

# **ĐỀ CƯƠNG ÔN TẬP NGUYÊN LÝ HỆ ĐIỀU HÀNH**

## **I - LÝ THUYẾT**

### **Câu 1: Các thành phần của hệ điều hành, nhân hệ điều hành, tải hệ điều hành.**

Trả lời

#### **1. Các thành phần của HĐH**

##### **a. Hệ thống quản lý tiến trình**

- Tạo lập, hủy bỏ một tiến trình
- Tạm dừng, tái kích hoạt một tiến trình
- Cung cấp các cơ chế trao đổi thông tin giữa các tiến trình
- Cung cấp cơ chế đồng bộ hóa các tiến trình

##### **b. Hệ thống quản lý bộ nhớ**

- Cấp phát và thu hồi vùng nhớ cho các tiến trình khi cần thiết
- Ghi nhận tình trạng bộ nhớ chính: vùng đã cấp phát, vùng còn có thể sử dụng...
- Quyết định tiến trình nào được nạp vào bộ nhớ chính khi có một vùng nhớ trống.

##### **c. Hệ thống quản lý nhập xuất**

- Gửi các lệnh điều khiển đến các thiết bị
- Tiếp nhận các ngắt
- Xử lý lỗi

##### **d. Hệ thống quản lý tập tin**

- Tạo lập, hủy bỏ một tập tin hoặc một thư mục
- Cung cấp các thao tác xử lý tập tin và thư mục
- Tạo lập quan hệ tương ứng giữa tập tin và bộ nhớ phụ chứa nó

##### **e. Hệ thống bảo vệ**

Xây dựng cơ chế bảo vệ thích hợp (Trong trường hợp nhiều người cùng sử dụng đồng thời các tiến trình)

##### **f. Hệ thống dịch lệnh (Shell)**

- Đóng vai trò giao diện giữa người sử dụng và HĐH
- Các lệnh được chuyển đến HĐH dưới dạng chỉ thị điều khiển
- Shell nhận lệnh và thông dịch lệnh để HĐH có xử lý tương ứng

##### **g. Quản lý mạng**

- Một hệ thống phân bố nhiều bộ xử lý với các bộ nhớ độc lập
- Các tiến trình trong hệ thống có thể kết nối với nhau qua mạng truyền thông.
- Việc truy xuất đến tài nguyên mạng thông qua các trình điều khiển giao tiếp mạng.

#### **2. Nhân của HĐH (Kernel)**

- Là thành phần trung tâm của hầu hết các HĐH máy tính
- Nhân của HĐH là các module của hệ thống luôn luôn có mặt trong bộ nhớ trong.
- Chia làm 2 loại nhân:
  - \* Nhân bắt buộc phải có ở bộ nhớ trong
  - \* Nhân khi cần thiết mới gọi vào
- Nhân của HĐH thông thường bao gồm:

\* Module chương trình tải (Loader): Chức năng là đưa một chương trình vào bộ nhớ trong bắt đầu từ địa chỉ nào đó để sau đó cho phép chương trình đã được tải nhận điều khiển để chạy hoặc không.

\* Module chương trình dẫn dắt (Monitor): Chức năng là lựa chọn các bước làm việc cho hệ thống

\* Module chương trình lập dịch (Scheduler): Chức năng chọn chương trình tiếp theo để chạy.

- Nhân là vùng nhớ liên tục và không gây cản trở cho việc cấp phát các chương trình và được đặt ở các vùng biên của bộ nhớ trong và đặc biệt là vùng biên dưới.

### **3. Tải HĐH**

HĐH không thể tự đặt ngay trong máy tính được. Do HĐH là tập hợp các chương trình được cài đặt sẵn nhưng ở trên vật dẫn ngoài (các file trên đĩa từ) và muốn máy tính hoạt động được phải qua một giai đoạn đưa HĐH vào trong máy tính. Quá trình này gọi là tải HĐH và được thực hiện qua các bước sau:

- Khởi động chương trình nguyên thủy. Chương trình này có tên là IPL (Initial Program Loader) đã được cứng hóa sẽ tự kích hoạt để thực hiện mỗi khi bật máy gây xung đột điện.

- IPL kiểm tra tính sẵn sàng của các hệ thống thiết bị

- Các chương trình điều khiển thiết bị cũng được tải vào và các chỉ dẫn đến chúng được thiết lập

- IPL tải đoạn chương trình "mồi" NIP (Nucleus Initial Program) , thường được đặt ở sector đầu tiên ở đĩa chứa HĐH vào bộ nhớ trong và truyền địa chỉ định sẵn và truyền điều khiển cho đoạn chương trình mồi. (chương trình này còn được gọi là chương trình khởi động nhân). Đoạn chương trình mồi NIP thực hiện chức năng tải nhân vào HĐH

- Sau khi tải nhân xong, chương trình mồi NIP sẽ trả quyền điều khiển cho chương trình dẫn dắt hệ thống làm việc.

### **Câu 2: Liệt kê sơ bộ một số hệ điều hành và đặc trưng của hệ điều hành đó.**

Trả lời:

#### **\* MS\_DOS:**

- HĐH đĩa từ Microsoft (Microsoft Disk Operating System, gọi tắt là MS\_DOS) là HĐH của hãng phần mềm Microsoft. Đây là một HĐH có giao diện dòng lệnh (command-line interface) được thiết kế cho các máy tính họ PC (Personal Computer).

- MS\_DOS là HĐH đơn nhiệm. Tại mỗi thời điểm chỉ thực hiện một thao tác duy nhất.

- Quá trình định dạng đĩa từ (Đĩa mềm hay đĩa cứng logic) trong MS\_DOS sẽ chia không gian đĩa đó ra làm 2 phần cơ bản là: Vùng hệ thống (System Area) và vùng dữ liệu (Data Area). Đồng thời hệ thống ghi các thông tin cần thiết vào vùng hệ thống để chuẩn bị cho việc lưu trữ và quản lý dữ liệu sau này.

+ Vùng dữ liệu: gồm các block (cluster) có kích thước bằng nhau và được đánh địa chỉ (12 hay 16 bit) để phân biệt. Đây chính là các Cluster trên đĩa.

+ Vùng hệ thống: Bao gồm các thành phần như Boot Sector, bảng FAT1, bảng FAT2, Root Directory(RD) chứa các chương trình, các thông tin liên quan đến file, directory để giúp hệ điều hành quản lý các file và directory sau này.

**\* Microsoft Window:**

- Microsoft Windows là tên của các dòng phần mềm HĐH độc quyền của hãng MicroSoft
- Windows là phần mềm nguồn đóng
- Windows là HĐH đa nhiệm (Multi tasking) có thể xử lý nhiều chương trình cùng một lúc.
- Windows khởi đầu được phát triển cho những máy tính tương thích với IBM (dựa vào kiến trúc x86 của Intel), và ngày nay hầu hết mọi phiên bản của Windows đều được tạo ra cho kiến trúc này (tuy nhiên Windows NT đã được viết như là một hệ thống xuyên cấu trúc cho bộ xử lý Intel và MIPS, và sau này đã xuất hiện trên các cấu trúc PowerPC và DEC Alpha. Sự phổ biến của Windows đã khiến bộ xử lý trung ương của Intel trở nên phổ biến hơn và ngược lại. Thật vậy, thuật ngữ Wintel đã được sử dụng để miêu tả những máy tính cá nhân đang chạy một phiên bản của Windows.
- Windows có giao diện dễ sử dụng, bắt mắt với độ đồ họa cao

**\* Window Vista:**

- Là HĐH của hãng Microsoft
- Win Vista có giao diện với đồ họa đẹp với hiệu ứng Flip 3D
- Nhưng Win Vista cũng có những hạn chế như tốn nhiều tài nguyên máy và rối rắm hơn Windows XP, tình trạng treo máy thường xuyên xảy ra

**\* Linux:**

- Linux là tên gọi của một hệ điều hành máy tính và cũng là tên hạt nhân của hệ điều hành. Nó có lẽ là một ví dụ nổi tiếng nhất của phần mềm tự do và của việc phát triển mã nguồn mở
- Khởi đầu, Linux được phát triển cho dòng vi xử lý 386, hiện tại hệ điều hành này hỗ trợ một số lượng lớn các kiến trúc vi xử lý, và được sử dụng trong nhiều ứng dụng khác nhau từ máy tính cá nhân cho tới các siêu máy tính và các thiết bị nhúng như là các máy điện thoại di động.
- Đặc tính nổi bật của Linux so với các hệ thống khác: chi phí phần cứng thấp, tốc độ cao (khi so sánh với các phiên bản Unix độc quyền) và khả năng bảo mật tốt, độ tin cậy cao (khi so sánh với Windows) cũng như là các đặc điểm về giá thành rẻ, không bị phụ thuộc vào nhà cung cấp. Một đặc tính nổi trội của nó là được phát triển bởi một mô hình phát triển phần mềm nguồn mở hiệu quả.

**\* Unix:**

- Là một hệ điều hành đa nhiệm (có thể cùng lúc thực hiện nhiều nhiệm vụ) hỗ trợ một cách lý tưởng đối với các ứng dụng nhiều người dùng.
- Unix được viết bằng ngôn ngữ lập trình C, một ngôn ngữ rất mạnh và mềm dẻo. Unix hỗ trợ các ứng dụng mạng và hỗ trợ nhiều môi trường lập trình khác nhau.

- Unix là một trong những hệ điều hành 64 bit đầu tiên.
- Hiện nay Unix được sử dụng bởi nhiều công ty tập đoàn lớn trên thế giới vì mức độ bảo mật của nó tương đối cao.

#### **\* Uturbu:**

- Ubuntu là phần mềm mã nguồn mở tự do, có nghĩa là người dùng được tự do chạy, sao chép, phân phối, nghiên cứu, thay đổi và cải tiến phần mềm.
- Ubuntu kết hợp những đặc điểm nổi bật của dòng Linux, như tính bảo mật trước mọi virus và malware, khả năng tùy biến cao, tốc độ, hiệu suất làm việc, và những đặc điểm của Ubuntu như giao diện bắt mắt, bóng bẩy, cài đặt đơn giản, và sự dễ dàng trong việc sao lưu dữ liệu.
- Ubuntu sử dụng giao diện đồ họa thân thiện GNOME, qua đó hướng đến sự đơn giản hóa trong quá trình sử dụng.

### **Câu 3: Khái niệm tiến trình và tiểu trình. Các trạng thái của tiến trình. Cấu trúc dữ liệu của khối quản lý tiến trình.**

Trả lời:

#### **1. Khái niệm tiến trình, tiểu trình**

- Tiến trình (Process):
  - \* Tiến trình là chương trình đang thực hiện. Mỗi tiến trình có một tài nguyên và môi trường riêng. Các tiến trình hoàn toàn độc lập với nhau, có thể liên lạc với nhau thông qua cơ chế truyền tin giữa các tiến trình.
  - \* Tiến trình hệ thống: được sinh ra khi thực hiện các lời gọi hệ thống
  - \* Tiến trình người sử dụng: được sinh ra khi thực thi các thị chỉ của người sử dụng
  - \* Tiến trình được chia làm 2 loại:
    - + Tiến trình kế tiếp
    - + Tiến trình song song
- Tiểu trình (Threads):
  - \* Tiểu trình là một đơn vị cơ bản của hệ thống. Một tiểu trình cũng có thể tạo lập một tiến trình con.
  - \* Một tiến trình có thể sở hữu nhiều tiểu trình
  - \* Các tiểu trình trong cùng một tiến trình có thể chia sẻ một không gian địa chỉ hoặc truy xuất đến một Stack cùng nhau.

#### **2. Các trạng thái của tiến trình**

- Mới tạo: Tiến trình đang được tạo lập
- Running: Tiến trình đang được xử lý
- Blocked: Tiến trình bị chặn, không thể tiếp tục
- Ready: Tiến trình đang sẵn sàng, chờ cấp CPU để xử lý
- Kết thúc: Tiến trình đã hoàn tất xử lý

#### **3. Cấu trúc dữ liệu của khối quản lý tiến trình.**

HHĐ quản lý các tiến trình thông qua khối quản lý tiến trình (Process Control Block - PCB). Thành phần chủ yếu của PCB bao gồm:

- Định dạng của tiến trình: Phân biệt các tiến trình

- Trạng thái tiến trình: Xác định hoạt động hiện hành của tiến trình
- Ngưỡng cảnh của tiến trình: Mô tả các tài nguyên tiến trình đang trong quá trình hoặc để phục vụ cho hoạt động hiện tại hoặc để làm cơ sở phục hồi hoạt động cho tiến trình, bao gồm các thông tin:
  - \* Trạng thái CPU
  - \* Bộ xử lý
  - \* Bộ nhớ chính
  - \* Tài nguyên sử dụng
  - \* Tài nguyên tạo lập
- Thông tin giao tiếp: Phản ánh các thông tin về quan hệ của tiến trình với các tiến trình khác trong hệ thống:
  - \* Tiến trình cha: Tiến trình tạo lập tiến trình này
  - \* Tiến trình con: Các tiến trình do tiến trình này tạo lập
  - \* Độ ưu tiên: Giúp bộ điều phối có thông tin để lựa chọn tiến trình được cấp CPU
- Thông tin thống kê: Là những thông tin thống kê về hoạt động của tiến trình như thời gian đã sử dụng CPU, thời gian chờ... Các thông tin này sẽ đánh giá tình hình hệ thống và dự đoán các tình huống tương lai.

**Câu 4: Tài nguyên Găng. Điều bộ của tiến trình qua đoạn Găng (Giải pháp phần cứng, Semaphore).**

Trả lời

**1. Tài nguyên Găng**

- Tài nguyên Găng là các tài nguyên logic và vật lý phân bổ cho các tiến trình là song hành.
- Miền Găng là đoạn lệnh trong chương trình có khả năng phát sinh mâu thuẫn. Để không xảy ra mâu thuẫn truy xuất, cần đảm bảo tại một thời điểm chỉ có một tiến trình được vào miền Găng.

**2. Điều bộ của tiến trình qua giai đoạn Găng**

**2.1 Giải pháp phần cứng: 2 giải pháp**

\* Cấm ngắt

- Cho phép tiến trình cấm tất cả các ngắt trước khi vào miền Găng, và phục hồi ngắt khi ra khỏi miền Găng. Khi đó hệ thống không thể tạm dừng hoạt động của tiến trình đang xử lý để cấp phát CPU cho tiến trình khác, nhờ đó tiến trình hiện hành yên tâm thao tác trên miền Găng mà không sợ bị tiến trình khác tranh chấp

- Hạn chế: + Thiếu thận trọng khi cho tiến trình người dùng được phép thực hiện lệnh ngắt
  - + Lệnh cấm ngắt chỉ có tác dụng trên bộ xử lý đang xử lý tiến trình, các tiến trình hoạt động trên các bộ xử lý khác vẫn có thể truy xuất miền Găng nếu hệ thống có nhiều bộ xử lý

\* Chỉ thị TSL (Test and Set Lock)

- Giải pháp đòi hỏi sự trợ giúp của cơ chế phần cứng. Máy tính cung cấp một chỉ thị đặc biệt cho phép kiểm tra và

cập nhập nội dung một vùng nhớ trong một thao tác không thể phân chia gọi là chỉ thị TSL

- Nếu có 2 chỉ thị TSL xử lý đồng thời chúng sẽ được xử lý tuần tự

- Hạn chế: + Không dễ dàng cài đặt chỉ thị TSL

- + Phải thực hiện một vòng lặp để kiểm tra xem tiến trình có được phép vào miền Găng không. Nếu chưa phải chờ tiếp tục trong vòng lặp kiểm tra.

## **2.2 Giải pháp Semaphore**

- Được Dijkstra đề xuất vào năm 1965, một Semaphore  $s$  là một biến có các thuộc tính

- + Một giá trị nguyên dương  $e(s)$

- + Một hàng đợi  $f(s)$  lưu danh sách các tiến trình đang bị khóa trên Semaphore

- + Chỉ có hai thao tác được định nghĩa trên Semaphore

- Semaphore cho phép đảm bảo nhiều tiến trình cùng truy xuất đến miền Găng mà không có sự mâu thuẫn truy xuất.  $n$  tiến trình cùng sử dụng một Semaphore.

- Với Semaphore có thể đồng bộ hóa hoạt động của hai tiến trình trong tình huống một tiến trình phải đợi một tiến trình khác hoàn tất thao tác nào đó mới có thể bắt đầu hoặc tiếp tục xử lý.

- Việc sử dụng Semaphore đúng cách để đồng bộ hóa phụ thuộc hoàn toàn vào người lập trình viên

## **Câu 5: Tắc nghẽn ( Điều kiện hình thành và nhận biết tắc nghẽn).**

Trả lời

### **1. Định nghĩa**

Tắc nghẽn là trạng thái khi hai hoặc nhiều tiến trình cùng chờ đợi một số sự kiện nào đó và nếu không có sự tác động đặc biệt từ bên ngoài thì sự chờ đợi đó là vô hạn.

### **2. Điều kiện hình thành tắc nghẽn**

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

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

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

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

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

### **3. Phương pháp xử lý tắc nghẽn**

Có 3 phương pháp xử lý tắc nghẽn:

- Sử dụng nghi thức (Protocol) để đảm bảo hệ thống không bao giờ xảy ra tắc nghẽn
- Cho phép xảy ra tắc nghẽn và tìm cách sửa chữa tắc nghẽn
- Bỏ qua việc xử lý tắc nghẽn

#### **4. Ngăn chặn tắc nghẽn**

Để tắc nghẽn không bao giờ xảy ra, cần đảm bảo tối thiểu 1 trong 4 điều kiện cần không xảy ra

- Mô phỏng các tài nguyên không thể chia sẻ thành các tài nguyên có thể dùng chung được (áp dụng phương pháp Sooling)
- Thực hiện phân bổ trước tài nguyên: Tiến trình chỉ có thể thực hiện khi đã cung cấp đủ tài nguyên. Tiến trình chỉ được phép đòi tài nguyên khi chúng không giữ tài nguyên nào. Nếu tiến trình phải đợi tài nguyên thì mọi tài nguyên của tiến trình phải được giải phóng
- Cho phép hệ thống thu hồi tài nguyên khi tiến trình bị khóa và cập nhập lại cho tiến trình khi chúng thoát khỏi tình trạng khóa
- Cấp phát tài nguyên theo một sự phân cấp nhất định như sau:  
 $R = \{R_1, R_2, \dots, R_m\}$  tập các loại tài nguyên. Các loại tài nguyên được phân cấp từ 1 - N

#### **Câu 6: Nhiệm vụ quản lý bộ nhớ.**

Trả lời

- Bộ nhớ là thiết bị lưu trữ duy nhất mà thông qua đó CPU có thể trao đổi thông tin với môi trường ngoài
  - Bộ nhớ chính được tổ chức như một mảng một chiều các từ nhớ (word), mỗi từ nhớ có một địa chỉ
  - Việc trao đổi thông tin với môi trường bên ngoài được thực hiện thông qua các thao tác đọc hoặc ghi dữ liệu vào một địa chỉ nhất định trong bộ nhớ.
- => Vì thế tổ chức, quản lý bộ nhớ là nhiệm vụ trọng tâm hàng đầu của HĐH
- \* Nhiệm vụ của quản lý bộ nhớ:
- Cấp phát và thu hồi vùng nhớ cho các tiến trình khi cần thiết
  - Ghi nhận tình trạng bộ nhớ chính: vùng đã cấp phát, vùng còn có thể sử dụng...
  - Quyết định tiến trình nào được nạp vào bộ nhớ chính khi có một vùng nhớ trống.
  - HĐH chịu trách nhiệm cấp phát vùng nhớ cho các tiến trình có yêu cầu. Để thực hiện tốt HĐH cần phải xem xét nhiều khía cạnh như:
    - + Sự tương ứng giữa địa chỉ Logic và địa chỉ vật lý: làm thế nào để chuyển đổi một địa chỉ tương trưng trong chương trình thành địa chỉ thực trong bộ nhớ
    - + Quản lý bộ nhớ vật lý: làm sao để mở rộng bộ nhớ có sẵn nhằm lưu trữ nhiều tiến trình đồng thời
    - + Chia sẻ thông tin: Làm sao cho phép 2 tiến trình chia sẻ thông tin trong bộ nhớ

+ Bảo vệ: làm thế nào để ngăn chặn các tiến trình xâm phạm đến vùng nhớ được cấp phát cho tiến trình khác

### **Câu 7: Cấu trúc chương trình.**

Trả lời:

Có nhiều phương pháp tổ chức chương trình để thực hiện. Các phương pháp này khác nhau ở kiểu định vị chương trình trong bộ nhớ và thời điểm thực hiện phép ánh xạ địa chỉ tương đối thành địa chỉ tuyệt đối.

Cấu trúc chương trình thể hiện cách quản lý bộ nhớ logic và cho ta thấy hình ảnh của chương trình ở bộ nhớ vật lý khi thực hiện. Cấu trúc chương trình có thể ở các dạng sau:

\* Cấu trúc tuyến tính

- Là cấu trúc mà sau khi biên dịch, các module được tập hợp thành chương trình hoàn thiện, chứa đầy đủ mọi thông tin để có thể thực hiện

- Ưu điểm: Đơn giản, dễ tổ chức biên dịch và định vị, thời gian thực hiện nhanh, có tính lưu động cao và dễ dàng sao chép chương trình.

- Nhược điểm: Lãng phí bộ nhớ.

\* Cấu trúc động

- Các module được tổ chức riêng biệt. Khi thực hiện chương trình, hệ thống chỉ cần định vị module gốc. Trong quá trình thực hiện cần tới module nào thì hệ thống cấp phát không gian nhớ và khi hoạt động xong thì giải phóng module khỏi bộ nhớ và thu hồi không gian nhớ.

- Ưu điểm: Tiết kiệm bộ nhớ

- Nhược điểm: Xóa và nạp các module là trách nhiệm do người sử dụng đảm nhiệm, do đó các câu lệnh phải được nêu ngay trong chương trình nguồn => kích thước chương trình nguồn lớn.

\* Cấu trúc Overlay

- Các module chương trình sau khi biên dịch được chia thành các mức 0,1,...i. Khi thực hiện chương trình, module gốc được định vị vào bộ nhớ. Cần tới module nào hệ thống sẽ tìm kiếm trong sơ đồ Overlay và nạp vào bộ nhớ ở mức tương ứng.

- Ưu điểm: Có tính chất định vị động, do đó cho phép sử dụng bộ nhớ nhiều hơn.

- Nhược điểm: Hiệu quả tiết kiệm bộ nhớ vẫn phụ thuộc vào cách tổ chức, bố trí các module của chương trình

\* Cấu trúc phân đoạn

- Chương trình của người sử dụng được biên dịch thành từng module độc lập. Thông tin về các module được chứa trong 1 bảng điều khiển gọi là bảng quản lý đoạn (Segment Control Block - SCB). Khi thực hiện chương trình, hệ thống sẽ dựa vào SCB để nạp các module cần thiết vào trong bộ nhớ cho tới khi hết khả năng. Nếu cần nạp các module mới nhưng thiếu bộ nhớ thì hệ thống sẽ đưa bớt ra ngoài các module có khả năng không sử dụng nữa.

- Ưu điểm: Khi dung lượng bộ nhớ tăng thì tốc độ thực hiện chương trình cũng tăng



- Nhược điểm: Hiệu quả sử dụng bộ nhớ phụ thuộc vào cách phân chia chương trình thành các module độc lập và chỉ áp dụng được khi bộ nhớ quản lý theo kiểu phân đoạn

\* Cấu trúc phân trang

- Chương trình được biên dịch như cấu trúc tuyến tính, sau đó được phân chia thành các phần bằng nhau gọi là trang. Thông tin các trang được chứa trong một bảng điều khiển gọi là bảng quản lý trang (Page Control Block - PCB). Khi thực hiện, hệ thống sẽ dựa vào bảng quản lý trang để nạp các trang cần thiết vào bộ nhớ.

- Ưu điểm: Phát huy được hiệu quả sử dụng của bộ nhớ

- Nhược điểm: Chỉ áp dụng được khi bộ nhớ quản lý theo kiểu phân trang

### **Câu 8: Phân biệt kỹ thuật phân vùng cố định và phân vùng động.**

#### **Cơ chế quản lý bộ nhớ.**

Trả lời:

#### **1. Phân biệt kỹ thuật phân vùng cố định và phân vùng động.**

##### **a. Phân vùng cố định**

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

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

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

##### **b. Phân vùng động**

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

- Ưu điểm: Tận dụng được không gian nhớ tự do.

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

#### **2. Cơ chế quản lý bộ nhớ**

HĐH chia không gian nhớ thành các khối nhớ bằng nhau (block) và quản lý theo danh sách liên kết hoặc quản lý bằng bảng các bit

### **Câu 9: Kỹ thuật phân trang. Bảng trang nhị cấp, tam cấp, bảng trang nghịch đảo.**

Trả lời:

#### **1. Kỹ thuật phân trang**

- Phân bộ nhớ vật lý thành các khối (Block) có kích thước cố định và bằng nhau, gọi là khung trang. Không gian địa chỉ cũng được chia thành các khối có cùng kích thước với khung trang và được gọi là trang. Khi cần nạp một tiến trình để xử lý các

trang của tiến trình sẽ được nạp vào những khung trang còn trống. Một tiến trình có kích thước N trang sẽ yêu cầu N khung trang tự do

- Cơ chế MMU (Memory Management Unit) là cơ chế phần cứng hỗ trợ thực hiện chuyển đổi địa chỉ trong cơ chế phân trang gọi là bảng trang. Mỗi phần tử trong bảng trang cho biết địa chỉ bắt đầu của vị trí lưu trữ trang tương ứng trong bộ nhớ vật lý
- Mỗi địa chỉ phát sinh bởi CPU được chia làm 2 phần:

- + Số hiệu trang (p)

- + Địa chỉ tương ứng trong trang (d)

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

2. Bảng trang nhị cấp, tam cấp, bảng trang nghịch đảo

- Bảng trang đa cấp: phân chia các bảng trang thành các phần nhỏ, bản thân bảng trang cũng sẽ được phân trang

Bảng trang nhị cấp

Bảng trang tam cấp

~~~~~

- Bảng trang nghịch đảo: Sử dụng duy nhất một bảng trang nghịch đảo cho tất cả các tiến trình. Mỗi phần tử trong bảng nghịch đảo phản ánh một khung trang trong bộ nhớ bao gồm địa chỉ logic của một trang được lưu trữ trong bộ nhớ vật lý tại khung trang này, cùng với thông tin về tiến trình đang được sở hữu trang. Mỗi địa chỉ ảo khi đó là bộ ba <idp,p,d> trong đó

- + idp: định danh của tiến trình

- + p: số hiệu trang

- + d: địa chỉ tương đối trong trang.

Mỗi phần tử trong bảng trang nghịch đảo là một cặp <idp,p>

Câu 10: Kỹ thuật bộ nhớ ảo. Lỗi trang.

Trả lời:

1. Kỹ thuật bộ nhớ ảo

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

- Kỹ thuật :

- + Kết hợp kỹ thuật Swapping đến chuyển các phần của chương trình vào - ra giữa bộ nhớ chính và bộ nhớ phụ khi cần thiết

+ Nhờ việc tách biệt bộ nhớ ảo và bộ nhớ vật lý, có thể tổ chức một bộ nhớ ảo có kích thước lớn hơn bộ nhớ vật lý

+ Cho phép giảm nhẹ công việc của lập trình viên.

- Cài đặt bộ nhớ ảo: Bộ nhớ ảo thường được thực hiện với kỹ thuật phân trang theo yêu cầu. Cũng có thể sử dụng kỹ thuật phân đoạn theo yêu cầu để cài đặt bộ nhớ ảo.

## 2. Hiện tượng lỗi trang

Khi hệ thống truy xuất đến một trang nhưng trang này chưa được nạp vào bộ nhớ trong sẽ làm phát sinh lỗi trang. HĐH sẽ xử lý lỗi trang theo các bước sau:

- B1: Kiểm tra truy xuất đến bộ nhớ có hợp lệ không. Nếu có đến B2, ngược lại kết thúc chương trình

- B2: Tìm vị trí chứa trang cần truy xuất trên đĩa từ

- B3: Tìm một trang vật lý trống trong bộ nhớ chính. Nếu tìm thấy đến B4, ngược lại chọn một trang đang sử dụng chuyển ra bộ nhớ ngoài, cập nhật bảng quản lý trang tương ứng

- B4: Chuyển trang muốn truy xuất từ bộ nhớ ngoài vào bộ nhớ trong: Nạp trang cần truy xuất vào khung trang trống (hoặc vừa làm trống), cập nhật bảng quản lý trang.

- B5: Tái kích hoạt chương trình

Câu 11: Quản lý tập tin (Bảng danh mục, tập tin chia sẻ, các block: quản lý, cấp phát liên tục theo danh sách liên kết).

Trả lời:

### 1. Bảng danh mục

- Bảng danh mục bao gồm các phần tử, mỗi một phần tử để chứa thông tin về một tập tin, thư mục trên đĩa.

- HĐH quản lý các tập tin, thư mục này đang tồn tại trên đĩa thông qua bảng danh mục: HĐH phải truy xuất đến tập tin phải thông qua bảng danh mục, khi có 1 tập tin được tạo ra thì HĐH phải thêm phần tử vào bảng danh mục và xóa phần tử ở bảng danh mục khi tập tin này bị xóa khỏi đĩa

- Số lượng các phần tử này không hạn chế.

- Bảng danh mục có thể nằm ở một vị trí đặc biệt nào đó trên đĩa. Khi hoạt động bảng danh mục phải được nạp vào bộ nhớ

- Mỗi phần tử trong bảng danh mục phải chứa thông tin tối thiểu của file như tên file, kiểu file, địa chỉ vật lý...

- Bảng danh mục có 2 mức:

+ Mức 1: Bảng danh mục chỉ bao gồm các con trỏ trỏ đến bảng danh mục mức 2

+ Mức 2: Bảng danh mục của người sử dụng bao gồm các thông tin về tập tin

### 2. Tập tin chia sẻ

- Được sử dụng trong HĐH đa nhiệm nhằm giúp nhiều người sử dụng cùng một tập tin nào đó trên hệ thống.

- Tập tin nay xuất hiện đồng thời trong các thư mục khác nhau của người sử dụng khác nhau

### 3. Block

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

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

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

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

Câu 12: Kể tên một số hệ thống file được sử dụng trên hệ điều hành hiện nay.

Trả lời:

\* FAT (File Allocation Table)

- FAT gồm nhiều phần tử. Chiều dài mỗi phần tử được tính bằng Bit, biểu thị số đếm của bảng FAT.

- Mỗi loại FAT có 1 con số để chỉ ra số lượng Bit mà hệ thống file để nhận dạng các chearter.

- + FAT 12: Được dùng cho ổ đĩa mềm, ổ đĩa có dung lượng từ 32Mb trở xuống. FAT 12 sử dụng 12 Bit để đếm nên chỉ có khả năng quản lý các ổ đĩa có dung lượng thấp hơn 32Mb và số lượng các Cluster thấp.

- + FAT 16: Với HĐH MS-DOS, hệ thống tập tin FAT 16 đưa ra một cách thức mới về việc ổ đĩa và quản lý tập tin trên đĩa cứng, đĩa mềm. Tuy nhiên khi dung lượng đĩa ngày càng tăng nhanh FAT 16 đã bộc lộ hạn chế. Với không gian địa chỉ 16 Bit, FAT 16 chỉ hỗ trợ đến 65536 Cluster trên một Partition, gây ra sự lãng phí dung lượng.

- + FAT 32: Do sử dụng không gian địa chỉ 32 Bit nên FAT 32 hỗ trợ nhiều Cluster trên một Partition hơn, do đó không gian đĩa cứng được tận dụng nhiều hơn. Ngoài ra với khả năng hỗ trợ kích thước của phân vùng từ 2Gb - 2Tb và chiều dài tối đa của tập tin được mở rộng đến 255 kí tự. Tuy nhiên FAT 32 có nhược điểm là không có tính bảo mật và khả năng chịu lỗi không cao.

\* NTFS (New Technology File System)

- Là hệ thống file dành riêng cho Win NT hoặc Win 2000 trở lên.

- NTFS dùng 64 Bit để định danh nên nó có thể quản lý dung lượng ổ đĩa lên đến 16 tỉ Gb.

- NTFS sử dụng bảng quản lý tập tin MFT (Master File Table) thay cho bảng FAT quen thuộc nhằm tăng cường khả năng lưu trữ, tính bảo mật cho tập tin và thư mục, khả năng mã hóa dữ liệu đến từng tập tin. Ngoài ra, NTFS có khả năng chịu lỗi cao, cho phép người dùng đóng một ứng dụng "chết" (not responding) mà không làm ảnh hưởng đến những ứng dụng khác. Tuy nhiên, NTFS lại không thích hợp với những ổ đĩa có dung lượng thấp (dưới 400 MB) và không sử dụng được trên đĩa mềm.

- Tên file trong NTFS có độ dài không quá 255 ký tự và có đường dẫn. Tên file có sự phân biệt chữ thường chữ hoa
- \* CDFS (CD File System)
  - Là hệ thống quản lý file trên các đĩa CD\_ROM
  - CDFS sử dụng một bộ điều khiển VCACHE để điều khiển đĩa CD\_ROM
  - Tên file không quá 32 kí tự cây thư mục không quá 8 mức
- \* UDF (Universal Disk Format)
  - Là hệ thống quản lý file trên DVD\_ROM
  - Tên file dài 255 ký tự, đường dẫn có thể dài 1032 ký tự (trên 8 mức)

Câu 13: NTFS. Cửa sổ bảng FAT. Nêu trình tự xóa 1 file và ghi 1 file trên đĩa.

Trả lời:

1. NTFS

2. Cửa sổ bảng FAT

- FAT được đặt trên volume ngay sau vùng để dành, nếu không có vùng để dành nó nằm sau Boot Sector. Mỗi đề mục của FAT tương đương với một Cluster, mỗi cluster có thể chứa một hoặc nhiều Sector kế tiếp nhau. Số lượng các sector được tạo lên một cluster được đặt trong ô nhớ 0Dh của Boot Sector, giá trị này luôn là lũy thừa của 2 và thay đổi tùy theo loại đĩa
- Bảng FAT là một danh sách móc nối các Cluster trên đĩa
- Ý nghĩa của các phần tử trong bảng FAT:

Mã số Ý nghĩa

- (0)000h Cluster tự do
- (F)FF0h - (F)FF6h Cluster để dành
- (F)FF7h Cluster bị hỏng, không dung
- (F)FF8h - (F)FFFh Cluster cuối cùng của một file
- (X)XXXh Cluster tiếp theo của file

- Bảng FAT biểu diễn một "bản đồ" lưu trữ File trong vùng DATA

3. Trình tự xóa 1 file, ghi & đọc 1 file trên đĩa

- Ghi 1 file trên đĩa: Đầu tiên HĐH sẽ tìm xem có file nào cùng tên đã tồn tại trên cùng thư mục định ghi chưa. Nếu chưa thì nó sẽ tiến hành ghi, nếu có nó sẽ hỏi bạn có muốn ghi đè không.

Nếu sau bước trên mà nhận được lệnh ghi, HĐH sẽ tìm một khoảng không gian trống trên đĩa đủ để lưu file rồi Creat file, ghi dữ liệu lên. Cuối cùng tất nhiên là phải đóng file lại.

- Xóa 1 file trên đĩa: Đầu tiên khi xóa file, HĐH sẽ tìm file cần xóa trên đĩa, mở file đó ra, xóa toàn bộ mọi thông tin trong file, cuối cùng là đóng file lại rồi xóa mọi dấu vết của file để giải phóng không gian đĩa. Nếu có trục trặc trong việc xóa thông tin trong file (thường là Data file) thì file sau khi đóng sẽ không xóa được.

\* Cách lưu trữ dữ liệu trên đĩa

Đa số ổ cứng trên thị trường hiện nay dùng công nghệ ghi gọi là theo chiều dọc, ở đó các bit được lưu trữ bên cạnh nhau trên bề mặt từ tính .

Công nghệ ghi theo chiều dọc có từ khi bắt đầu có ổ cứng. Và công nghệ ghi mới hiện nay gọi là ghi vuông góc, công nghệ này được các ổ cứng mới ngày nay sử dụng , nó cho phép ghi với mật độ cao .

Để hiểu dữ liệu được đọc, viết trong ổ cứng chúng ta cần nhớ đặc điểm sau :

- Tất cả các dây dẫn tạo ra một từ trường xung quanh chúng khi có dòng điện chạy qua.
- Một từ trường đủ mạnh có thể sinh ra dòng điện trong một dây dẫn.
- Chiều của từ trường phụ thuộc vào chiều của dòng điện và ngược lại.

Có một vật liệu dẫn hình chữ U có cuộn dây dẫn xung quanh (nam châm hình móng ngựa) để làm đầu đọc ghi trên ổ cứng. Trong quá trình ghi số liệu lên ổ cứng, một dòng điện được cung cấp vào cuộn dây dẫn tạo ra một từ trường xung quanh đầu đọc ghi.

Trường này sẽ từ hoá bề mặt bên dưới đầu đọc ghi, những hạt từ tính được sắp thành hàng, chúng theo chiều trái hoặc phải phụ thuộc vào chiều của dòng điện được cung cấp qua cuộn dây dẫn. Bit được lưu trữ liên tiếp trong các hạt từ tính.

Trong quá trình đọc dữ liệu trên ổ cứng, khi đầu đọc ghi qua vùng có từ trường xuất hiện dòng điện trong cuộn dây dẫn, chiều của dòng điện phụ thuộc vào chiều của từ trường mà đầu đọc ghi đi qua, nó cho phép mạch điều khiển ổ đĩa đọc được Bit đã lưu trữ.

## II - BÀI TẬP

### 1. Các chiến lược điều phối

Bài 1 / t12

| Tiến trình | Thời gian vào RL | Thời gian CPU | Độ ưu tiên |
|------------|------------------|---------------|------------|
| P1         | 0                | 10            | 3          |
| P2         | 1                | 1             | 1          |
| P3         | 2                | 2             | 3          |
| P4         | 3                | 1             | 4          |
| P5         | 4                | 5             | 2          |

#### a. Điều phối tiến trình

\* FIFO: (Vào trước ra trước)

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| P1 | P2 | P3 | P4 | P5 |    |
| 0  | 10 | 11 | 13 | 14 | 19 |

\* Theo độ ưu tiên độc quyền

- Độ ưu tiên: độ ưu tiên của tiến trình nào cao hơn có trong CPU thì xử lý trước. độ ưu tiên  $1 > 2 > 3 > 4 \dots$

- Độc quyền: đang xử lý tiến trình nào thì xử lý hết tiến trình đó mới xử lý tiến trình khác.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| P1 | P2 | P5 | P3 | P4 |    |
| 0  | 10 | 11 | 16 | 18 | 19 |

\* SJF không độc quyền

- SJF: Xử lý tiến trình có "thời gian CPU" xử lý ngắn nhất trong các tiến trình có trong CPU.

- SJF không độc quyền: Tiến trình nào vào sau nhưng có "thời gian CPU" xử lý ngắn nhất thì đc xử lý trước. Các tiến trình trước nó thì bị đẩy về đằng sau.

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| P1 | P2 | P3 | P4 | P5 | P1 |    |
| 0  | 1  | 2  | 4  | 5  | 10 | 19 |

\* SJF độc quyền (ghấn 2 định nghĩa bên trên)

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| P1 | P2 | P4 | P3 | P5 |    |
| 0  | 10 | 11 | 12 | 14 | 19 |

\* RR ( $q = 2$ ) (Sau thời gian " $q = 2$ " tiến trình chưa xử lý xong cũng phải nhường CPU cho tiến trình tiếp theo. Cứ vậy cho đến khi tất cả các tiến trình được xử lý hết.)

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| P1 | P2 | P3 | P4 | P5 | P1 | P5 | P1 | P5 | P1 |
| 0  | 2  | 3  | 5  | 6  | 8  | 10 | 12 | 14 | 15 |

19

b. Thời gian lưu lại trong hệ thống ( = thời gian xử lý xong - thời gian tiến trình đó vào.)

FIFO Theo độ ưu tiên độc quyền SJF không độc quyền SJF độc quyền RR ( $q = 2$ )

|         |         |
|---------|---------|
| P1 - 10 |         |
| P2 - 10 |         |
| P3 - 11 |         |
| P4 - 11 |         |
| P5 - 15 | P1 - 10 |
| P2 - 10 |         |
| P3 - 16 |         |
| P4 - 16 |         |
| P5 - 12 | P1 - 19 |
| P2 - 1  |         |
| P3 - 2  |         |
| P4 - 2  |         |
| P5 - 6  | P1 - 10 |
| P2 - 9  |         |

P3 - 12  
 P4 - 9  
 P5 - 15            P1 - 19  
 P2 - 1  
 P3 - 3  
 P4 - 3  
 P5 - 11  
 T = 11,12        T = 12,8        T = 6   T = 11   T = 7,4

c. Thời gian chờ trong hệ thống: ( = thời gian có mặt trong CPU mà không được CPU xử lý.)

FIFO Theo độ ưu tiên độc quyền SJF không độc quyền SJF độc quyền RR ( $q = 2$ )

P1 - 0  
 P2 - 9  
 P3 - 9  
 P4 - 10  
 P5 - 10            P1 - 0  
 P2 - 9  
 P3 - 14  
 P4 - 15  
 P5 - 7 P1 - 9  
 P2 - 0  
 P3 - 0  
 P4 - 1  
 P5 - 1 P1 - 0  
 P2 - 9  
 P3 - 10  
 P4 - 8  
 P5 - 10            P1 - 9  
 P2 - 1  
 P3 - 1  
 P4 - 2  
 P5 - 6  
 T = 7,6            T = 9   T = 2.2        T = 7,4        T = 3,8

d. Với thuật toán điều phối SJF không độc quyền cho thời gian chờ trung bình là cực tiểu

2. Bài tập về các thuật toán cấp phát bộ nhớ và nêu ưu nhược điểm các thuật toán.

3. Các bài tập về thuật toán thay trang.

Bài 11 / t29

Xét chuỗi truy xuất bộ nhớ sau: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 3, 7, 6, 3, 2, 1, 2, 3, 6

a. Thuật toán LRU

- Trang lâu nhất chưa được sử dụng sẽ bị thay thế (so sánh ô đó với 3 ô trước nó theo hàng ngang)



|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 |
| 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |   |
| 1 | 1 | 1 | 4 | 4 | 4 | 5 | 5 | 5 | 1 |
| 1 | 1 | 6 | 6 | 6 | 1 | 1 | 1 | 6 |   |
|   | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |   |
|   |   | 3 | 3 | 3 | 1 | 1 | 1 | 2 | 2 |
| 2 | 7 | 7 | 7 | 2 | 2 | 2 | 2 | 2 |   |
| * | * | * | * |   | * | * | * | * | * |
| * | * | * |   | * | * |   |   | * |   |

- Có 15 lỗi trang

b. Thuật toán FIFO

- Trang ở bộ nhớ lâu nhất sẽ bị thay thế. (theo hàng dọc)

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 |
| 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |   |
| 1 | 1 | 1 | 4 | 4 | 4 | 4 | 6 | 6 | 6 |
| 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 6 |   |
|   | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 |
| 2 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 |   |
|   |   | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 1 |
| 1 | 1 | 6 | 6 | 6 | 6 | 6 | 3 | 3 |   |
| * | * | * | * |   | * | * | * | * | * |
| * | * | * |   | * | * |   | * | * |   |

- Có 16 lỗi trang

c. Chiến lược tối ưu

- Trang sẽ lâu được sử dụng nhất trong tương lai sẽ bị thay thế

(so sánh ô đó với cột dọc trước nó, nếu ko có thì thay thế => xét hàng ngang sau nó: thay thế số trong cột dọc xa nhất có trong hàng ngang sau nó.)

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 |
| 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |   |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 6 |   |
|   | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 7 | 7 | 7 | 2 | 2 | 2 | 2 | 2 |   |
|   |   | 3 | 4 | 4 | 4 | 5 | 6 | 6 | 6 |
| 6 | 6 | 6 | 6 | 6 | 1 | 1 | 1 | 1 |   |
| * | * | * | * |   |   | * | * |   |   |
| * | * |   |   | * | * |   |   | * |   |

- Có 11 lỗi trang

4. Bài tập 7 ( Trang 28 - Quản lý bộ nhớ)

Tóm tắt

Page Swap - 8 ms = 8000 ns

Page modify - 20 ms = 20000 ns

Memory R/W - 100 ns

Modify rate - 70%

Nếu Memory R/W < 200 ns thì Page swap rate (p) = ?

Giải:

$Tea = (1-p)ma + p(tdp)[+swap\ out][+swap\ in] + \text{tái kích hoạt}$

Có  $tdp = 8000.30\% + 20000.70\% = 16400\ ns$

Vậy

$$200 < (1-p).100 + p.16400 \Rightarrow p > 0.006135$$