này là:
  1. Ngăn chặn điều kiện “loại trừ tương hỗ” (Mutual exclusion)
  2. Ngăn chặn điều kiện “giữ và chờ tài nguyên” (Hold and Wait)
  3. Ngăn chặn điều kiện “không trưng dụng” (Non-preemption)
  4. Ngăn chặn điều kiện “chờ xoay vòng” (Circular wait)

1. Hệ điều hành ngăn chặn deadlock bằng cách đặt ra quy định:

* Nếu: tiến trình P đang giữ các tài nguyên Ri và P yêu cầu thêm tài nguyên Rj, thì:
* P phải trả lại các tài nguyên Ri với thứ tự (i > j) Quy định này là:
  1. Ngăn chặn điều kiện “loại trừ tương hỗ” (Mutual exclusion)
  2. Ngăn chặn điều kiện “giữ và chờ tài nguyên” (Hold and Wait)
  3. Ngăn chặn điều kiện “không trưng dụng” (Non-preemption)
  4. Ngăn chặn điều kiện “chờ xoay vòng” (Circular wait)

1. Nếu Hệ điều hành đặt số thứ tự cho mỗi tài nguyên; đồng thời quy định tiến trình chỉ được

yêu cầu tài nguyên theo thứ tự tăng lên. Quy định này ngăn chặn trường hợp nào?

* 1. Điều kiện Loại trừ.
  2. Điều kiện Đợi xoay vòng.
  3. Điều kiện Giữ và chờ cấp thêm tài nguyên.
  4. chờ cấp thêm tài nguyên.

1. Một hệ thống điều phối tiến trình sao cho chuỗi thứ tự tiến trình được cấp phát tài nguyên đạt trạng thái an toàn (safe) thì sẽ đảm bảo được điều gì?
   1. tránh được Deadlock (tắc nghẽn).
   2. cấp vừa hết các tài nguyên.
   3. không xảy ra “Đợi xoay vòng”.
   4. cấp được nhiều tài nguyên nhất.
2. Chuỗi thứ tự tiến trình được cấp phát tài nguyên <P1, P2, …, Pn> được cho là an toàn (safe) nếu: với mỗi tiến trình Pi, các tài nguyên mà Pi yêu cầu được thoả mãn bởi số tài nguyên sẵn có (+) số tài nguyên được giữ bởi tất cả tiến trình Pj. Hãy cho biết phát biểu nào đúng?
   1. Pj cũng là tiên trình Pi trong chuỗi.
   2. Pj là tiên trình bất kỳ trong chuỗi.
   3. Tiến trình Pj < Pi trong chuỗi.
   4. Tiến trình Pj > Pi trong chuỗi.
3. Các giải thuật tránh Deadlock dựa trên ý tưởng nào?
   1. Luôn kiểm tra đồ thị có chu trình không.
   2. Luôn giữ chuỗi thứ tự tiến trình được cấp phát tài nguyên ở trạng thái an toàn.
   3. Luôn cấp tài nguyên ít nhất có thể.
   4. cấp tài nguyên nhiều nhất có thể.
4. Để tránh Deadlock, trường hợp nào sau đây hệ thống không cấp tài nguyên cho tiến trình mà bắt phải chờ tới khi giải phóng thêm tài nguyên, dù tài nguyên đang sẵn có:
   1. Khi dẫn tới đồ thị tài nguyên có chu trình.
   2. Khi số tài nguyên chỉ còn vừa đủ để cấp.
   3. Khi số tiến trình đợi nhiều hơn số tài nguyên.
   4. Khi chuỗi thứ tự tiến trình sẻ bị chuyển sang không an toàn (unsafe).
5. Đồ thị chờ tài nguyên (Wait for Graph) được phát triển từ là đồ thị cấp phát tài nguyên

(Resource allocation graph – RAG) với điều chỉnh nào?

* 1. bỏ đi các đỉnh biểu diễn tài nguyên.
  2. bỏ đi các đỉnh biểu diễn tiến trình.
  3. bỏ đi các cạnh yêu cầu và các cạnh cấp phát.
  4. bỏ đi các cạnh cấp phát và các cạnh đòi hỏi.

1. Phương pháp tránh Deadlock bằng cách sử dụng đồ thị chờ tài nguyên (Wait for Graph)

được sử dụng trong trường hợp hệ thống có loại tài nguyên nào?

* 1. Tài nguyên có nhiều thực thể.
  2. Tài nguyên có một thực thể.
  3. Tài nguyên không có thực thể.
  4. Tài nguyên có một hoặc nhiều thực thể.

1. Giả sử, hệ thống có **12** tài nguyên. Có 3 tiến trình A, B, C đang chạy. Tại thời điểm, trạng thái sử dung tài nguyên các tiến trình như sau:

Hãy chọn phát biểu ĐÚNG khi đánh giá an toàn cho chuỗi tiến trình nếu cho tiến trình C thực thi trước?

* 1. Sẽ là an toàn (safe) vì C đang cần thêm tài nguyên nhiều nhất
  2. Sẽ là an toàn (safe) vì C đang giữ tài nguyên ít hơn số tài nguyên sẵn có.
  3. Sẽ là không an toàn (unsafe) vì số tài nguyên C cần thêm lớn hơn số tài nguyên sẵn có.
  4. Sẽ là không an toàn (unsafe) vì số tài nguyên C cần tối đa lớn hơn số tài nguyên sẵn có

1. Giả sử, hệ thống có **12** tài nguyên. Có 3 tiến trình A, B, C đang chạy. Tại thời điểm, trạng

thái sử dung tài nguyên các tiến trình như sau:



Hãy chọn phát biểu ĐÚNG khi đánh giá an toàn cho chuỗi tiến trình nếu cho tiến trình B

thực thi trước?

* 1. Sẽ là an toàn (safe) vì số tài nguyên B cần thêm ít hơn số tài nguyên sẵn có.
  2. Sẽ là an toàn (safe) vì B đang giữ tài nguyên ít hơn số tài nguyên sẵn có.
  3. Sẽ là không an toàn (unsafe) số tài nguyên B cần tối đa là ít nhất.
  4. Sẽ là không an toàn (unsafe) số tài nguyên B cần tối đa lớn hơn số tài nguyên sẵn có.

1. Giả sử, hệ thống có **12** tài nguyên. Có 3 tiến trình A, B, C đang chạy. Tại thời điểm, trạng thái sử dung tài nguyên các tiến trình như sau:



Hãy chọn phát biểu ĐÚNG khi đánh giá an toàn cho chuỗi tiến trình nếu cho tiến trình B thực thi trước sau đó đến tiến trình A?

* 1. Sẽ là an toàn (safe) vì số tài nguyên A cần thêm lớn hơn số tài nguyên sẵn có.
  2. Sẽ là an toàn (safe) vì số tài nguyên A cần thêm ít hơn (số sẵn + số B đang giữ).
  3. Sẽ là không an toàn (unsafe) số tài nguyên A cần tối đa là lớn nhất.
  4. Sẽ là không an toàn (unsafe) số tài nguyên A ít hơn số cần thêm của B.

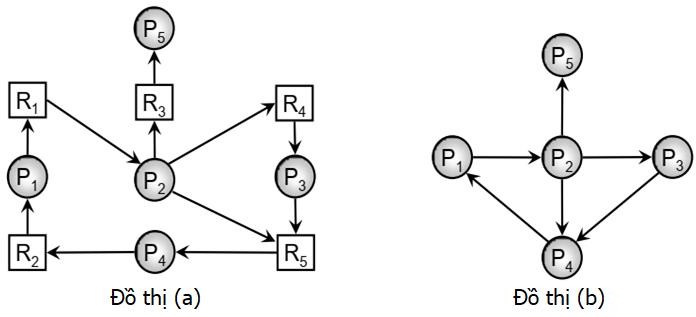
1. Giả sử, hệ thống có **12** tài nguyên. Có 3 tiến trình A, B, C đang chạy. Tại thời điểm, trạng

thái sử dung tài nguyên các tiến trình như sau:

Hãy cho biết, chuỗi thực thi tiến trình nào dưới đây là an toàn (safe)?

* 1. Chuỗi thứ tự tiến trình: A > B > C
  2. Chuỗi thứ tự tiến trình: A > C > B
  3. Chuỗi thứ tự tiến trình: B > A > C
  4. Chuỗi thứ tự tiến trình: B > C > A

1. Xem ảnh 2 loại đồ thị dưới đây và chọn phát biểu đúng.



* 1. (a) là đồ thị cấp phát tài nguyên; (b) là đồ thị chờ tài nguyên.
  2. (a) là đồ thị chờ tài nguyên; (b) là đồ thị cấp phát tài nguyên.
  3. (a) là đồ thị cấp phát tài nguyên; (b) là đồ thị chuỗi thứ tự tiến trình.
  4. (a) là đồ thị chờ tài nguyên; (b) là đồ thị chuỗi thứ tự tiến trình.

1. Sau khi phát hiện Deadlock, Hệ điều hành dùng nhiều phương pháp để tự phá deadlock. Phương pháp nào dưới đây là KHÔNG ĐÚNG?
   1. Chấm dứt tiến trinh có tham gia vào Deadlock.
   2. Thu hồi tài nguyên của tiến trình tham gia vào Deadlock.
   3. Rollback (quay lại từ đầu) cho tiến trình tham gia vào Deadlock
   4. Dùng Task Manager để Kill tiến trình tham gia vào Deadlock.
2. Trong trường hợp hệ thống dùng loại tài nguyên có một thực thể, giải thuật nào sau đây để

phát hiện Deadlock?

* 1. Giải thuật chạy lại tiến trình (Rollback).
  2. Giải thuật đồ thị cấp phát tài nguyên (Resource allocation graph).
  3. Giải thuật đồ thị chờ tài nguyên (Wait for Graph).
  4. Giải thuật nhà băng (Banker).

1. Trong trường hợp hệ thống dùng loại tài nguyên có nhiều thực thể, giải thuật nào sau đây để phát hiện Deadlock?
   1. Giải thuật chạy lại tiến trình (Rollback).
   2. Giải thuật đồ thị cấp phát tài nguyên (Resource allocation graph).
   3. Giải thuật đồ thị chờ tài nguyên (Wait for Graph).
   4. Giải thuật nhà băng (Banker).
2. Hệ điều hành sử dụng thuật toán phá hủy “chu trình chò” (Circular wait) của đồ thị cấp phát tài nguyên RAG. Giải thuật này nhằm giải quyết Deadlock theo phương pháp nào?
   1. Phương pháp Ngăn chặn (preventing)
   2. Phương pháp Phòng tránh (avoiding)
   3. Phương pháp Phát hiện và Khôi phục
   4. Phương pháp Bỏ qua Deadlock.
3. Phương pháp giải quyết Deadlock nào mà Hệ điều hành sử dụng giải thuật “Đồ thị chờ”

(Wait for Graph)?

* 1. Phương pháp Ngăn chặn (preventing)
  2. Phương pháp Phòng tránh (avoiding)



* 1. Phương pháp Phát hiện và Khôi phục
  2. Phương pháp Bỏ qua Deadlock.

1. Phương pháp giải quyết Deadlock nào mà Hệ điều hành sử dụng việc đánh giá tính an toàn của chuỗi tiến trình là “safe” hay “unsafe”?
   1. Phương pháp Ngăn chặn (preventing)
   2. Phương pháp Phòng tránh (avoiding)
   3. Phương pháp Phát hiện và Khôi phục
   4. Phương pháp Bỏ qua Deadlock.

# CHƯƠNG 5: QUẢN LÝ BỘ NHỚ (MEMORY MANAGEMENT)

1. Để quản lý không gian bộ nhớ cấp phát cho tiến trình, hệ điều hành sử dụng 2 thanh ghi

nào?

* 1. **Base register & Limit register**
  2. Address register & Base register
  3. Start register & Limit register
  4. Address register & Full register

1. Một hệ thống có: kernel Hệ điều hành = 4MB, chương trình P1 = 3MB, chương trình P1 = 6MB, bộ nhớ vật lý = 12MB. Hãy cho biết không gian bộ nhớ vật lý tối đa có thể cấp cho chương trình là bao nhiêu?
   1. 12 MB
   2. **8 MB**
   3. 6 MB
   4. 4 MB
2. Để quản lý bộ nhớ, Hệ điều hành cần sự hỗ trợ từ thiết bị phần cứng nào?
   1. CPU (Center Processing Unit) và RAM (Random Access Memory)
   2. CPU (Center Processing Unit) và HDD (Hard Disk Drive)
   3. **CPU (Center Processing Unit) và MMU (Memory Management Unit)**
   4. MMU (Memory Management Unit) và RAM (Random Access Memory)
3. Khái niệm “không gian địa chỉ vật lý” (Physical address) là gì?
   1. **là vùng địa chỉ trỏ đến toàn bộ các Bytes trên bộ nhớ vật lý.**
   2. là vùng địa chỉ trỏ đến toàn bộ các Bytes trong vùng nhớ cấp cho process.
   3. là vùng địa chỉ trỏ đến toàn bộ các Bytes trên đĩa cứng vật lý.
   4. là vùng địa chỉ trỏ đến toàn bộ các Bytes trên bộ nhớ ảo.
4. Khái niệm “không gian địa chỉ luận lý” (Logical address) là gì?
   1. là vùng địa chỉ trỏ đến toàn bộ các Bytes trên bộ nhớ vật lý.
   2. **là vùng địa chỉ trỏ đến toàn bộ các Bytes trong vùng nhớ cấp cho process.**
   3. là vùng địa chỉ trỏ đến toàn bộ các Bytes trên đĩa cứng cấp cho process.
   4. là vùng địa chỉ trỏ đến toàn bộ các Bytes trên bộ nhớ ảo.
5. Người dùng sử dụng một ngôn ngữ lập trình để viết (code) một phần mềm. Để có thể thực thi, anh ta tiến hành biên dịch (Compile) phần mềm đó. Không gian địa chỉ của phần mềm sau khi được chuyển đổi bởi trình biên dịch thuộc loại nào?
   1. Symbolic address.
   2. **Relocatable address**
   3. Physical address
   4. Logical address.
6. Một chương trình sau khi được biên dịch (Compile), bộ nạp (Loader) tiến hành nạp chương trình đó vào bộ nhớ. Không gian địa chỉ do Loader chuyển đổi từ chương trình dịch thuộc loại nào?
   1. Symbolic address.



* 1. Relocatable address
  2. **Physical address**
  3. Logical address.

1. Để nạp một lệnh và dữ liệu của một tiến trình đang chạy trong bộ nhớ, CPU sẽ sử dụng loại địa chỉ nào?
   1. Địa chỉ danh biểu.
   2. Địa chỉ ảo.
   3. Địa chỉ luận lý.
   4. **Địa chỉ vật lý.**
2. Trong quá trình viết (code) một phần mềm, người lập trình sử dụng nhiều không gian địa chỉ cho lệnh và dữ liệu. Những không gian địa chỉ này sẽ được chuyển đổi thành “địa chỉ tái định vị” vào thời điểm nào?
   1. **Thời gian biên dịch chương trình**
   2. Thời gian nạp chương trình
   3. Thời gian thực thi chương trình
   4. Biên dịch, nạp và thực thi chương trình
3. Tại thời điểm biên dịch (Compile), nếu biết trước địa chỉ bộ nhớ vật lý được cấp cho tiến trình thì Compiler có thể liên kết địa chỉ như thế nào?
   1. biên dịch ngay “Relocatable address” thành “Physical address”.
   2. biên dịch ngay “Physical address” thành “Relocatable address”.
   3. biên dịch ngay “Symbolic address” thành “Relocatable address”.
   4. **biên dịch ngay “Symbolic address” thành “Physical address”.**
4. Một tiến trình được Hệ điều hành cấp 1 không gian bộ nhớ vật lý, giới hạn bởi “Based address” và “limit address”. Khi nạp chương trình vào bộ nhớ, bộ Loader phải chuyển đổi không gian chương trình thành địa chỉ vật lý bằng cách nào?
   1. sử dụng “Based address” (+) “Physical address”
   2. **sử dụng “Based address” (+) “Relocatable address”**
   3. sử dụng “Limit address” (+) “Physical address”
   4. sử dụng “Limit address” (+) “Relocatable address”
5. Tại thời điểm thực thi tiến trình, một process có thể được di chuyển từ vùng nhớ này sang vùng nhớ khác. Việc chuyển đổi không gian địa chỉ này được thực hiện bằng cách nào?
   1. dùng bộ quản lý bộ nhớ MMU
   2. dùng kỹ thuật phân trang (Paging).
   3. dùng kỹ thuật phân đoạn (Segmentation)
   4. **dùng cả 3 loại trên.**
6. Tại thời điểm thực thi, việc ánh xạ địa chỉ để di chuyển tiến trình giữa không gian bộ nhớ vật lý (physical memory) và không gian bộ nhớ ảo (virtual memory) được thực hiện bởi:
   1. Trình biên dịch.
   2. Bộ nạp.
   3. **Bộ quản lý bộ nhớ MMU.**
   4. Trình soạn thảo liên kết.
7. Bộ nhớ ảo (Virtual Memory) sử dụng thiết bị nào để lưu trữ các phần của tiến trình?
   1. bộ nhớ RAM.
   2. **bộ nhớ phụ.**
   3. các thanh ghi CPU.
   4. bộ nhớ ROM.
8. Nhằm tối ưu sử dụng bộ nhớ, Hệ điều hành biên dịch sẵn các hàm, thủ tục… thành các External module dùng chung cho nhiều tiến trình khác nhau. Tên gọi của kỹ thuật này là gì?
   1. Static linking
   2. **Dynamic linking**
   3. Overlay
   4. Swapping
9. Nhằm tối ưu sử dụng bộ nhớ, Hệ điều hành chỉ giữ lại trong bộ nhớ những lệnh / dữ liệu cần dùng của tiến trình, không nạp vào bộ nhớ lệnh / dữ liệu chưa đến lượt xử lý. Tên gọi của kỹ thuật này là gì?
   1. Static linking
   2. Dynamic linking
   3. **Overlay**
   4. Swapping
10. Nhằm tối ưu sử dụng bộ nhớ, Hệ điều hành đưa một phần tiến trình ra khỏi bộ nhớ chính, lưu tạm trên ổ cứng. Khi tiến trình thì nạp vào. Tên gọi của kỹ thuật này là gì?
    1. Static linking
    2. Dynamic linking
    3. Overlay
    4. **Swapping**
11. Trong quản lý bộ nhớ, cơ chế phủ lấp (overlay) thực hiện nhằm mục đích nào?
    1. **Cấp phát bộ nhớ có dung lượng nhỏ hơn so với dung lượng tiến trình.**
    2. Ánh xạ tại thời điểm thực thi từ địa chỉ ảo tới địa chỉ vật lý
    3. Chương trình chính được nạp vào bộ nhớ và được thực thi
    4. Nạp các chương trình vào vùng nhớ có địa chỉ vật lý
12. Kỹ thuật Swapping tiến trình là việc hoán chuyển các phần của tiến trình giữa 2 thiết bị

nào?

* 1. Bộ nhớ chính và bộ nhớ đệm cache.
  2. Bộ nhớ đệm và đĩa cứng.
  3. **Bộ nhớ chính và vùng lưu trữ phụ.**
  4. Bộ nhớ chính và đĩa cứng.

1. Kỹ thuật Swapping tiến trình giải quyết được vần đề nào? vấn đề tổng không gian tiến

trình:



* 1. **tổng không gian tiến trình lớn hơn tổng bộ nhớ vật lý.**
  2. tổng không gian tiến trình nhỏ hơn tổng bộ nhớ vật lý.
  3. tổng không gian tiến trình lớn hơn các thanh ghi CPU.
  4. tổng không gian tiến trình lớn hơn bộ nhớ đệm cache.

1. Hiện tượng bộ nhớ có những vùng trống rời rạc, không chứa tiến trình nào được gọi là gì?
   1. Bộ nhớ phân tán.
   2. Bộ nhớ không liên tục.
   3. Phân mảnh nội.
   4. **Phân mảnh ngoại.**
2. Hiện tượng bộ nhớ có những vùng trống bên trong không gian đã cấp cho tiến trình được gọi là gì?
   1. Bộ nhớ phân tán.
   2. Bộ nhớ không liên tục.
   3. **Phân mảnh nội.**
   4. Phân mảnh ngoại.
3. Hình dưới mô tả một bộ nhớ đã được cấp phát cho các tiến trình. Hãy cho biết đó là kỹ thuật cấp phát bộ nhớ nào?
   1. **Kỹ thuật chia khối nhớ cố định (fixed partitioning)**
   2. Kỹ thuật chia khối nhớ động (dynamic partitioning)
   3. Kỹ thuật phân trang (Paging)
   4. Kỹ thuật phân đoạn (Segmentation)
4. Hình dưới mô tả một bộ nhớ đã được cấp phát cho các tiến trình. Hãy cho biết đó là kỹ thuật cấp phát bộ nhớ nào?



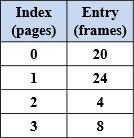
* 1. Kỹ thuật chia khối nhớ cố định (fixed partitioning)
  2. **Kỹ thuật chia khối nhớ động (dynamic partitioning)**
  3. Kỹ thuật phân trang (Paging)
  4. Kỹ thuật phân đoạn (Segmentation)

1. Hình dưới mô tả một bộ nhớ đã được cấp phát cho các tiến trình. Địa chỉ truy cập bộ nhớ vật lý của mỗi tiến trình có dạng (**p,d**). Hãy cho biết đó là kỹ thuật cấp phát bộ nhớ nào?
   1. Kỹ thuật chia khối nhớ cố định (fixed partitioning)
   2. Kỹ thuật chia khối nhớ động (dynamic partitioning)
   3. **Kỹ thuật phân trang (Paging)**
   4. Kỹ thuật phân đoạn (Segmentation)
2. Hệ điều hành sử dụng kỹ thuật cấp phát bộ nhớ liên tục sẽ đễ xảy ra hiện tượng phân mảnh trong bộ nhớ. Nguyên nhân là do:
   1. **Các tiến trình thường xuyên được “nạp” rồi “xóa” (kết thúc) khỏi bộ nhớ.**
   2. Cấp phát bộ nhớ theo các đơn vị khối.
   3. Sử dụng một hàng đợi cấp phát bộ nhớ.
   4. Các tiến trình thường xuyên được “nạp” rồi “xóa” (kết thúc) khỏi CPU.
3. Giải pháp nào giảm thiểu hiện tượng phân mảnh ngoại cho bộ nhớ:
   1. Sử dụng kỹ thuật liên kết khối (compaction).
   2. Sử dụng kỹ thuật phân trang bộ nhớ.
   3. Sử dụng kỹ thuật phân đoạn bộ nhớ.
   4. **Tất cả đúng.**
4. Hệ điều hành dùng kỹ thuật chia khối nhớ động (dynamic partitioning), sau thời gian hoạt động sẽ tạo ra nhiều khối nhớ trống rời rạc, có kích thước không giống nhau. Các giải thuật nào được sử dụng để cấp phát bộ nhớ cho tiến trình mới?
   1. FIFO, SJF, Round-robin.
   2. Giải thuật nhà băng.
   3. **First-fit, Best-fit, Worst-fit.**
   4. FIFO, SJF.
5. Hệ điều hành dùng kỹ thuật phân trang bộ nhớ, không gian địa chỉ vật lý của một tiến trình là thường ở dạng nào?
   1. liên tục.
   2. **không liên tục.**
   3. chia thành từng khối có kích thước tùy ý.
   4. Nằm rải rác trên đĩa cứng.
6. Trong kỹ thuật phân trang (paging), bộ nhớ vật lý được chia thành những khối nhớ bằng nhau, có kích thước 2n Bytes. Mỗi khối nhớ được gọi là gì?
   1. **Frame.**
   2. Page.
   3. Offset.
   4. Page table.
7. Hệ điều hành dùng kỹ thuật phân trang bộ nhớ sẽ giải quyết được vấn đề nào?
   1. **Tránh được phân mảnh ngoại.**
   2. Tăng kích thước bộ nhớ ảo.
   3. Tránh được phân mảnh nội.
   4. Giảm kích thước bộ nhớ cho tiến trình.
8. Kỹ thuật phân trang bộ nhớ có gây ra phân mảnh trong không?
   1. **Có gây ra phân mảnh nội.**
   2. Không gây ra phân mảnh nội.
   3. Phụ thuộc số frame của bộ nhớ vật lý.
   4. Phụ thuộc số page của bộ nhớ luận lý.



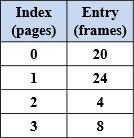
1. Trong kỹ thuật phân trang (paging), bộ nhớ vật lý được chia thành những khối nhớ bằng nhau, có kích thước 2n Bytes. Mỗi tiến trình được cấp số lượng X khối nhớ. Hệ điều hành sẽ đánh thứ tự cho X khối nhớ từ 0 đến X-1. Mỗi khối nhớ đó được Hệ điều hành gọi là gì?
   1. **Frame.**
   2. Page.
   3. Offset.
   4. Page table.
2. Trong kỹ thuật phân trang bộ nhớ (paging), kích thước Frame so với kích thước của Page là như thế nào?
   1. **kích thước Frame bằng kích thước Page.**
   2. kích thước Frame lớn hơn kích thước Page.
   3. kích thước Frame nhỏ hơn kích thước Page.
   4. kích thước Frame bằng hoặc lớn hơn kích thước Page.
3. Ảnh bên dưới mô tả 1 bảng trang (Page table) của tiến trình P chạy trong Hệ điều hành dùng

kỹ thuật phân trang. Hãy cho biết tiến trình P được cấp bao nhiêu Frame?



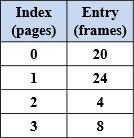
* 1. 20 frame.
  2. 24 frame.
  3. **4 frame.**
  4. 8 frame.

1. Ảnh bên dưới mô tả 1 bảng trang (Page table) của tiến trình P chạy trong Hệ điều hành dùng kỹ thuật phân trang. Nếu mỗi Frame = 8 Bytes, hãy cho biết tiến trình P được cấp bao nhiêu dung lượng bộ nhớ?



* 1. 160 bytes.
  2. 192 bytes.
  3. **32 bytes.**
  4. 64 bytes.

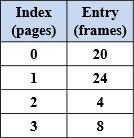
1. Ảnh bên dưới mô tả 1 bảng trang (Page table) của tiến trình P chạy trong Hệ điều hành dùng

kỹ thuật phân trang. Cho biết các số thuộc cột “Index (pages)” là gì?

* 1. độ dời của page.
  2. **chỉ số page (p) trong địa chỉ luận lý.**
  3. chỉ số page (p) trong địa chỉ vật lý.
  4. độ dời của frame.

1. Ảnh bên dưới mô tả 1 bảng trang (Page table) của tiến trình P chạy trong Hệ điều hành dùng

kỹ thuật phân trang. Cho biết các số thuộc cột “Entry (frames)” là gì?



* 1. độ dời của page.
  2. frame number trong địa chỉ luận lý.
  3. **frame number trong địa chỉ vật lý.**
  4. độ dời của frame.

1. Trong kỹ thuật phân trang bộ nhớ, địa chỉ Hệ điều hành cấp cho CPU truy cập tiến trình trong bộ nhớ (gọi là địa chỉ luận lý) có dạng nào?
   1. số page (p), số frame (f) và độ dời frame (t)
   2. số page (p), số frame (f) và độ dời frame (t).
   3. số page (p) và kích thước page (s).
   4. **số page (p) và độ dời page (d).**
2. Trong kỹ thuật phân trang bộ nhớ, tiến trình được cấp không gian bộ nhớ là 2m bytes, kích thước mỗi page là 2n byte. Địa chỉ luận lý của tiến trình có dạng (p,d). Hãy cho biết p gồm bao nhiêu bits?
   1. **số bit của p = *m-n* bits**
   2. số bit của p = *m* bits.
   3. số bit của p = *n* bits.
   4. số bit của p = 2m bits.
3. Trong kỹ thuật phân trang bộ nhớ, tiến trình được cấp không gian bộ nhớ là 2m bytes, kích thước mỗi page là 2n byte. Địa chỉ luận lý của tiến trình có dạng (p,d). Hãy cho biết d gồm bao nhiêu bits?
   1. số bit của d = *m-n* bits
   2. số bit của d = *m* bits.



* 1. **số bit của d = *n* bits.**
  2. số bit của d = 2m bits.

1. Trong kỹ thuật phân trang bộ nhớ, kích thước frame = 212 bytes. Tiến trình P dùng Page table có chỉ số page (p) = 32 bit. Hỏi tiến trình P được cấp không gian bộ nhớ là bao nhiêu bytes:
   1. 225 bytes.
   2. **244 bytes.**
   3. 215 bytes.
   4. 217 bytes.
2. Trong kỹ thuật phân trang bộ nhớ, một không gian địa chỉ luận lý có 24 pages, mỗi page có kích thước 211 bytes, ánh xạ vào bộ nhớ vật lý có 26 frames. Hỏi địa chỉ luận lý dùng bao nhiêu bit?
   1. 17 bit.
   2. 10 bit.
   3. **15 bit.**
   4. 21 bit.
3. Trong kỹ thuật phân trang bộ nhớ, một không gian địa chỉ luận lý có 24 pages, mỗi page có kích thước 211 bytes, ánh xạ vào bộ nhớ vật lý có 26 frames. Hỏi địa chỉ vật lý của bộ nhớ dùng bao nhiêu bit?
   1. **17 bit.**
   2. 10 bit.
   3. 15 bit.
   4. 21 bit.
4. Trong kỹ thuật phân trang bộ nhớ, một không gian địa chỉ luận lý có 24 pages, mỗi page có kích thước 211 bytes, ánh xạ vào bộ nhớ vật lý có 26 frames. Hỏi địa chỉ luận lý dùng bao nhiêu bits cho chỉ số page (p)?
   1. 4 bit.
   2. 10 bit.
   3. 11 bit.
   4. **6 bit.**
5. Lỗi trang (page fault) là gì?
   1. Thông tin trong page đó bị hư.
   2. **Thông tin trong page đó đã được swap out ra bộ nhớ phụ.**
   3. Thông tin trong page đó đã bị xóa bỏ.
   4. Tiến trình không được truy xuất page đó.
6. Trong kỹ thuật phân trang bộ nhớ, bit “invalid” gán vào số hiệu page trong Page table có ý

nghĩa page gì?

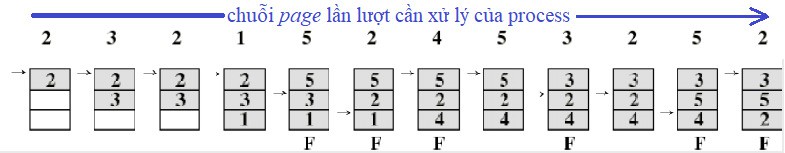
* 1. **page đó đã bị lỗi trang (page fault).**
  2. page đó không được phép truy xuất.
  3. page đó chứa dữ liệu không định dạng.
  4. page đó đang có tranh chấp.

1. Một hệ thống sử dụng kỹ thuật phân trang, với Page table được lưu trữ trong bộ nhớ chính. Thời gian cho một lần truy xuất bộ nhớ chính là 200 nanoseconds. Hỏi mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này, nếu không xảy ra lỗi trang (page fault)?
   1. 200 ns.
   2. **400 ns.**
   3. 600 ns.
   4. 800 ns.
2. Một hệ thống sử dụng kỹ thuật phân trang, với Page table được lưu trữ trong bộ nhớ chính. Thời gian cho một lần truy xuất bộ nhớ chính là 200 nanoseconds. Hỏi mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này, nếu xảy ra lỗi trang (page fault)?
   1. 200 ns.
   2. 400 ns.
   3. **600 ns.**
   4. 800 ns.
3. Một hệ thống sử dụng kỹ thuật phân trang, với Page table được lưu trữ trong bộ nhớ chính.

Thời gian cho một lần truy xuất bộ nhớ chính là 200 nanoseconds. Sử dụng bộ đệm TLB (*translation look-aside buffer)* với hit- ratio (tỉ lệ tìm thấy) là 75%, thời gian để tìm trong TLB xem như bằng 0. Hỏi thời gian truy xuất bộ nhớ trong hệ thống (effective memory reference time).

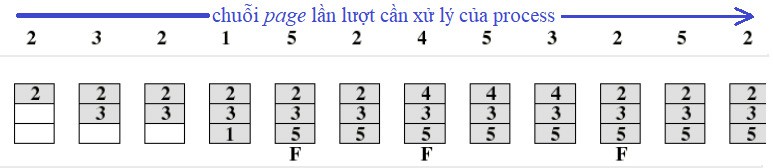
* 1. **250 ns.**
  2. 300 ns.
  3. 450 ns.
  4. 200 ns.

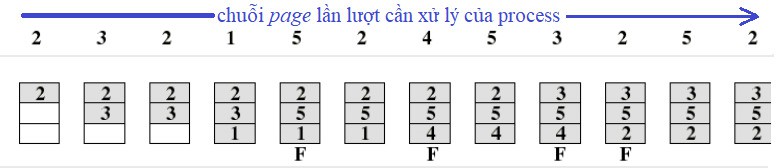
1. Trong kỹ thuật phân trang bộ nhớ, sử dụng thanh ghi cho Page table trong trường hợp:
   1. số page bằng số frame.
   2. **Page table có kích thước nhỏ.**
   3. kích thước page nhỏ.
   4. Page table có nhiều số hiệu page.
2. Trong kỹ thuật phân trang bộ nhớ, cấu trúc TLB (Translation look-aside buffers) thường được dùng cho trường hợp:
   1. sử dụng thanh ghi cho Page table.
   2. Page table có kích thước nhỏ.
   3. **Page table lưu trong bộ nhớ chính.**
   4. Page table có ít số hiệu page.
3. Ảnh dưới cho thấy chuỗi page cần xử lý của 1 tiến trình (tiến trình chỉ được cấp 3 pages), kèm theo đó là minh họa cho 1 giải thuật thay thế trang (page replacement). Những vị trí có ký hiệu F là thời điểm cần thay thế trang. Hãy cho biết tên của giải thuật thay trang.





* 1. **Giải thuật FIFO (First In – First Out)**
  2. Giải thuật OTP (Optimal)
  3. Giải thuật LRU (Least Recently Used).
  4. Giải thuật SJF (Shortest Job First)

1. Ảnh dưới cho thấy chuỗi page cần xử lý của 1 tiến trình (tiến trình chỉ được cấp 3 pages), kèm theo đó là minh họa cho 1 giải thuật thay thế trang (page replacement). Những vị trí có ký hiệu F là thời điểm cần thay thế trang. Hãy cho biết tên của giải thuật thay trang.
   1. Giải thuật FIFO (First In – First Out)
   2. **Giải thuật OTP (Optimal)**
   3. Giải thuật LRU (Least Recently Used).
   4. Giải thuật SJF (Shortest Job First)
2. Ảnh dưới cho thấy chuỗi page cần xử lý của 1 tiến trình (tiến trình chỉ được cấp 3 pages), kèm theo đó là minh họa cho 1 giải thuật thay thế trang (page replacement). Những vị trí có ký hiệu F là thời điểm cần thay thế trang. Hãy cho biết tên của giải thuật thay trang.



* 1. Giải thuật FIFO (First In – First Out)
  2. Giải thuật OTP (Optimal)
  3. **Giải thuật LRU (Least Recently Used).**
  4. Giải thuật SJF (Shortest Job First)

1. Giải thuật thay thế trang (page replacement) nào sau đây xác suất gây lỗi trang (page fault)

là thấp nhất:

* 1. Giải thuật FIFO (First In – First Out)
  2. Giải thuật OTP (Optimal)
  3. **Giải thuật LRU (Least Recently Used).**
  4. Giải thuật SJF (Shortest Job First).

***CHƯƠNG 5***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1A** | **2B** | **3C** | **4A** | **5B** | **6B** | **7C** | **8D** | **9A** | **10D** | **11B** | **12D** | **13C** |
| **14B** | **15B** | **16C** | **17D** | **18A** | **19C** | **20A** | **21D** | **22C** | **23A** | **24B** | **25C** | **26A** |
| **27D** | **28C** | **29B** | **30A** | **31A** | **32A** | **33B** | **34A** | **35C** | **36C** | **37B** | **38C** | **39D** |
| **40A** | **41C** | **42B** | **43C** | **44A** | **45D** | **46B** | **47A** | **48B** | **49C** | **50A** | **51B** | **52C** |
| **53A** | **54B** | **55C** | **56C** |  |  |  |  |  |  |  |  |  |

* 1. Seek time, Rotational delay, Read time.



1. Bảng phân khu (partition table) được lưu trữ ở đâu?
   1. Sector 0 của phân khu khởi động.
   2. Khối khởi động của đĩa hệ thống.
   3. Sector 0 của đĩa vật lý.
   4. Sector 0 của phân khu đầu tiên.
2. Cho 5 hạng mục khởi động máy tính như bên dưới.
3. Tải MBR để đọc thông tin bản phân vùng
4. tìm vùng Active
5. POST (Power On Self Test)
6. Nạp và thực thi Kernel và các dịch vụ liên quan của Hệ điều hành.
7. Nạp và thực thi Boot loader của Hệ điều hành.

Sắp xếp theo trình tự ĐÚNG cho các giai đoạn khởi động của máy tính.

* 1. 1, 3, 2, 5, 4
  2. 3, 1, 5, 4, 2
  3. 3, 1, 2, 5, 4
  4. 3, 4, 5, 2, 1

1. Công việc nào được thực hiện đầu tiên trong quá trình khởi động của hệ thống máy tính?
   1. Thực thi chương trình Power On Self Test (POST)
   2. Kiểm tra BIOS
   3. Kiểm tra RAM
   4. Tìm và nạp Hệ điều hành.
2. Thông tin của Master Boot Record (MBR) lưu ở đâu trong ổ đĩa?
   1. Sector 0 của phân khu khởi động.
   2. Khối khởi động của đĩa hệ thống.
   3. Sector 0 của phân khu đầu tiên.
   4. Sector 0 của đĩa vật lý.
3. Chương trình bootstrap (dùng nạp boot loader) được lưu giữ ở đâu?
   1. Bộ nhớ RAM.
   2. Sector thứ 1 của đĩa cứng vật lý.
   3. Bộ nhớ ROM.
   4. Bộ nhớ đệm.
4. Nhiệm vụ của chương trình khởi động mồi (bootstrap):
   1. Khởi động các phần cứng của máy tính.
   2. Nạp nhân hệ điều hành vào bộ nhớ.
   3. Nhảy đến địa chỉ ban đầu để thực thi hệ điều hành.
   4. Nạp và thực thi Boot loader của Hệ điều hành.
5. Bộ nạp khởi động Hệ điều hành (Boot loader) lưu ở đâu trong đĩa cứng?
   1. Thư mục gốc của phân khu khởi động (Active partition).
   2. Khối khởi động của đĩa hệ thống.
   3. Sector 0 của phân khu đầu tiên.
   4. Sector 0 của đĩa vật lý.
6. Cấu trúc luận lý của một đĩa cứng gồm các thành phần nào?
   1. MBR, NTFS, Partition
   2. MBR, Partition table, Extended
   3. MBR, Partition table, FAT32
   4. MBR, Partition table, Partitions
7. Các giải thuật định thời truy cập đĩa bao gồm những giải thuật nào?
   1. FCFS, SSTF, SCAN, C-SCAN (Circular SCAN), Round-robin.
   2. FCFS, SSTF, SCAN, First-fit, Best-fit, Worst-fit.
   3. FCFS, SSTF, C-SCAN (Circular SCAN), C-LOOK.
   4. FCFS, SSTF, SCAN, C-SCAN (Circular SCAN), C-LOOK.
8. Các giải thuật định thời cho đĩa cứng:
   1. FIFO, SJF, Round Robin.
   2. FIFO, OPT, LRU.
   3. FCFS, SSTF, SCAN.
   4. LOOK, FIFO.
9. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi thứ tự phục vụ của thuật toán ***First Come First Serve* (FCFS)**
   1. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 122, 124, 183
   2. thứ tự phục vụ 53, 98, 183, 37, 122, 14, 124, 65, 67
   3. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 183, 124, 122
   4. thứ tự phục vụ 53, 98, 183, 37, 67, 14, 124, 65, 122
10. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi thứ tự phục vụ của thuật toán ***Shortest-Seek-Time First (SSTF)***
    1. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 122, 124, 183
    2. thứ tự phục vụ 53, 98, 183, 37, 122, 14, 124, 65, 67
    3. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 183, 124, 122
    4. thứ tự phục vụ 53, 98, 183, 37, 67, 14, 124, 65, 122
11. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi thứ tự phục vụ của thuật toán ***SCAN***
    1. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 122, 124, 183
    2. thứ tự phục vụ 53, 37, 14, 65, 67, 98, 122, 124, 183
    3. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 183, 124, 122
    4. thứ tự phục vụ 53, 98, 183, 37, 122, 14, 124, 65, 67



1. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi thứ tự phục vụ của thuật toán C-***SCAN***
   1. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 122, 124, 183
   2. thứ tự phục vụ 53, 98, 183, 37, 122, 14, 124, 65, 67
   3. thứ tự phục vụ 53, 65, 67, 98, 122, 124, 183, 199, 0, 14, 37
   4. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 183, 124, 122
2. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi thứ tự phục vụ của thuật toán **C-LOOK.**
   1. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 122, 124, 183
   2. thứ tự phục vụ 53, 98, 183, 37, 122, 14, 124, 65, 67
   3. thứ tự phục vụ 53, 65, 67, 37, 14, 98, 183, 124, 122
   4. thứ tự phục vụ 53, 65, 67, 98, 122, 124, 183, 14, 37
3. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi tổng số bước di chuyển đầu đọc của thuật toán ***First Come First Serve* (FCFS)**
   1. 236 cylinder
   2. 640 cylinder
   3. 226 cylinder
   4. 650 cylinder
4. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi tổng số bước di chuyển đầu đọc của thuật toán ***Shortest-Seek-Time First (SSTF)***
   1. 236 cylinder
   2. 640 cylinder
   3. 226 cylinder
   4. 650 cylinder
5. Đĩa cứng có 200 cylinder (từ 0 đến 199). Đầu từ đang ở cylinder số 53. Hàng đợi đang có các yêu cầu truy xuất đĩa cứng tại các cylinder 98, 183, 37, 122, 14, 124, 65, 67. Hỏi tổng số bước di chuyển đầu đọc của thuật toán ***SCAN***
   1. 236 cylinder
   2. 640 cylinder
   3. 226 cylinder
   4. 208 cylinder
6. Hiện tượng nào có thể xảy ra khi áp dụng giải thuật định thời đĩa “Thời gian tìm kiếm ngắn nhất” (SSTF):
   1. Đầu đọc bị xoay vòng vô tận.
   2. Hàng đợi phục vụ bị gián đoạn.
   3. Yêu cầu không bao giờ được phục vụ.
   4. Hàng đợi phục vụ liên tục.
7. Trong giải thuật định thời đĩa SCAN, khi đầu đọc đạt tới đầu kia của đĩa:
   1. Quay về và phục vụ I/O trên đường về.
   2. Quay về điểm xuất phát ngay lập tức.
   3. Đợi chì thị của bộ định thời.
   4. Đợi chì thị của CPU.
8. Trong giải thuật định thời đĩa C-SCAN, khi đầu đọc đạt tới đầu kia của đĩa sẽ:
   1. quay về và phục vụ I/O trên đường về.
   2. quay về điểm xuất phát ngay lập tức.
   3. đợi chì thị của bộ định thời.
   4. đợi chì thị của CPU.
9. Trong giải thuật định thời đĩa LOOK, đầu đọc quay về khi:
   1. đạt tới đầu kia của đĩa.
   2. khi nhận được chỉ thị của bộ định thời.
   3. tới vị trí của yêu cầu I/O cuối cùng trên hướng đi.
   4. khi nhận được chỉ thị của CPU.
10. Trong giải thuật định thời đĩa C-LOOK, khi chuyển hướng quay về, đầu đọc sẽ dịch chuyển:
    1. về điểm đầu của đĩa.
    2. về vị trí của yêu cầu I/O đầu tiên.
    3. khi nhận được chỉ thị của CPU.
    4. khi nhận được chỉ thị của bộ định thời.
11. Các chuẩn đĩa cứng nào hỗ trợ tạo hệ thống RAID?
    1. ATA, PATA, IDE.
    2. PATA và SATA.
    3. SATA và SCSI.
    4. IDE và SCSI.
12. Kỹ thuật phối hợp dãy các đĩa cứng RAID-0 có tên gọi là gì?
    1. striping (tạm dịch: tước / tách dữ liệu).
    2. mirroring (tạm dịch: ảnh dữ liệu qua gương).
    3. parity (tạm dịch: tương trợ dữ liệu).
    4. clustering (tạm dịch: chuỗi kết nối dữ liệu).
13. Kỹ thuật phối hợp dãy các đĩa cứng RAID-1 có tên gọi là gì?
    1. striping (tạm dịch: tước / tách dữ liệu).
    2. mirroring (tạm dịch: ảnh dữ liệu qua gương).
    3. parity (tạm dịch: tương trợ dữ liệu).
    4. clustering (tạm dịch: chuỗi kết nối dữ liệu).



1. Kỹ thuật phối hợp dãy các đĩa cứng RAID-5 có tên gọi là gì?
   1. striping (tạm dịch: tước / tách dữ liệu).
   2. mirroring (tạm dịch: ảnh dữ liệu qua gương).
   3. parity (tạm dịch: tương trợ dữ liệu).
   4. clustering (tạm dịch: chuỗi kết nối dữ liệu).
2. Kỹ thuật phối hợp dãy các đĩa cứng RAID-10 cần tối thiểu bao nhiêu ổ đĩa vật lý?
   1. 2 ổ đĩa vật lý.
   2. 3 ổ đĩa vật lý.
   3. 4 ổ đĩa vật lý.
   4. 6 ổ đĩa vật lý.
3. Kỹ thuật phối hợp dãy các đĩa cứng RAID-5 cần tối thiểu bao nhiêu ổ đĩa vật lý?
   1. 2 ổ đĩa vật lý.
   2. 3 ổ đĩa vật lý.
   3. 4 ổ đĩa vật lý.
   4. 5 ổ đĩa vật lý.
4. Kỹ thuật phối hợp dãy các đĩa cứng RAID-1 cần tối thiểu bao nhiêu ổ đĩa vật lý?
   1. 2 ổ đĩa vật lý.
   2. 3 ổ đĩa vật lý.
   3. 4 ổ đĩa vật lý.
   4. 5 ổ đĩa vật lý.
5. Ưu điểm của kỹ thuật phối hợp dãy các đĩa cứng RAID-0 là:
   1. tốc độ cao.
   2. độ tin cậy cao.
   3. tốc độ và độ tin cậy cao.
   4. chỉ cần 1 ổ đĩa.
6. Ưu điểm của kỹ thuật phối hợp dãy các đĩa cứng RAID-1 là:
   1. tốc độ cao.
   2. độ tin cậy cao.
   3. tốc độ và độ tin cậy cao.
   4. chỉ cần 1 ổ đĩa.
7. Ưu điểm của kỹ thuật phối hợp dãy các đĩa cứng RAID-10 là:
   1. tốc độ cao.
   2. độ tin cậy cao.
   3. tốc độ và độ tin cậy cao.
   4. chỉ cần 1 ổ đĩa.
8. Nếu dùng kỹ thuật phối hợp dãy các đĩa cứng RAID-0 cho 2 ổ đĩa, mỗi ổ có dung lượng 100GB thì tổng dung lượng có thể lưu trữ trên ổ đĩa luận lý tạo bởi RAID là bao nhiêu?
   1. 50 GB
   2. 100 GB
   3. 150 GB
   4. 200 GB.
9. Nếu dùng kỹ thuật phối hợp dãy các đĩa cứng RAID-1 cho 2 ổ đĩa, mỗi ổ có dung lượng 100GB thì tổng dung lượng có thể lưu trữ trên ổ đĩa luận lý tạo bởi RAID là bao nhiêu?
   1. 50 GB
   2. 100 GB
   3. 150 GB
   4. 200 GB.
10. Nếu dùng kỹ thuật phối hợp dãy các đĩa cứng RAID-5 cho 3 ổ đĩa, mỗi ổ có dung lượng 100GB thì tổng dung lượng có thể lưu trữ trên ổ đĩa luận lý tạo bởi RAID là bao nhiêu?
    1. 100 GB
    2. 150 GB
    3. 200 GB
    4. 300 GB.

# CHƯƠNG 7: HỆ THỐNG TẬP TIN (FILE SYSTEM)



1. Để đảm bảo tính bền vững cho thông tin lưu trữ, thiết bị lưu trữ (gọi là đĩa) sẽ sử dụng loại vật liệu nào dưới đây?
   1. Đĩa từ tính
   2. Đĩa quang
   3. Chip bán dẫn bằng chất dẻo.
   4. Tất cả các loại trên.
2. Đặc tính nào dưới đây KHÔNG đáp ứng được yêu cầu tăng tốc độ truy xuất thông tin trên thiết bị lưu trữ?
   1. Tăng tốc độ xoay đĩa.
   2. Tăng tốc truyền dẫn.
   3. Tăng tốc độ xử lý của CPU.
   4. Giảm thời gian tìm kiếm / đọc / ghi.
3. Hạng mục nào dưới đây KHÔNG phải là yêu cầu về lưu trữ thông tin đối với Hệ điều

hành?

* 1. kích thước (size)
  2. tính bền vững (persistence).
  3. dễ sử dụng (ease of use).
  4. cho phép chia sẻ và bảo vệ (Sharing/Protection).

1. Khái niệm tập tin (file) nào dưới đây là đúng?
   1. File là chuỗi các bytes thông tin được lưu trữ trên bộ nhớ RAM.
   2. File là chuỗi các bytes thông tin được lưu trữ trên CPU
   3. File là chuỗi các bytes thông tin được lưu trữ trên thiết bị đĩa.
   4. File là chuỗi các bytes thông tin được lưu trữ trên thanh ghi.
2. Tập tin trên ổ đĩa được lưu trữ theo cấu trúc nào?
   1. cấu trúc dạng cây (tree) và không có cấu trúc.
   2. cấu trúc đơn cấp và không có cấu trúc.
   3. có cấu trúc và không có cấu trúc.
   4. cấu trúc đa cấp và không có cấu trúc.
3. Tập tin không cấu trúc là loại nào?
   1. tập tin là một dãy tuần tự các byte
   2. tập tin là một dãy các mẫu tin có kích thước cố định
   3. tập tin gồm một cây của những mẫu tin không cần thiết có cùng chiều dài
   4. tập tin có một trường khóa giúp việc tìm kiếm nhanh hơn
4. Tập tin có cấu trúc là loại nào?
   1. tập tin là một dãy tuần tự các byte
   2. tập tin là một dãy các mẫu tin có kích thước cố định
   3. tập tin gồm một cây của những mẫu tin không cần thiết có cùng chiều dài
   4. tập tin có một trường khóa giúp việc tìm kiếm nhanh hơn
5. Đặc điểm của tập tin chia sẻ:
   1. Mỗi người dùng có một bản sao tập tin.
   2. Chỉ có một tập tin thực sự tồn tại.
   3. Mỗi người dùng lưu sửa đổi tập tin riêng.
   4. Tất cả người dùng có thể đồng thời sửa tập tin.
6. Truy cập tập tin KHÔNG dùng phương pháp nào?
   1. Truy cập tuần tự (sequential access).
   2. Truy cập ngẫu nhiên (random access).
   3. Truy cập dùng khóa (Key).
   4. Truy cập dùng ma trận (matrix).
7. Những tác vụ nào của tiến trình tác động trực tiếp đến tập tin?
   1. load, execute, create process, terminate process.
   2. request device, release device, read from device, write to device.
   3. create / delete, open / close, read, write.
   4. create messages, delete messages, send messages, receive messages.
8. Cấu trúc thư mục có các dạng nào?
   1. tương đối và tuyệt đối.
   2. chu trình và không chu trình.
   3. đơn cấp, hai cấp, cây, đồ thị.
   4. liên kết và không liên kết.
9. Dạng cấu trúc thư mục nào của Hệ điều hành không cho phép trùng tên file lưu trữ bên trong một hệ thống tập tin?
   1. đơn cấp (Single-level directory).
   2. hai cấp (Two-level directory).
   3. dạng cây (Tree-structured directory).
   4. ba cấp (Three-level directory).
10. Dạng cấu trúc thư mục nào của Hệ điều hành phân chia mỗi người dùng có danh sách tập tin riêng biệt?
    1. đơn cấp (Single-level directory).
    2. hai cấp (Two-level directory).
    3. dạng cây (Tree-structured directory).
    4. ba cấp (Three-level directory).
11. Dạng cấu trúc thư mục nào của Hệ điều hành cho phép người dùng có thể truy xuất vào tập tin của người dùng khác trong cùng hệ thống tập tin?
    1. đơn cấp (Single-level directory).
    2. hai cấp (Two-level directory).
    3. dạng cây (Tree-structured directory).



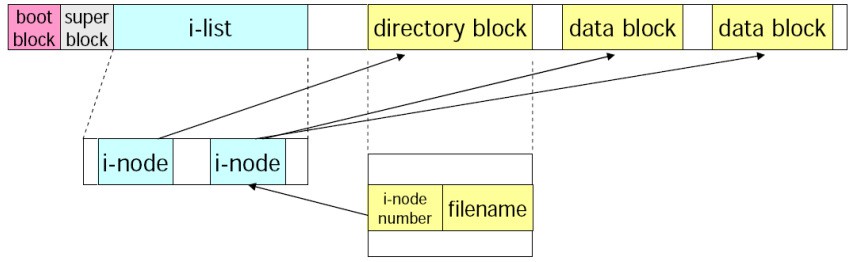
* 1. ba cấp (Three-level directory).

1. Dạng cấu trúc thư mục nào của Hệ điều hành cho phép người dùng có thể phân nhóm tập tin theo ý muốn?
   1. đơn cấp (Single-level directory).
   2. hai cấp (Two-level directory).
   3. dạng cây (Tree-structured directory).
   4. ba cấp (Three-level directory).
2. Dạng cấu trúc thư mục nào của Hệ điều hành cho phép một file (hay thư mục con) chia sẻ cho nhiều thư mục cha:
   1. đơn cấp (Single-level directory).
   2. hai cấp (Two-level directory).
   3. dạng cây (Tree-structured directory).
   4. dạng đồ thị không chứa chu trình (Acyclic-graph directory).
3. Khi duyệt thư mục trong cấu trúc dạng đồ thị không chứa chu trình (Acyclic-graph directory), hệ điều hành sẽ:
   1. bỏ qua các liên kết.
   2. duyệt cả các liên kết.
   3. duyệt riêng các liên kết.
   4. đếm số liên kết tới tập tin.
4. Trong hệ thống tập tin, tên đường dẫn tuyệt đối tới tập tin xuất phát từ:
   1. thư mục gốc của người dùng.
   2. thư mục hiện thời.
   3. thư mục gốc.
   4. thư mục cha.
5. Trong hệ thống tập tin, tên đường dẫn tương đối tới tập tin xuất phát từ:
   1. thư mục người dùng.
   2. thư mục hiện thời.
   3. thư mục gốc.
   4. thư mục cha.
6. Hệ thống quản lý tập tin (file system) được quản lý bởi:
   1. người dùng
   2. hệ điều hành
   3. tiến trình
   4. luồng (thread)
7. Volume (tạm dịch: bộ lưu) là gì?
   1. một thực thể chứa hệ thống tập tin.
   2. một phân khu trên đĩa cứng.
   3. một phân khu tráo đổi.
   4. một phân vùng nhớ.
8. Hệ điều hành KHÔNG THỂ tổ chức 1 hệ thống quản lý tập tin (file system) trên đối tượng lưu trữ nào?
   1. một phân khu (partition).
   2. nhiều phân khu (partitions).
   3. một ổ đĩa (disk)
   4. không gian tráo đổi (swap partition).
9. Các hệ thống quản lý tập tin thường nhóm các cung từ (sector) thành các liên cung (cluster).

Mục đích của việc này là gì?

* 1. tăng số dòng chỉ mục (index) cho bảng cấp phát tập tin.
  2. nội dung của 1 tập tin nằm trên các sector gần kề => tăng hiệu suất truy xuất.
  3. tiết kiệm không gian lưu trữ.
  4. Tất cả đúng.

1. Điều nào sau đây là KHÔNG ĐÚNG đối với việc tổ chức không gian lưu trữ trên ổ đỉa (Layout) của Hệ điều hành?
   1. Chia không gian ổ đĩa thành những block (hay Cluster) bằng nhau.
   2. Xây dựng bảng “Partition Control Block” ghi địa chỉ block, used blocked…
   3. Xây dựng bảng “File Control Block” ghi lưu thông tin các file, địa chỉ lưu trữ file…
   4. Định thời cho CPU truy xuất các file trong đĩa.
2. Quan sát “layout tổ chức không gian lưu trữ tập tin trên đĩa” bên dưới. Cho biết đây là layout của Hệ điều hành nào?



* 1. MS-DOS File system.
  2. Windows XP File system.
  3. Unix File system.
  4. Netware File system.

1. Quan sát “layout tổ chức không gian lưu trữ tập tin trên đĩa” bên dưới. Cho biết đây là layout của Hệ điều hành nào?



* 1. MS-DOS File system.
  2. Windows XP File system.
  3. Unix File system.



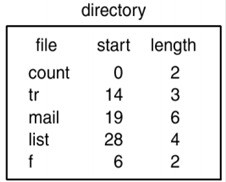
* 1. Netware File system.

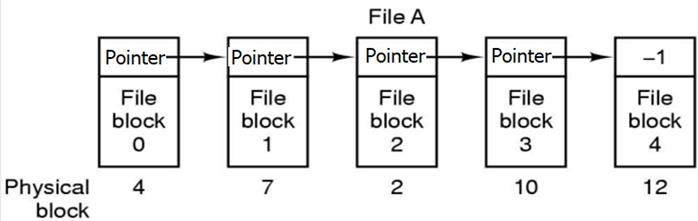
1. Tổ chức của MS– DOS
   1. Boot block, FAT, Master boot, data blocks
   2. Boot block, FAT32, root directory, data blocks
   3. Boot block, FAT, root directory, data blocks
   4. Boot block, NTFS, root directory, Master Boot
2. Cấu trúc thư mục của hệ điều hành MS-DOS là

A: Cấu trúc thư mục dạng đơn cấp B: Cấu trúc thư mục dạng hai cấp

C. Cấu trúc thư mục dạng cây

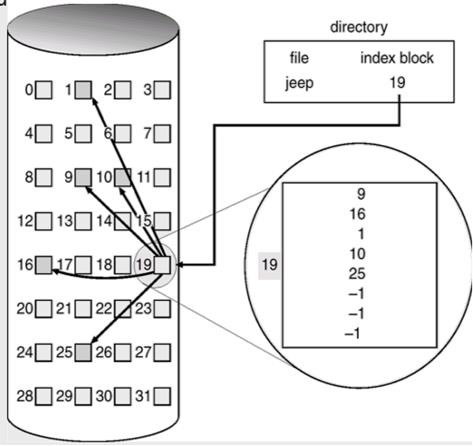
D. Cấu trúc thư mục dạng ba cấp

1. Hệ điều hành thực hiện việc cấp phát không gian lưu trữ trên đĩa cho đối tương nào?
   1. Thư mục (Folder)
   2. Tập tin (File)
   3. Bộ nhớ (Memory)
   4. Tiến trình (Process)
2. Phát biểu nào KHÔNG ĐÚNG về VFS (Virtual File System)?
   1. VFS là hệ thống quản lý tập tin, được lưu trữ trên mỗi ổ đĩa.
   2. là một thư viện dùng chung do Hệ điều hành tạo ra
   3. hỗ trợ ứng dụng truy cập Files trên các ổ đĩa dùng File System khác nhau
   4. ứng dụng chỉ cần gởi yêu cầu truy xuất file cho VFS.
3. Phương pháp “cấp phất không gian lưu trữ trên đĩa” nào dưới đây là không có?
   1. Cấp phát liên tục (continuous allocation)
   2. Cấp phát theo danh sách liên kết (linked list allocation)
   3. Cấp phát theo nhu cầu (required allocation).
   4. Cấp phát dùng chỉ mục (indexed allocation)
4. Quan sát bảng thư mục dưới đây và cho biết thuộc loại “cấp phất không gian lưu trữ trên đĩa” nào?
   1. Cấp phát liên tục (continuous allocation)
   2. Cấp phát theo danh sách liên kết (linked list allocation)
   3. Cấp phát dùng chỉ mục (indexed allocation).
   4. Cấp phát theo nhu cầu (required allocation).
5. Quan sát cấu trúc cấp phát block cho “File A” ở hình dưới. Cho biết đây là loại “cấp phất không gian lưu trữ trên đĩa” nào?



* 1. Cấp phát liên tục (continuous allocation)
  2. Cấp phát theo danh sách liên kết (linked list allocation)
  3. Cấp phát dùng chỉ mục (indexed allocation).
  4. Cấp phát theo nhu cầu (required allocation).

1. Quan sát cấu trúc cấp phát block ở hình dưới. Cho biết đây là loại “cấp phất không gian lưu trữ trên đĩa” nào?



* 1. Cấp phát liên tục (continuous allocation)
  2. Cấp phát theo danh sách liên kết (linked list allocation)
  3. Cấp phát dùng chỉ mục (indexed allocation).
  4. Cấp phát theo nhu cầu (required allocation).



1. Hệ thống quản lý tập tin FAT của Hệ điều hành MS-DOS dùng phương pháp “cấp phất không gian lưu trữ trên đĩa” nào?
   1. Cấp phát liên tục (continuous allocation)
   2. Cấp phát theo danh sách liên kết (linked list allocation)
   3. Cấp phát dùng chỉ mục (indexed allocation).
   4. Cấp phát theo nhu cầu (required allocation).
2. Hệ thống quản lý tập tin Ext2 của Hệ điều hành Linux dùng phương pháp “cấp phất không gian lưu trữ trên đĩa” nào?
   1. Cấp phát liên tục (continuous allocation)
   2. Cấp phát theo danh sách liên kết (linked list allocation)
   3. Cấp phát dùng chỉ mục (indexed allocation).
   4. Cấp phát theo nhu cầu (required allocation).
3. Phương pháp cấp phát không gian đĩa liên tục (continuous allocation) có đặc điểm:
   1. Dễ thao tác và cài đặt.
   2. Tập tin không thể phát triển kích thước.
   3. Gây phân mảnh đĩa.
   4. dễ gây phân mảnh, dễ thao tác cài đặt, không thay đổi kích thước.
4. Phương pháp cấp phát không gian đĩa liên tục (continuous allocation), thời gian tìm kiếm (seek time) cho 1 tập tin lưu trữ trên *n* blocks sẽ bằng bao nhiêu?
   1. bằng *n* lần seek time.
   2. bằng thời gian cho 1 lần seek time.
   3. bằng *n/2* lần seek time.
   4. bằng *2n* lần seek time.
5. Phương pháp cấp phát không gian đĩa liên tục (continuous allocation), dòng thông tin file trong bảng thư mục (directory table) cần phải có các thông tin?
   1. số hiệu khối bắt đầu và số khối đã cấp cho file.
   2. số hiệu khối đầu và cuối của file.
   3. số hiệu cung từ (sector) bắt đầu của tập tin.
   4. số cung từ (sector) đã cấp cho tập tin.
6. Trong bảng thư mục (directory table) của phương pháp cấp phát đĩa kiểu danh sách liên kết (linked allocation), mỗi dòng thông tin file phải có hạng mục nào?:
   1. số hiệu khối bắt đầu và số khối đã cấp cho file.
   2. số hiệu khối đầu và khối cuối của file.
   3. số hiệu cung từ (sector) bắt đầu của tập tin.
   4. số cung từ (sector) đã cấp cho tập tin.
7. Các phương pháp thường dùng để quản lý không gian trống:
   1. Danh sách liên kết (Linked list) và bit vector.
   2. Bảng chỉ mục.
   3. Ngăn xếp.
   4. Bảng cấp phát tập tin.
8. Phương pháp quản lý không gian trống bằng “vector bit”, mỗi khối trống (free) hoặc đã

dùng (used) trên đĩa sẽ được biểu diễn bằng:

* 1. một byte (Yes hoặc No).
  2. một con trỏ trong bảng các khối trống.
  3. một bit (0 hoặc 1).
  4. một chỉ mục trong bảng các khối trống.

1. Trong phương pháp quản lý không gian trống bằng “Linked list” (danh sách liên kết), mỗi khối trống (free block) sẽ có:
   1. một con trỏ chỉ đến khối trống kế.
   2. một thông tin báo hiệu là khối trống.
   3. một ngăn xếp các khối trống.
   4. một bảng chỉ mục các khối trống.

# CHƯƠNG 8: QUẢN LÝ NHẬP XUẤT (I/O MANAGEMENT)



1. Thiết bị nào sau đây KHÔNG được quản lý bởi thiết bị nhập xuất (Input / Output): A: Card mạng

B: Ổ đĩa

C. RAM

D. ROM

1. Nhóm thiết bị nào dưới đây không thuộc thiết bị nhập xuất (Input / Output)?
   1. Keyboard, mouse, Webcam.
   2. Monitor, Projector, Printer.
   3. VGA card, Sound card, Network card.
   4. RAM, ROM, CPU.
2. Thiết bị nào mang vai trò vừa nhập vừa xuất dữ liệu?
   1. máy in
   2. máy scan
   3. card mạng
   4. camera
3. Phân nhóm thiết bị theo hình thức giao tiếp, bàn phím (Keyboard) thuộc nhóm thiết bị dùng kiểu giao tiếp nào?
   1. Lệnh và dữ liệu truyền đi theo từng khối (block).
   2. Lệnh và dữ liệu truyền đi theo từng dòng ký tự (character).
   3. Lệnh và dữ liệu được đóng gói theo một Port (TCP/UDP)
   4. Lệnh và dữ liệu truyền theo kênh (channel)
4. Phân nhóm thiết bị theo hình thức giao tiếp, ổ đĩa cứng (Hard Disk Drive) thuộc nhóm thiết bị dùng kiểu giao tiếp nào?
   1. Lệnh và dữ liệu truyền đi theo từng khối (block).
   2. Lệnh và dữ liệu truyền đi theo từng dòng ký tự (character).
   3. Lệnh và dữ liệu được đóng gói theo một Port (TCP/UDP)
   4. Lệnh và dữ liệu truyền theo kênh (channel)
5. Phân nhóm thiết bị theo hình thức giao tiếp, card mạng (Network Controller) thuộc nhóm thiết bị dùng kiểu giao tiếp nào?
   1. Lệnh và dữ liệu truyền đi theo từng khối (block).
   2. Lệnh và dữ liệu truyền đi theo từng dòng ký tự (character).
   3. Lệnh và dữ liệu được đóng gói theo một Port (TCP/UDP)
   4. Lệnh và dữ liệu truyền theo kênh (channel)
6. Loại cổng (port) kết nối nào sau đây có tốc độ cao nhất
   1. COM
   2. USB 2.0
   3. LPT
   4. Gigabit Ethernet
7. Loại Bus nào được sử dụng giao tiếp với thiết bị nhập xuất (Input / Output)?
   1. PCI bus, PCI express bus.
   2. Memory bus.
   3. Clock bus.
   4. Address bus.
8. Loại Controller nào KHÔNG được sử dụng giao tiếp với thiết bị nhập xuất (Input /

Output)?

* 1. Network Controller.
  2. Memory Controller.
  3. VGA Controller.
  4. IDE Controller.

1. Hệ điều hành giao tiếp luận lý với thiệt bị ngoại vi, dùng phương pháp I/O port (hay I/O

instructions). Phát biểu nào là KHÔNG ĐÚNG?

* 1. Dùng I/O port để đọc / ghi thanh ghi trạng thái của I/O controller.
  2. Dùng I/O port để chuyển lệnh vào thanh ghi lệnh của I/O controller.
  3. Dùng I/O port để chuyển dữ liệu vào thanh ghi dữ liệu của I/O controller
  4. Dùng I/O port để điều khiển trực tiếp các phần cứng của thiết bị I/O

1. Hệ điều hành giao tiếp với thiệt bị ngoại vi bằng phương pháp “I/O instructions”. Địa chỉ trỏ đến các thanh ghi trên thiết bị I/O là:
   1. nằm ở phía đầu của vùng địa chỉ bộ nhớ chính.
   2. nằm ngoài vùng địa chỉ bộ nhớ chính.
   3. nằm ở phía cuối của vùng địa chỉ bộ nhớ chính.
   4. nằm ở vùng địa chỉ bộ nhớ đệm.
2. Hệ điều hành giao tiếp luận lý với thiệt bị ngoại vi, dùng phương pháp “memory-mapped I/O”. địa chỉ trỏ đến các thanh ghi trên thiết bị I/O là:
   1. nằm ở vùng địa chỉ bộ nhớ đệm.
   2. nằm ngoài vùng địa chỉ bộ nhớ.
   3. ánh xạ địa chỉ các thanh ghi I/O vào không gian địa chỉ bộ nhớ chính.
   4. ánh xạ vào bộ nhớ cache.
3. Để xuất nội dung đồ họa ra màn hình, Hệ điều hành sử dụng kỹ thuật giao tiếp luận lý với thiệt bị ngoại vi nào sau đây:
   1. Ánh xạ bộ nhớ (memory-mapped I/O).
   2. Chỉ thị nhập xuất (I/O instructions)
   3. Nhập xuất qua Port (I/O port).
   4. Truyền dữ liệu đến CPU.
4. Kỹ thuật nào sau đây KHÔNG PHẢI là “Kỹ thuật điều khiển truy cập thiết bị I/O”?
5. Kỹ thuật Polling (thăm dò)
6. Truy cập bộ nhớ gián tiếp



1. Kỹ thuật interrupt (ngắt)
2. Truy cập bộ nhớ trực tiếp DMA
3. Kỹ thuật Polling (thăm dò) cần kiểm tra trạng thái Busy (bận) của thiết bị I/O trước khi giao tiếp, Hệ điều hành tiến hành thao tác gì?
   1. Định kỳ đọc thanh ghi điều khiển của thiết bị I/O.
   2. Thăm dò lặp vòng thanh ghi trạng thái của thiết bị I/O.
   3. Dò tìm trong bộ nhớ chính.
   4. dò tìm trong CPU.
4. CPU lắng nghe yêu cầu ngắt (IRQ) từ thiết bị I/O theo chu kỳ nào?
   1. Sau khi thực thi mỗi chỉ thị.
   2. Sau khi thực thi 5 chỉ thị.
   3. Sau mỗi giây.
   4. Khi CPU nhàn rỗi.
5. Cấu trúc của Vector ngắt, ngoài chứa số hiệu ngắt (IRQ number) còn chứa nội dung gì?
   1. Tên của chương trình xử lý ngắt tương ứng.
   2. Các lời gọi hệ thống liên quan ngắt tương ứng.
   3. Địa chỉ nhớ nơi chứa chương trình xử lý ngắt tương ứng.
   4. Các trình xử lý ngắt của hệ thống.
6. Cơ chế DMA thực hiện nhiệm vụ gì?
   1. quản lý địa chỉ bộ nhớ mà không có sự trợ giúp của CPU
   2. chuyển dữ liệu mà không có sự trợ giúp của CPU
   3. tính toán dữ liệu mà không có sự trợ giúp của CPU
   4. tính toán ngoài thiết bị nhập xuất
7. Cấu trúc chung của một phần cứng I/O có khả năng giao tiếp với máy tính thường gồm các

lớp nào?

* 1. lớp driver và lớp các bộ điều khiển.
  2. lớp I/O Subsystem và Kernel
  3. lớp thiết bị vật lý và lớp các bộ điều khiển.
  4. lớp thiết bị vật lý và lớp nhân (kernel).

1. Giao diện điều khiển nhập xuất (I/O interface) của Hệ điều hành dùng làm gì?
   1. Vận hành một cổng, bus hoặc một thiết bị.
   2. Điều hòa tốc độ truyền dữ liệu.
   3. Chuyển định dạng dữ liệu.
   4. Chuyển định dạng địa chỉ.
2. Trình tự phân lớp để Kernel của Hệ điều hành truy cập được thiết bị I/O (hardware)
   1. Kernel – I/O subsystem – Driver – Device Controller – Device (hardware).
   2. Kernel – Driver – I/O subsystem – Device Controller – Device (hardware).
   3. Kernel – Driver – Device Controller – I/O subsystem – Device (hardware).
   4. Kernel – I/O subsystem – Device Controller – Driver –Device (hardware).
3. I/O subsystem của Hệ điều hành có chức năng gì?
4. Mã hóa dữ liệu nhập xuất
5. Quản lý dữ liệu nhập xuất giữa CPU và Memory
6. Trao đổi thông tin giữa máy tính với thiết bị ngoại vi
7. Xây dựng cơ chế giao tiếp dữ liệu trong đời sống thực
8. Hệ điều hành dùng các thành phần nào để quản lý hệ thống nhập xuất (Input / Output)

(*Input* / *Output*):

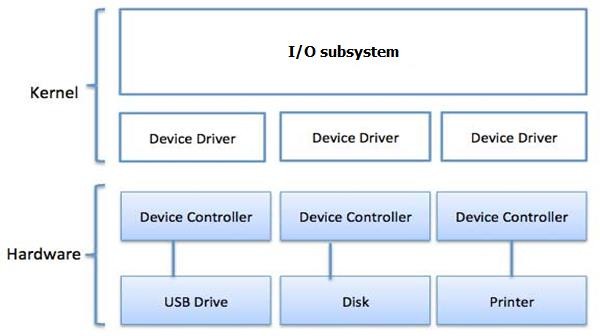
* 1. I/O Subsystem. drivers, bộ nhớ đệm.
  2. Các cổng, bàn phím, màn hình, chuột.
  3. Card song song và nối tiếp, cổng USB.
  4. CPU, RAM, bàn phím, chuột, ROM.

1. I/O Subsystem của Hệ điều hành sử dụng vùng nhớ SPOOL để làm gì?
   1. làm vùng nhớ đệm dữ liệu xuất cho một thiết bị, như máy in
   2. làm vùng nhớ đệm dữ liệu tính toán phục vụ cho CPU
   3. làm vùng nhớ lưu trữ dữ liệu đã từng truy xuất
   4. không có vùng nhớ này trong hệ thống.
2. I/O Subsystem của Hệ điều hành sử dụng vùng nhớ CACHE để làm gì?
   1. làm vùng nhớ đệm dữ liệu xuất cho một thiết bị, như máy in
   2. làm vùng nhớ đệm dữ liệu tính toán phục vụ cho CPU
   3. làm vùng nhớ lưu trữ dữ liệu đã từng truy xuất.
   4. không có vùng nhớ này trong hệ thống.
3. I/O Subsystem của Hệ điều hành sử dụng vùng nhớ BUFFER để làm gì?
   1. làm vùng nhớ đệm dữ liệu xuất cho một thiết bị, như máy in
   2. làm vùng nhớ đệm lưu các tiến trình đang chò phục vụ của I/O.
   3. làm vùng nhớ lưu trữ dữ liệu đã từng truy xuất.
   4. không có vùng nhớ này trong hệ thống.
4. Khi có cùng lúc nhiều tiến trình yêu cầu truy xuất I/O (gọi là I/O requests), bộ I/O

Subsystem của Hệ điều hành sẽ làm gì?

* 1. Đưa các tiến trình đó vào bộ nhớ Cache.
  2. Đưa các tiến trình đó vào bộ nhớ Spool.
  3. Định thời cho các I/O requests để đưa vào hàng đợi Ready.
  4. Định thời cho các I/O requests để đưa vào hàng đợi I/O.

1. Bảng dưới là phân lớp giao tiếp I/O của Hệ điều hành. Hãy cho biết: một ứng dụng chạy trên Hệ điều hành đó cần làm gì để giao tiếp I/O?





* 1. Sử dụng System call để chuyển yêu cầu vào I/O subsystem.
  2. Sử dụng System call để chuyển yêu cầu vào Device Driver.
  3. Gởi lệnh yêu cầu xuống cho Driver.
  4. Gởi lệnh yêu cầu xuống cho Device Controller.

***CHƯƠNG 4***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1A** | **2C** | **3A** | **4B** | **5A** | **6A** | **7B** | **8A** | **9B** | **10C** | **11D** | **12D** | **13A** |
| **14A** | **15B** | **16C** | **17B** | **18D** | **19C** | **20B** | **21C** | **22D** | **23B** | **24A** | **25C** | **26B** |
| **27D** | **28A** | **29B** | **30C** | **31A** | **32B** | **33C** | **34A** | **35D** | **36C** | **37D** | **38A** | **39C** |
| **40B** |  |  |  |  |  |  |  |  |  |  |  |  |

***CHƯƠNG 6***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1B** | **2A** | **3C** | **4B** | **5D** | **6D** | **7A** | **8C** | **9C** | **10A** | **11D** | **12C** | **13D** |
| **14A** | **15D** | **16D** | **17C** | **18B** | **19A** | **20B** | **21C** | **22D** | **23B** | **24A** | **25D** | **26C** |
| **27A** | **28B** | **29C** | **30B** | **31C** | **32A** | **33B** | **34C** | **35C** | **36B** | **37A** | **38A** | **39B** |
| **40C** | **41D** | **42B** | **43C** |  |  |  |  |  |  |  |  |  |

***CHƯƠNG 7***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1D** | **2C** | **3A** | **4B** | **5C** | **6A** | **7B** | **8B** | **9D** | **10C** | **11C** | **12A** | **13B** |
| **14C** | **15C** | **16D** | **17A** | **18C** | **19B** | **20B** | **21A** | **22D** | **23B** | **24D** | **25C** | **26A** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **27C** | **28C** | **29B** | **30A** | **31C** | **32A** | **33B** | **34C** | **35B** | **36C** | **37D** | **38B** | **39A** |
| **40B** | **41A** | **42C** | **43A** |  |  |  |  |  |  |  |  |  |



***CHƯƠNG 8***

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1C** | **2D** | **3C** | **4B** | **5A** | **6C** | **7D** | **8A** | **9B** | **10D** | **11B** | **12C** | **13A** |
| **14B** | **15B** | **16A** | **17C** | **18B** | **19C** | **20A** | **21A** | **22C** | **23A** | **24A** | **25C** | **26B** |
| **27D** | **28A** |  |  |  |  |  |  |  |  |  |  |  |