**BÁO CÁO HỌC TẬP HỌC KỲ 20172**

GIẢNG VIÊN: NGUYỄN TUẤN ANH

HỌC PHẦN: HỆ ĐIỀU HÀNH

Họ Tên: NGUYỄN VĂN LÂN

MSSV: 20162338

Lớp: CNTT1.01-K61

*Đề Tài*: *Tài Nguyên Bộ Nhớ*

Tìm hiểu về ổ đĩa cứng, đọc và hiển thị thông tin đĩa viết bằng ngôn ngữ lập trình C.

**PHẦN 1.** **Tổng Quan Về Ổ Đĩa Cứng**

**I.Tổng Quan**

Ổ đĩa cứng, hay còn gọi là ổ cứng (tiếng Anh: Hard Disk Drive, viết tắt: HDD) là thiết bị dùng để lưu trữ dữ liệu trên bề mặt các tấm đĩa hình tròn (bằng nhôm, thủy tinh hay gốm) phủ vật liệu từ tính. Ổ đĩa cứng là loại bộ nhớ “không thay đổi” (nonvolatile), có nghĩa là chúng không bị mất dữ liệu khi ngừng cung cấp nguồn điện cho chúng. Về mặt kỹ thuật thì ổ đĩa cứng là một khối duy nhất, các phiến đĩa được lắp ráp cố định trong ổ ngay từ khi sản xuất nên không thể thay thế được các “đĩa cứng” như cách hiểu đối với ổ đĩa mềm hoặc ổ đĩa quang (CD/DVD) Bên trong một ổ đĩa cứng các bộ phận chính: Cơ cấu truyền động đầu từ (actuator) điều khiển cánh tay đầu từ (actuator arm) đọc thông tin từ phiến đĩa (plater) được quay liên tục nhờ gắn “chết” vào trục quay của động cơ liền trục (spindle motor). Dữ liệu được truyền ra ngoài nhờ cáp nối có dạng dãi băng mềm (ribbon cable). Vỏ ổ đĩa (chassis) bao giữ tất cả các bộ phận của đĩa cứng. Ổ đĩa cứng là một thiết bị rất quan trọng trong hệ thống bởi chúng chứa dữ liệu thành quả của một quá trình làm việc của những người sử dụng máy tính. Những sự hư hỏng của các thiết bị khác trong hệ thống máy tính có thể sửa chữa hoặc thay thế được, nhưng dữ liệu bị mất do yếu tố hư hỏng phần cứng của ổ đĩa cứng thường rất khó lấy lại được. Các công ty chế tạo IDE đã cố gắng khắc phục khoảng cách tốc độ này bằng phương pháp đánh địa chỉ logic khối (Logical Block Addressing - LBA). Các ổ đĩa này được gọi là EIDE. Cùng lúc với sự ra đời của EIDE, các nhà sản xuất SCSI đã tiếp tục cải tiến tốc độ SCSI. Những cải tiến đó đồng thời khiến cho giá thành của giao tiếp SCSI cao thêm.

**II. Các loại Ổ Đĩa**

1. *Ổ đĩa SISC*

Để có thể vừa nâng cao hiệu suất của EIDE vừa không làm tăng chi phí cho các linh kiện điện tử không có cách nào khác là phải thay giao diện kiểu parallel (song song) bằng kiểu serial (nối tiếp), và kết quả là sự ra đời của giao diện SATA

1. *Ổ đĩa SATA*

Ổ đĩa SATA Trước đây, các chuẩn ATA (IDE, EIDE) và SATA thế hệ đầu tiên được sử dụng phổ biến trong máy tính cá nhân thông thường trong khi chuẩn SCSI có tốc độ cao hơn được sử dụng chủ yếu trong máy chủ và máy trạm. Gần đây, các chuẩn SATA thế hệ tiếp theo với tốc độ giao tiếp cao hơn đang được sử dụng rộng rãi trong các máy tính cá nhân sử dụng các thế hệ chipset mới.

**III. Các thông số của ổ đĩa**

1. *Chuẩn Giao tiếp*

Có nhiều chuẩn giao tiếp khác nhau giữa ổ đĩa cứng bo mạch chủ, sự đa dạng này một phần xuất phát từ yêu cầu tốc độ đọc/ghi dữ liệu khác nhau giữa các hệ thống máy tính, phần còn lại do các ổ giao tiếp nhanh có giá thành cao hơn nhiều so với các chuẩn thông dụng. Ba chuẩn thông dụng hiện nay là EIDE, SCSI, và SATA.

1. *Dung Lượng*

Dung lượng: Dung lượng của ổ đĩa cứng được tính theo các đơn vị dung lượng cơ bản thông thường là Byte, KB, MB, GB, TB. Trước đây, khi dung lượng ổ cứng còn thấp người ta thường dùng đơn vị là MB. Bây giờ, người ta lại dùng đơn vị là GB và trong tương lai, chắc người ta sẽ tính theo TB. Đa số các hãng sản xuất đều tính dung lượng theo cách tính 1GB = 1000MB trong khi hệ điều hành (hoặc các phần mềm kiểm tra) lại tính 1GB = 1024MB nên dung lượng do hệ điều hành báo cáo thường thấp hơn so với dung lượng ghi trên nhãn đĩa (ví dụ ổ đĩa cứng 40 GB thường chỉ đạt khoảng 37-38 GB).

1. *Tốc Độ Quay*

Tốc độ quay của đĩa cứng được ký hiệu là rpm (revolutions per minute - số vòng quay trong một phút). Tốc độ quay càng cao thì ổ đĩa làm việc càng nhanh do chúng thực hiện việc đọc/ghi nhanh hơn, thời gian tìm kiếm thấp hơn. Các tốc độ quay thông dụng hiện nay là 5.400 rpm (thông dụng với các ổ đĩa cứng 3,5” sản xuất cách đây 2-3 năm) và 7.200 rpm (thông dụng với các ổ đĩa cứng sản xuất từ 2008). Ngoài ra, tốc độ của các ổ đĩa cứng trong các máy tính cá nhân cao cấp, máy trạm và các máy chủ có sử dụng giao tiếp SCSI có thể lên tới 10.000 rpm hay 15.000 rpm.

1. *Bộ Nhớ Đệm (cache hoặc buffer)*

Bộ nhớ đệm có nhiệm vụ lưu tạm dữ liệu trong quá trình làm việc của ổ đĩa cứng nên độ lớn của bộ nhớ đệm có ảnh hưởng đáng kể tới hiệu suất hoạt động của ổ đĩa cứng bởi việc đọc/ghi không xảy ra tức thời (do phụ thuộc vào sự di chuyển của đầu đọc/ghi, dữ liệu được truyền tới hoặc đi) sẽ được đặt tạm trong bộ nhớ đệm. Trong thời điểm năm 2007, dung lượng bộ nhớ đệm thường là 2 hoặc 8 MB cho các loại ổ đĩa cứng dung lượng đến 160 GB và 16 MB hoặc cao hơn cho các ổ đĩa cứng dụng lượng lớn hơn.

1. *Tốc Độ Truyền Dữ Liệu*

Đa phần tốc độ truyền dữ liệu trên các chuẩn giao tiếp thấp hơn so với thiết kế của nó bởi có nhiều thông số ảnh hưởng đến tốc độ truyền dữ liệu của ổ đĩa cứng như: tốc độ quay của đĩa từ, số lượng đĩa từ trong ổ đĩa cứng, công nghệ chế tạo, dung lượng bộ nhớ đệm…

1. *Kích Thước*

Để đảm bảo thay thế lắp ráp vừa với các loại máy tính, kích thước của ổ đĩa cứng được chuẩn hoá thành 6 loại là: 5,25 inch dùng trong các máy tính các thế hệ trước. 3,5 inch dùng cho các máy tính cá nhân, máy trạm, máy chủ. 2,5 inch dùng cho máy tính xách tay. 1,8 inch hoặc nhỏ hơn dùng trong các thiết bị kỹ thuật số cá nhân và PC Card. 1,0 inch dùng cho các thiết bị siêu nhỏ (micro device). Kích thước ổ đĩa cứng 1,8”, 2,5” và 3,5” (từ trái qua phải)

**IV. Các Công Nghệ Đặc Biệt Của Ổ Đĩa Cứng**

1. *S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology)*

Là công nghệ tự động giám sát tình trạng hoạt động của ổ đĩa cứng để thông qua BIOS, các phần mềm thông báo cho người sử dụng biết trước các hư hỏng có thể xẩy ra để có các hành động chuẩn bị đối phó (như sao chép dữ liệu dự phòng hoặc có các kế hoạch thay thế ổ đĩa cứng mới). S.M.A.R.T chỉ giám sát những sự thay đổi, ảnh hưởng của phần cứng đến quá trình lỗi xảy ra của ổ đĩa cứng bao gồm những hư hỏng theo thời gian của phần cứng như: đầu đọc/ghi (mất kết nối, khoảng cách làm việc với bề mặt đĩa thay đổi), động cơ (xuống cấp, mòn trục), bo mạch của ổ đĩa (hư hỏng linh kiện hoặc làm việc sai). S.M.A.R.T không hề ảnh hưởng đến tốc độ làm việc và truyền dữ liệu của ổ đĩa cứng. Gần đây S.M.AR.T được coi là một tiêu chuẩn quan trọng trong ổ đĩa cứng và người sử dụng có thể bật (enable) hoặc tắt (disable) chức năng này trong BIOS.

1. *Ổ Cứng Lai (hybrid hard disk drive)*

Là các ổ đĩa cứng thông thường được gắn thêm các phần bộ nhớ flash trên bo mạch của ổ đĩa cứng. Cụm bộ nhớ flash này hoạt động khác với cơ chế làm việc của bộ nhớ đệm (cache) của ổ đĩa cứng do dữ liệu chứa trên chúng không bị mất đi khi mất điện nguồn. Vai trò của phần bộ nhớ flash là: Lưu trữ dữ liệu trước khi ghi vào đĩa cứng, chỉ khi máy tính đã đưa các dữ liệu đến một mức nhất định (tuỳ từng loại ổ cứng lai) thì ổ đĩa cứng mới tiến hành ghi dữ liệu vào các đĩa từ, điều này giúp sự vận hành của ổ đĩa cứng hiệu quả và tiết kiệm điện năng hơn nhờ việc không phải thường xuyên hoạt động. Giúp tăng tốc độ giao tiếp với máy tính vì việc đọc dữ liệu từ bộ nhớ flash nhanh hơn so với việc đọc dữ liệu tại các đĩa từ. Giúp hệ điều hành khởi động nhanh hơn nhờ việc lưu các tập tin khởi động của hệ thống vào bộ nhớ flash. Kết hợp với bộ nhớ đệm của ổ đĩa cứng tạo thành một hệ thống hoạt động hiệu quả. Những ổ cứng lai được sản xuất hiện nay thường sử dụng bộ nhớ flash với dung lượng không lớn (128, 256 MB) vì giá thành sản xuất bộ nhớ flash quá cao. Do sử dụng dung lượng nhỏ như vậy nên tốc độ làm việc chưa được cải thiện nhiều và người sử dụng thường khó nhận ra sự hiệu quả của chúng khi thực hiện các tác vụ thông thường. Do ổ cứng lai có giá thành khá đắt (khoảng 300 USD cho dung lượng 32 GB) nên chúng chỉ mới được sử dụng trong một số loại máy tính xách tay cao cấp.

1. *Ổ Cứng SSD (Solid-State Drive)*

Ổ cứng SSD :(Solid-State Drive) Là ổ cứng không sử dụng đĩa từ mà hoàn toàn dựa trên bộ nhớ flash NAND. Do đó, ổ cứng SSD có độ bền cao khi di chuyển (do không có bộ phận cơ khí), tiêu thụ ít điện năng hơn, nhẹ hơn, nhỏ hơn và tốc độ làm việc nhanh hơn ổ cứng thông thường rất nhiều. Hiện giá SSD vẫn còn cao so với túi tiền của người tiêu dùng, dù vẫn giảm 30%/năm. Ví dụ, loại 64 GB giá khoảng 900 USD, 128 GB giá 1.500 USD trong khi ổ cứng thông thường dung lượng tương đương chỉ có giá trên dưới 100 USD

**V. Các Thành Phần Cấu Tạo Ổ Đĩa Cứng**

*1. Cum đĩa từ(platter)*

Thường bằng nhôm hoặc thuỷ tinh, trên bề mặt được phủ một lớp vật liệu từ tính để chứa dữ liệu. Tuỳ theo hãng sản xuất mà số lượng đĩa có thể nhiều hơn một và các đĩa này được sử dụng một hoặc cả hai mặt trên và dưới. Các đĩa từ được gắn song song, quay đồng trục, cùng tốc độ với nhau khi hoạt động. Động cơ và trục quay: các đĩa từ được gắn lên trục quay nối trực tiếp với động cơ quay đĩa cứng. Trục quay có nhiệm vụ truyền chuyển động quay từ động cơ đến các đĩa từ. Trục quay được chế tạo bằng các vật liệu nhẹ như hợp kim nhôm và được chế tạo tuyệt đối chính xác để đảm bảo trọng tâm của chúng không bị sai lệch.

1. *Cụm đầu đọc ghi (head)*

Đầu đọc đơn giản được cấu tạo gồm lõi ferit và cuộn dây (giống như nam châm điện). Đầu đọc trong đĩa cứng có công dụng đọc dữ liệu dưới dạng từ hoá trên bề mặt đĩa từ hoặc từ hoá lên các mặt đĩa khi ghi dữ liệu. Số đầu đọc ghi luôn bằng số mặt hoạt động được của các đĩa cứng. Cần di chuyển đầu đọc/ghi (head arm hoặc actuator arm): Đầu đọc/ghi được gắn vào cần di chuyển đầu đọc/ghi. Cần có nhiệm vụ di chuyển theo phương song song với các đĩa từ ở một khoảng cách nhất định, dịch chuyển và định vị chính xác đầu đọc tại các vị trí từ mép đĩa đến vùng phía trong của đĩa (phía trục quay). Các cần di chuyển đầu đọc được di chuyển đồng thời với nhau do chúng được gắn chung trên một trục quay (đồng trục).

1. *Cụm mạch điện mạch điều khiển*

Có nhiệm vụ điều khiển động cơ đồng trục, điều khiển sự di chuyển của cần di chuyển đầu đọc để đảm bảo đến đúng vị trí trên bề mặt đĩa. Mạch xử lý dữ liệu: dùng để xử lý những dữ liệu đọc/ghi của ổ đĩa cứng. Bộ nhớ đệm (cache hoặc buffer): là nơi tạm lưu dữ liệu trong quá trình đọc/ghi dữ liệu. Dữ liệu trên bộ nhớ đệm sẽ mất đi khi ổ đĩa cứng ngừng được cấp điện. Đầu cắm nguồn cung cấp điện cho ổ đĩa cứng. Đầu kết nối giao tiếp với máy tính. Các cầu nối (jumper): Lựa chọn chế độ làm việc của ổ đĩa cứng (SATA 150 hoặc SATA 300) hay thứ tự trên các kênh trên giao tiếp IDE (master hay slave hoặc tự lựa chọn), lựa chọn các thông số làm việc khác…

1. *Vỏ đĩa cứng*

Phần đế chứa các linh kiện gắn trên nó, phần nắp đậy lại để bảo vệ các linh kiện bên trong. Vỏ ổ đĩa cứng có chức năng chính nhằm định vị các linh kiện, chịu đựng sự va chạm (ở mức độ thấp) để bảo vệ ổ đĩa cứng và đảm bảo độ kín khít để không cho phép bụi được lọt vào bên trong của ổ đĩa cứng. Trên vỏ bảo vệ có các lỗ thoáng đảm bảo cản bụi và cân bằng áp suất giữa môi trường bên ngoài và môi trường không khí có độ sạch cao bên trong.

**VI. Câu Trúc Dữ Liệu Của Đĩa Cứng**

1. *Track*

Trên một mặt làm việc của đĩa từ chia ra nhiều vòng tròn đồng tâm tạo thành các track (rãnh) để xác định các vùng lưu trữ dữ liệu riêng biệt trên mặt đĩa. Track trên ổ đĩa cứng không cố định từ khi sản xuất, chúng có thể thay đổi vị trí khi định dạng cấp thấp ổ đĩa (low format). Khi một ổ đĩa cứng đã hoạt động nhiều năm liên tục, phần cơ của nó sẽ bị bào mòn và làm việc không chính xác như khi mới sản xuất. Nếu kiểm tra bằng các phần mềm cho thấy nhiều vùng trên đĩa bị hư hỏng (bad block) thì nên tín hành format cấp thấp cho nó để tái cấu trúc lại các track cho phù hợp hơn với sự xuống cấp của phần cơ.

1. *Sector*

Mỗi track được chia thành những phần nhỏ bằng các đoạn hướng tâm tạo thành các sector (cung từ). Sector là đơn vị chứa dữ liệu nhỏ nhất. Theo chuẩn thông thường thì một sector chứa (dung lượng) 512 byte. Số sector trên các track từ phần rìa đĩa vào đến vùng tâm đĩa là khác nhau, các ổ đĩa cứng đều chia ra hơn 10 vùng mà trong mỗi vùng có số sector/track bằng nhau.

1. *Cluster*

Trong lĩnh vực lưu trữ dữ liệu (đĩa mềm hoặc đĩa cứng) ở mức độ hệ điều hành (HĐH), cluster (liên cung) là một đơn vị lưu trữ gồm một hoặc nhiều sector. Khi HĐH lưu trữ một tập tin vào đĩa, nó ghi tập tin đó vào hàng chục, có khi hàng trăm cluster liền nhau. Nếu không sẵn cluster liền nhau, HĐH sẽ tìm kiếm cluster còn trống ở kế đó và ghi tiếp tập tin lên đĩa. Quá trình cứ tiếp tục như vậy cho đến khi toàn bộ được cất giữ hết.

1. *Cylinder*

Tập hợp các track cùng bán kính ở các mặt đĩa khác nhau tạo thành các cylinder (trụ). Trên đĩa hai mặt, một cylinder sẽ bao gồm rãnh 1 của mặt trên và rãnh 1 của mặt dưới. Trên các đĩa cứng sắp xếp cái này chồng lên cái kia, một cylinder gồm các rãnh trên cả hai mặt của tất cả các đĩa. Trên một ổ đĩa cứng có nhiều cylinder bởi có nhiều track trên mỗi mặt đĩa từ.

1. *Đọc và ghi dữ liệu trên bề măt đĩa*

Sự hoạt động của đĩa cứng cần thực hiện đồng thời hai chuyển động: Chuyển động quay của các đĩa và chuyển động ra vô của các đầu đọc. Đĩa từ quay được nhờ gắn cùng trục với động cơ và có tốc độ rất lớn từ 3600 đến 15.000 vòng/phút. Mỗi loại ổ đĩa cứng có một tốc độ nhất định tùy theo công nghệ chế tạo. Khi đĩa cứng quay đều, cần di chuyển đầu đọc sẽ di chuyển đến các vị trí trên bề mặt phủ vật liệu từ theo phương bán kính của đĩa. Chuyển động này kết hợp với chuyển động quay của đĩa có thể làm đầu đọc/ghi tới bất kỳ vị trí nào trên bề mặt đĩa. Tại các vị trí cần đọc ghi, đầu đọc/ghi có các bộ cảm biến với điện trường để đọc hay ghi dữ liệu. Dữ liệu được ghi/đọc đồng thời trên mọi đĩa. Việc thực hiện phân bổ dữ liệu trên các đĩa được thực hiện nhờ các mạch điều khiển trên bo mạch của ổ đĩa cứng.

**PHẦN 2.** Phần Mềm Đọc ***BOOT SECTOR*** Bằng Ngôn Ngữ Lập Trình ***C***

1. *Chức Năng Của Chương Trình*

1-in thông tin của nhà sản xuất và kích thước cluster

2-in nội dung 20 ô đầu tiên của fat

3-in ra tổng số cluster trống trên đĩa

1. *Mã Code*

#[include](http://diendan.congdongcviet.com/)<stdio.h>

#[include](http://diendan.congdongcviet.com/)<conio.h>

#[include](http://diendan.congdongcviet.com/)<dos.h>

[**struct**](http://diendan.congdongcviet.com/) [\_BOOT](http://diendan.congdongcviet.com/)

    {

**char** jmp[3];

**char** nsx[8];

**int** [bytes\_per\_sector](http://diendan.congdongcviet.com/);

**char** [sectors\_per\_cluster](http://diendan.congdongcviet.com/);

**int** [reserved\_sector](http://diendan.congdongcviet.com/);

**char** [num\_of\_fat](http://diendan.congdongcviet.com/);

**int** [num\_of\_kmuc](http://diendan.congdongcviet.com/);

**int** [total\_sector](http://diendan.congdongcviet.com/);

**char** [type\_disk](http://diendan.congdongcviet.com/);

**int** [sectors\_per\_fat](http://diendan.congdongcviet.com/);

**int** [sectors\_per\_track](http://diendan.congdongcviet.com/);

**int** [num\_of\_heads](http://diendan.congdongcviet.com/);

**long** [hidden](http://diendan.congdongcviet.com/);

**long** [total\_sector\_2](http://diendan.congdongcviet.com/);

**char** ex[26];

**char** loader[448];

**char** mark[2];

    }boot;

[**struct**](http://diendan.congdongcviet.com/) ROOT

    {

**char** file\_name[8];

**char** mo\_rong[3];

**char** [thuoc\_tinh](http://diendan.congdongcviet.com/);

**char** du\_phong[10];

**char** [time](http://cppreference.com/stddate/time.html)[2];

**char** day[2];

**int** [stt\_cluster](http://diendan.congdongcviet.com/);

**long** [size\_byte\_file](http://diendan.congdongcviet.com/);

    }root;

**void** [Ham\_1](http://diendan.congdongcviet.com/)()

    {

**if**(absread(2,1,0,&boot)==0)

        {

[printf](http://cppreference.com/stdio/printf.html)("\n Kich [thuoc](http://diendan.congdongcviet.com/) [cluster](http://diendan.congdongcviet.com/): %d [sector](http://diendan.congdongcviet.com/)",boot.[sectors\_per\_cluster](http://diendan.congdongcviet.com/));

[printf](http://cppreference.com/stdio/printf.html)("\n NSX : ");

**for**(**int** i=0;i<8;i++)

[printf](http://cppreference.com/stdio/printf.html)("%c",boot.nsx[i]);

        }

    }

**void** Ham\_2()

    {

    absread(2,1,0,&boot);

**char** \*fat;

    fat =new **char** [5\*boot.[bytes\_per\_sector](http://diendan.congdongcviet.com/)];

*//fat= new char [boot.*[*sectors\_per\_fat*](http://diendan.congdongcviet.com/)*\*boot.*[*bytes\_per\_sector*](http://diendan.congdongcviet.com/)*];*

    absread(2,5,boot.[reserved\_sector](http://diendan.congdongcviet.com/),fat);

*//absread(2,boot.*[*sectors\_per\_fat*](http://diendan.congdongcviet.com/)*,boot.*[*reserved\_sector*](http://diendan.congdongcviet.com/)*,fat);*

**int** \*fat\_int=(**int** \*)fat;

[printf](http://cppreference.com/stdio/printf.html)("\n Noi dung 20 o dau tien");

**for**(**int** i=0;i<20;i++)

[printf](http://cppreference.com/stdio/printf.html)("\t %c",fat\_int[i]);

    delete fat;

    }

**void** Ham\_3()

    {

**int** dem=0;

    absread(2,1,0,&boot);

**char** \*fat;

    fat = new **char** [boot.sectors\_per\_fat\*boot.[bytes\_per\_sector](http://diendan.congdongcviet.com/)];

    absread(2,boot.sectors\_per\_fat,boot.[reserved\_sector](http://diendan.congdongcviet.com/),fat);

**int** \* fat\_int=(**int** \*)fat;

**for**(**int** i=0;i<(boot.[sectors\_per\_fat](http://diendan.congdongcviet.com/)\*boot.[bytes\_per\_sector](http://diendan.congdongcviet.com/))/2;i++)

**if**(fat\_int[i]==0)dem++;

[printf](http://cppreference.com/stdio/printf.html)("\n So [cluster](http://diendan.congdongcviet.com/) con trong tren dia : %d",dem);

    }

**void** main()

    {

**char** p;

    clrscr();

[printf](http://cppreference.com/stdio/printf.html)("\n ==============MENU===========");

[printf](http://cppreference.com/stdio/printf.html)("\n 1-In thong tin cua nsx va kich [thuoc](http://diendan.congdongcviet.com/) [cluster](http://diendan.congdongcviet.com/)");

[printf](http://cppreference.com/stdio/printf.html)("\n 2-In noi dung 20 o dau tien cua bang fat");

[printf](http://cppreference.com/stdio/printf.html)("\n 3-In ra tong so [cluster](http://diendan.congdongcviet.com/) trong tren dia");

[printf](http://cppreference.com/stdio/printf.html)("\n a-Thoat");

[printf](http://cppreference.com/stdio/printf.html)("\n");

[printf](http://cppreference.com/stdio/printf.html)("\n Nhap thao tac");

[scanf](http://cppreference.com/stdio/scanf.html)("%c",&p);

**switch**(p)

        {

**case** '1': [Ham\_1](http://diendan.congdongcviet.com/)();**break**;

**case** '2': Ham\_2();**break**;

**case** '3': Ham\_3();**break**;

**case** 'a': **break**;

        }

    getch();

    }