

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

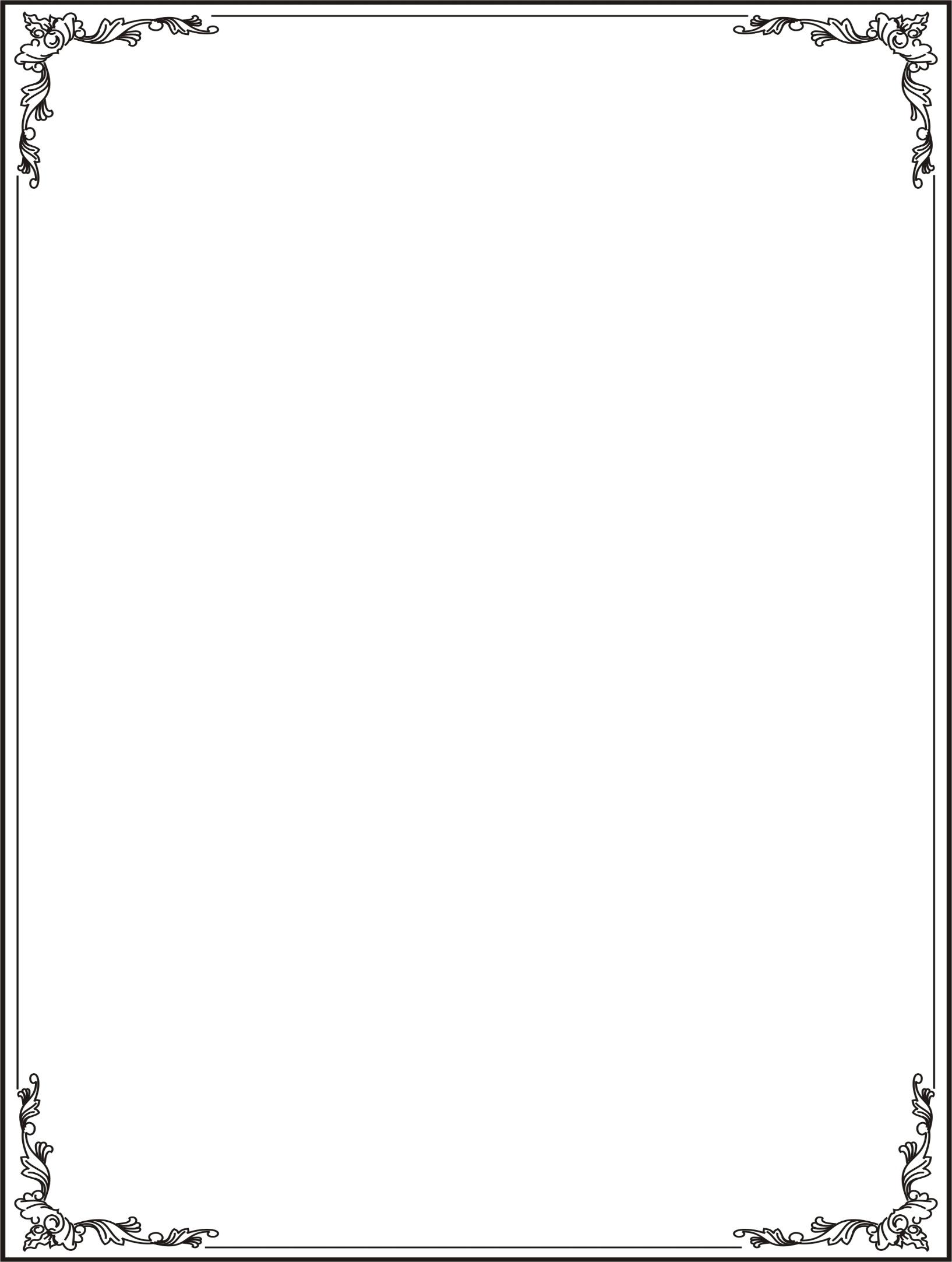
****

**BÀI TẬP LỚN MÔN HỌC HỆ ĐIỀU HÀNH**

**Đề tài: Nghiên cứu tìm hiểu về quản lý thiết bị ngoại vi trong HĐH Linux**

| **GIẢNG VIÊN:** | **Ths Nguyễn Tuấn Tú** |
| --- | --- |
| **NHÓM – LỚP:** | **Nhóm 8 - IT6025.6(006)** |
|  |  |
|  |  |
|  |

Hà Nội, 2022

****

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**BÀI TẬP LỚN MÔN HỌC HỆ ĐIỀU HÀNH**

**Đề tài: Nghiên cứu tìm hiểu về quản lý thiết bị ngoại vi trong HĐH Linux**

| **GIẢNG VIÊN:** | **Ths Nguyễn Tuấn Tú** |
| --- | --- |
| **NHÓM – LỚP:** | **Nhóm 8 - IT6025.6(006)** |
|  |  |
|  |  |
|  |
| **Sinh viên thực hiện :** | **Nguyễn Quỳnh Anh** |
|  | **Lưu Thị Ngọc Ánh** |
|  | **Lưu Văn Linh** |
|  |  |
|  |  |

Hà Nội, 2022

Mục lục

[**Chương 1: Mở đầu 6**](#_heading=h.30j0zll)

[**Chương 2: Nội dung 7**](#_heading=h.1fob9te)

[**1. Sơ lược về hệ điều hành linux và thiết bị ngoại vi 7**](#_heading=h.3znysh7)

[**1.1 Khái niệm Hệ điều hành linux 7**](#_heading=h.2et92p0)

[**1.2 Chức năng của Hệ điều hành 7**](#_heading=h.3dy6vkm)

[**1.3 Lịch sử Hệ điều hành linux 7**](#_heading=h.1t3h5sf)

[**1.4 Như thế nào là thiết bị ngoại vi ? 8**](#_heading=h.4d34og8)

[**2 Các thiết bị ngoại vi 9**](#_heading=h.2s8eyo1)

[**2.1 Thiết bị nhập dữ liệu 9**](#_heading=h.17dp8vu)

[**2.2 Thiết bị xuất dữ liệu 10**](#_heading=h.3rdcrjn)

[**3. Yêu cầu quản lý thiết bị 12**](#_heading=h.26in1rg)

[**4. Nguyên tắc tổ chức và quản lý thiết bị 12**](#_heading=h.lnxbz9)

[**4.1 Linux điều khiển các thiết bị phần cứng 13**](#_heading=h.35nkun2)

[**4.2 Linux quản lý các thiết bị lưu trữ 14**](#_heading=h.1ksv4uv)

[**5. Bộ điều khiển DMA 14**](#_heading=h.44sinio)

[**5.1 Khái niệm DMA 14**](#_heading=h.2jxsxqh)

[**5.2 Hoạt động DMA cơ bản 15**](#_heading=h.3j2qqm3)

[**Hình 2.1 16**](#_heading=h.1y810tw)

[**Hình 2.2 16**](#_heading=h.1ci93xb)

[**Hình 2.3 17**](#_heading=h.2bn6wsx)

[**6. Các kỹ thuật áp dụng trong quản lý thiết bị 17**](#_heading=h.qsh70q)

[**6.1. Kỹ thuật vùng đệm 17**](#_heading=h.3as4poj)

[**6.1.1 Khái niệm và mục đích của vùng đệm 17**](#_heading=h.1pxezwc)

[**6.1.2.phân loại vùng đệm 18**](#_heading=h.49x2ik5)

[**6.2. Kỹ thuật kết khối 20**](#_heading=h.19c6y18)

[**6.3.Xử lý lỗi 21**](#_heading=h.3tbugp1)

[**7. Cách truy xuất đĩa 22**](#_heading=h.nmf14n)

[**8. Các lệnh quản lí thiết bị ngoại vi 23**](#_heading=h.37m2jsg)

[**8.1 Quản lí thiết bị lưu trữ trong Linux 23**](#_heading=h.1mrcu09)

[**8.1.1. Lệnh mount và unmount 23**](#_heading=h.46r0co2)

[**8.1.2. Định dạng đĩa trong Linux 25**](#_heading=h.2lwamvv)

[**8.1.3. Quản lí đĩa 27**](#_heading=h.111kx3o)

[**8.2. Cổng nối tiếp và modem 27**](#_heading=h.3l18frh)

[**8.3. Cổng song song và máy in 29**](#_heading=h.206ipza)

[**8.3.1. Khởi tạo và thiết lập máy in trong lpd 29**](#_heading=h.4k668n3)

[**8.3.2. Các lệnh in ấn cơ bản 29**](#_heading=h.2zbgiuw)

[**8.4. Sound card 30**](#_heading=h.1egqt2p)

[**Chương 3: Kết luận 31**](#_heading=h.3ygebqi)

***Lời nói đầu***

*Linux – hệ điều hành mã nguồn mở từ lâu đã không còn xa lạ với người dùng máy tính, nó thu hút được nhiều sự chú ý nhất trong vòng vài năm trở lại đây. Ngay từ khi xuất hiện, nó đã được lan rộng một cách nhanh chóng và biết tới như một hệ điều hành Unix — với mã nguồn mở . Thật ngạc nhiên, sự thành công của Linux có được nhờ sự làm lại một trong những hệ điều hành lâu đời nhất và hiện đang được sử dụng rộng rãi — hệ điều hành Unix.*

*Linux bao gồm cả các công nghệ cũ và mới. Nhìn từ góc độ kỹ thuật, Linux chỉ là một nhãn hệ điều hành, nó hỗ trợ đầy đủ các phục vụ cơ bản về quản lý tiến trình, bộ nhớ ảo, quản lý file và vào ra thiết bị. Nói cách khác, bản thân Linux là phần thấp nhất của hệ điều hành. Tuy nhiên, còn khá nhiều rắc rối và bất cập khiến HDH miễn phí này chưa thể thay thế hoàn toàn Window là nó khả rắc rối khi cài đặt, cực hình với những dòng lệnh, không thể sử dụng tất cả những ứng dụng có thể chạy trên win và,… đặc biệt là không hỗ trợ hoàn toàn thiết bị ngoại vi. Hầu hết những thiết bị ngoại vi thông thường như modem, máy in, cạc mạng,… đều làm việc tốt dưới Linux. Tuy vậy có vài loại thiết bị ngoại vi làm việc kém, có loại lại không làm việc.*

*Nhằm giới thiệu thêm những kiến thức cơ bản về cách quản lý thiết bị ngoại vi trong hệ điều hành Linux. Nhóm chúng tôi viết bài luận này muốn chia sẻ với các bạn những hiểu biết chúng tôi về cách quản lý thiết bị ngoại vi trong hệ điều hành mà nguồn mở này.*

***Nhóm sinh viên thực hiện!***

# **Chương 1: Mở đầu**

Nhiệm vụ chung của bài: Nghiên cứu tìm hiểu về quản lý thiết bị ngoại vi trong HĐH Linux

Bố cục bao gồm:

* Phần 1: Sơ lược về hệ điều hành linux và thiết bị ngoại vi
* Phần 2: Các thiết bị ngoại vi
* Phần 3: Yêu cầu quản lý thiết bị
* Phần 4: Nguyên tắc tổ chức và quản lý thiết bị
* Phần 5: Cơ chế DMA
* Phần 6: các kỹ thuật áp dụng trong quản lý thiết bị
* Phần 7: Cách truy xuất đĩa
* Phần 8: Các lệnh quản lý thiết bị ngoại vi

# **Chương 2: Nội dung**

## **1. Sơ lược về hệ điều hành linux và thiết bị ngoại vi**

### **1.1 Khái niệm Hệ điều hành linux**

Linux là một hệ điều hành (OS) mã nguồn mở. Là phần mềm quản lý phần cứng và các tài nguyên khác của hệ thống như CPU, bộ nhớ và ổ cứng. Hệ điều này năm giữa ứng dụng và phần cứng tạo sự kết nối giữa tất cả các phần mềm và tài nguyên vật lý phục vụ cho việc thực hiện công việc.

### **1.2 Chức năng của Hệ điều hành**

* Tổ chức giao tiếp giữa người sử dụng và hệ thống.
* Cung cấp bộ nhớ, các thiết bị ngoại vi, ... cho chương trình và tổ chức thực hiện các chương trình đó.
* Tổ chức lưu trữ thông tin trên bộ nhớ ngoài, cung cấp các công cụ tìm kiếm và truy cập thông tin.
* Hỗ trợ phần mềm cho các thiết bị ngoại vi.
* Cung cấp các dịch vụ tiện ích hệ thống.

### **1.3 Lịch sử Hệ điều hành linux**

Linux bắt đầu từ một hệ điều hành lớn hơn có tên Unix. Unix là một trong những hệ điều hành được sử dụng rộng rãi nhất thế giới do tính ổn định và khả năng hỗ trợ của nó. Ban đầu hệ điều hành linux đã được phát triển như một hệ điều hành đa nhiệm cho các máy mini và các máy lớn (mainframe) trong những năm 70. Cho tới nay nó đã được phát triển trở thành một hệ điều hành phổ dụng trên toàn thế giới, mặc dù với giao diện chưa thân thiện và chưa được chuẩn hóa hoàn toàn.

Linux là phiên bản Unix được cung cấp miễn phí, ban đầu được phát triển bởi Linus Torvald năm 1991 khi còn là một sinh viên của trường đại học Helssiki Phần Lan. Hiện nay, Linux làm việc tại tập đoàn Transmeta và tiếp tục phát triển nhân hệ điều hành Linux ( Linux kernel).

Khi linux tung ra phiên bản miễn phí đầu tiên của Linux trên Internet, vô tình đã tạo ra một làn sóng phát triển phần mềm lớn nhất từ trước đến nay trên phạn vi toàn cầu. Hiện nay, Linux được phát triển và bảo trì bởi một nhóm hàng nghìn lập trình viên công tác chặt chẽ với nhau qua Internet. Nhiều công ty đã xuất hiện, cung cấp linux dưới dạng gói phần mềm dễ cài đặt, hoặc cung cấp các máy tính đã cài đặt sẵn Linux.

Tháng 11 năm 1991, Linux đưa ra bản chính thức đầu tiên của Linux, phiên bản 0.02.

Sau 3 năm nhân Linux ra đời, đến ngày 14-3-1994, hệ điều hành Linux phiên bản 1.0 được phổ biến, đây là phiên bản tương đối ổn định.

Vào tháng 3-1995, nhân 1.2 được phổ biến. Điều đáng kể của Linux 1.2 so với Linux 1.0 ở chỗ nó hỗ trợ một phạm vi rộng và phong phú phần cứng, bao gồm cả kiến trúc tuyến phần cứng PCI mới. Nhân Linux 1.2 nhân kết thúc dòng nhân Linux chỉ hỗ trợ PC.

Tháng 6-1996, nhân Linux 2.0 được phổ biến.

Tới năm 2000, nhân Linux 24 được phổ biến.

Với phiên bản Linux 2.2.6, bạn có thể làm việc trên môi trường đồ họa với các ứng dụng cao cấp như: các tiện ích đồ họa và các tiện ích khác.

Hiện nay, Linux là một hệ điều hành Unix đầy đủ và độc lập. Nó có thể chạy X Window, TCP/IP, Emacs, Web, thư điện tử và các phần mềm khác. Hầu hết các phần mềm miễn phí và thương mại đều được chuyển lên Linux.

### **1.4 Như thế nào là thiết bị ngoại vi ?**

Thiết bị ngoại vi là tên chung nói đến một số thiết bị bên ngoài thùng máy được gắn kết với máy tính với tính năng nhập xuất(I/O) hoặc mở rộng khả năng lưu trữ ( như một dạng bộ nhớ phụ).

Thiết bị Thiết ngoại vi của máy tính có thể là:

* Bị cấu thành lên máy tính và không thể thiếu được ở một số loại máy tính.
* Thiết bị có mục đích mở rộng tính năng hoặc khả năng của máy tính.

Có rất nhiều thiết bị ngoại vi của máy tính như: Màn hình máy tính, ổ đĩa mềm, ổ cứng, USB, ổ quang (CD ,DVD), chuột máy tính, bàn phím máy tính, máy in, webcam, modem các loại, loa máy tính, micro...

## **2 Các thiết bị ngoại vi**

### **2.1 Thiết bị nhập dữ liệu**

* Key Board: bàn phím.

Keyboard ra đời từ rất sớm, trong mỗi hệ máy tính hiện nay đều có trang bị bàn phím tiêu chuẩn, những hệ máy tính đặc biệt thì có trang bị bàn phím chuyên dụng.

Keyboard có nhiều loại khác nhau:

* Keyboard tiêu chuẩn
* Keyboard cho máy tính xách tay
* Keyboard ảo
* Mouse: con trỏ chuột.





Con trỏ chuột ra đời muộn hơn Keyboard.

Sự ra đời của con trỏ chuột là một cột mốc trong ngành chế tạo máy tính

Giúp sự điều khiển sử dụng máy tính dễ dàng và tiện lợi hơn.

Từ khi ra đời cho đến nay con trỏ chuột đã có nhiều thay đổi trong công nghệ chế tạo.

Máy tính lớn: chuột quang, chuột laser, chuột bi lăn...

Máy tính nhỏ: track ball, trackPad,chuột cảm ứng

### **2.2 Thiết bị xuất dữ liệu**

* Monitor: màn hình

CRT (Cathode Ray Tube): Màn hình dùng cồn nghệ ống cực Cathode lạnh, ra đời từ rất lâu, hiện công nghệ CRT đang dần được thay thế bằng LCD.

LCD (Liquid Crystal Display): Màn hình tinh thể lỏng ra đời từ sớm, tuy nhiên do hạn chế về tính năng và giá cả, nên LCD mới chỉ được dùng rộng rãi gần đây.

Màn hình cảm ứng: là màn hình CRT hoặc LCD thông thường và được lắp đặt trên tấm màn hình cảm ứng (cảm ứng điện trở hoặc điện dung).

* Printer: máy in được coi là thiết bị xuất dữ liệu cổ xưa, máy in ra đời trước khi mà hình ra đời.
* Máy in Kim: sử dụng ma trận kim và ruy băng mực (giống giấy than) để in ký từ.
* Máy in Phun: dùng công nghệ phun mực trực tiếp lên giấy in. Thường dùng để in ảnh.
* Máy in Laser: dùng công nghệ định vị điểm ảnh bằng tia laze trên trống in để in ảnh, tốc độ cao.
* Máy in Offset: công nghệ cao dùng để in tốc độ nhanh, in chi tiết.
* Ổ quang

A picture containing text, electronics, drive

Description automatically generated

Dùng lưu trữ dữ liệu, Ổ quang là phương tiện lưu trữ dữ liệu hiệu quả, tiện dụng ổ quang thế hệ đầu là CD-ROM có khả năng đọc dữ liệu trên các đĩa compact dung lượng chứa tối đa 800MB dữ liệu.

Thế hệ tiếp theo là công nghệ DVD với dung lượng tiêu chuẩn 4,8 GB đĩa DVD 2 lớp có 9GB dữ liệu, đĩa DVD 2 mặt 2 lớp có 18GB dữ liệu.

Thế hệ mới nhất là HD-DVD và bluray với tiêu chuẩn chưa 24GB dữ liệu.

* Các thiết bị ngoại vi khác
* Card âm thanh
* Card mạng Lan
* Modem Dial-up
* Ổ đĩa F
* loppy Disk
* Ổ đĩa ZIP
* Đầu đọc thẻ nhớ

Linux là một hệ điều hành (OS) mã nguồn mở. Là phần mềm quản lý phần cứng và các tài nguyên khác của hệ thống như CPU, bộ nhớ và ổ cứng. Hệ điều này năm giữa ứng dụng và phần cứng tạo sự kết nối giữa tất cả các phần mềm và tài nguyên vật lý phục vụ cho việc thực hiện công việc.

## **3. Yêu cầu quản lý thiết bị**

Chức năng của các thiết bị ngoại vi là đảm nhiệm việc truyền thông tin qua lại giữ các bộ phận của hệ thống. Do đó, yêu cầu của hệ điều hành là tìm phương pháp tổ chức và truy nhập thông tin trên các thiết bị.

Ngoài các thiết bị chuẩn có tính chất bắt buộc như màn hình, bàn phím, máy in,… thì các hệ thống máy tính phải có khả năng kết nối với số lượng tùy ý các thiết bị ngoại vi bổ sung. Các thiết bị này có thể khác nhau về bản chất và nguyên lý hoạt động, vì vậy hệ điều hành cần phải tìm cách quản lý, điều khiển và khai thác các thiết bị một cách có hiệu quả nhất.

CPU không làm việc trực tiếp với các thiết bị ngoại vi, do đó cần phải tổ chức các thiết bị sao cho CPU không phụ thuộc vào sự biến động của các thiết bị.

## **4. Nguyên tắc tổ chức và quản lý thiết bị**

Nguyên tắc cơ bản để tổ chức và quản lý thiết bị dựa trên cơ sở: CPU chỉ điều khiển các thao tác vào/ra chứ không trực tiếp thực hiện các thao tác này. Để đảm bảo được nguyên tắc này, các thiết bị không gắn trực tiếp với CPU mà gắn với các thiết bị đặc biệt – thiết bị quản lý (Control Device). Một thiết bị có thể kết nói với nhiều thiết bị vào/ra.

Thiết bị quản lý đóng vai trò như một máy tính chuyên dụng có nhiệm vụ điều khiển các thiết bị kết nối với nó và gọi kênh vào/ra. Mỗi kênh vào/ra có ngôn ngữ và hệ lệnh riêng. Chúng hoạt động độc lập với nhau, độc lập với CPU và độc lập với các thành phần khác trong hệ thống.

Ví dụ: Để chuyển thông tin từ bộ nhớ trong ra ngoài và ngược lại, kênh phải truy nhập trực tiếp bộ nhớ theo một cơ chế đặc biệt, song song và độc lập với CPU. Cơ chế này được gọi là DMA ( Direct Memory Access)

Một hệ thống máy tính có thể có nhiều kênh vào/ra, mỗi kênh vào/ra lại có thể có những kênh con của mình. Để điều khiển hoạt động của các kênh, cần có các chương trình điều khiển riêng gọi là chương trình điều khiển kênh.

Để hệ thống làm việc được với các kênh thì CPU phải hiểu được ngôn ngữ kênh. Ngôn ngữ kênh được nạp vào hệ thống khi nạp hệ điều hành hoặc ngay cả khi hệ điều hành đang hoạt động ( ngôn ngữ kênh thực chất là các trình điều khiển kênh).

### **4.1 Linux điều khiển các thiết bị phần cứng**

Linux cho phép chúng ta có quyền điều khiển phần cứng của hệ thống( tương tự như Control Panel của Windows). Tuy nhiên việc truy cập và điều khiển các thiết bị phần cứng không dễ như trong Windows, mặc dù nó tỏ ra khá cơ động và không bảo trì nhiều một khi nó đã xác lập. Trong một số trường hợp phải biên dịch lại nhân nếu muốn bổ sung phần cứng mới vào hệ thống. Các CD-ROM, Sound Card bắt buộc phải làm như vậy. Nhưng Modem, thiết bị chuột hoặc ổ đĩa cứng thì có thể không cần thiết. Mỗi thiết bị ngoại vi muốn được dùng thì cần phải có các trình điều khiển thiết bị đi kèm. Phần mềm dùng để điều khiển thiết bị được gọi là *Device Driver*. Trong Linux, các *Device Driver* của nhân Linux thực chất là thư viện dùng chung, thường trú trong bộ nhớ hoặc các trình điều khiển phần cứng ở mức thấp. Tất cả các thiết bị phần cứng đều được xem như là các tập tin thông thường, chúng có thể được mở, đóng, đọc, ghi bằng cách sử dụng các lời gọi hệ thống giống như các lời gọi hệ thống quản lý tập tin. Mỗi thiết bị được biểu diễn như một thiết bị tệp đặc biệt ( Device Special File).

Ví dụ: Thiết bị đĩa IDE thứ nhất trong hệ thống được biểu diễn bởi */dev/had*. Đối với các thiết bị khối (disk) và thiết bị ký tự (character device) thì các thiết bị tệp đặc biệt của chúng được khởi tạo bởi lệnh *mknod* và chúng mô tả thiết bị bằng cách sử dụng các số hiệu chính (major device number) và số hiệu nhỏ (minor device number). Thiết bị mạng còn được biểu diễn như là một tập tin thiết bị đặc biệt, nhưng chúng được Linux khởi tạo khi khởi sinh bộ điều khiển mạng trong hệ thống.

Các thiết bị được điều khiển bởi một bộ điều khiển chung (driver) sẽ được gán một số (định danh) chung gọi là số hiệu chính. Các thiết bị đó được phân biệt thông qua một số gọi là số hiệu nhỏ. Ví dụ: Mỗi phân vùng (partition) trên một đĩa cứng có một số hiệu nhỏ của mình, vậy */dev/hda2* ( partition thứ hai trên đĩa cứng IDE thứ nhất) có số hiệu chính cho cả thiết bị là 3 và số hiệu nhỏ để phân biệt là 2. Linux ánh xạ một tập tin thiết bị lên một driver thiết bị nhờ sử dụng số hiệu chính của thiết bị và số hiệu của bảng hệ thống.

### **4.2 Linux quản lý các thiết bị lưu trữ**

Linux có cách điều khiển các thiết bị rất khác biệt so với các hệ điều hành khác. Sẽ không có các tên thiết bị lưu trữ vật lý như ổ A hay ổ C,… mà lúc đó, các thiết bị lưu trữ này sẽ trở thành một phần của hệ thống tập tin cục bộ thông qua một thao tác được gọi là “ kết gắn – mounting ”. Khi đang sử dụng thiết bị lưu trữ đó, muốn tháo bỏ phải “ tháo bỏ kết gắn – unmount ” thiết bị.

Một số lệnh quản lý:

* Lệnh mount: ghép nối thiết bị vào cây thư mục
* Lệnh umount: gỡ bỏ kết nối
* Lệnh du: xem dung lượng đĩa đã dùng
* Lệnh df: kiểm tra dung lượng đĩa trống

…..

**5. Bộ điều khiển DMA**

Linux sử dụng cơ chế DMA để quản lý các kênh DMA (mỗi kênh có một vector).

**5.1 Khái niệm DMA**

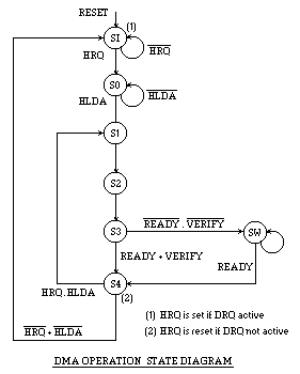
Kỹ thuật vào ra DMA(direct memory access) là phương pháp truy cập trực tiếp tới bộ nhớ hoặc I/O mà không có sự tham gia của CPU .Phương pháp này trao đổi dữ liệu giữa bộ nhớ và thiết bị ngoại vi với tốc độ cao và chỉ bị hạn chế bởi tốc độ của bộ nhớ hoặc của bộ điều khiển DMA .Tốc độ truyền DMA có thể đạt tới 10 - 12 Kbyte với các bộ nhớ RAM có tốc độ cao . DMA được ứng dụng trong nhiều mục đích nhưng thông thường nó được dùng trong quá trình "refresh" DRAM , màn hình ,đọc ghi đĩa ,truyền dữ liệu giữa các vùng nhớ với tốc độ cao .

**5.2 Hoạt động DMA cơ bản**

Hai tín hiệu để yêu cầu và xác nhận trong hệ thống là HOLD được sử dụng để yêu cầu DMA và HLDA là đầu ra xác nhận DMA .Khi tín hiệu HOLD hoạt động ( = 1) DMA được yêu cầu .Bộ VXL trả lời bằng cách kích hoạt tín hiệu HLDA ,xác nhận yêu cầu đồng thời thả nổi các công việc hiện thời cùng các bus dữ liệu và địa chỉ ,điều khiển được đặt ở trạng thái trở kháng cao . Trạng thái này cho phép các thiết bị I/O bên ngoài hoặc các bộ VXL khác nắm quyền điều khiển bus hệ thống để truy cập trực tiếp bộ nhớ . Tín hiệu HOLD có mức ưu tiên cao hơn INTR( interrupt request ) hoặc đầu vào NMI (ngắt không che được ) và chỉ sau RESET .Tín hiệu HOLD luôn có hiệu lực tại bất kỳ thời điểm nào trong suốt quá trình thực hiện các lệnh khác của VXL . Chú ý rằng từ lúc tín hiệu HOLD thay đổi cho đến khi tín hiệu HLDA thay đổi đã trải qua một số chu kỳ clock.

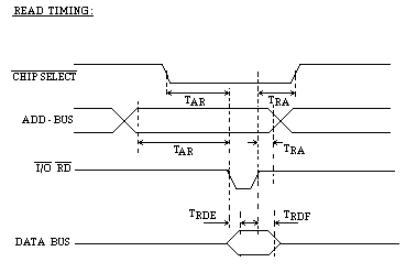
DMA thường được thực hiện giữa thiết bị I/O và bộ nhớ .Quá trình đọc DMA là quá trình đưa dữ liệu từ bộ nhớ ra thiết bị I/O và ngược lại ,quá trình ghi DMA là quá trình đưa dữ liệu từ I/O tới bộ nhớ . Trong cả hai chu trình này thiết bị I/O và bộ nhớ được điều khiển đồng thời dẫn đến cần có các tín hiệu điều khiển khác nhau .Ðể điều khiển quá trình đọc DMA ta cần hai tín hiệu hoạt động MEMR( đọc bộ nhớ ) và IOW (ghi I/O ) .Ðể điều khiển quá trình ghi ta có hai tín hiệu MEMW ( ghi bộ bộ nhớ ) và IOR (đọc I/O ).Bộ điều khiển DMA cung cấp địa chỉ bộ nhớ và tín hiệu chọn thiết bị I/O cho 8088 trong suốt quá trình DMA .Do tốc độ truyền DMA phụ thuộc vào tốc độ của bộ nhớ và tốc độ của bộ điều khiển DMA nên trong trường hợp tốc độ của bộ điều khiển DMA nhỏ hơn so với bộ nhớ thì bộ điều khiển DMA sẽ làm giảm tốc độ chung của hệ thống

Hình vẽ sau minh hoạ quá trình hoạt động DMA cơ bản cùng đồ thị thời gian đọc / ghi DMA :

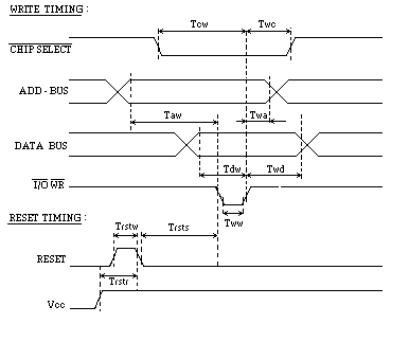


*Hình 2.1*

Ðồ thị thời gian đọc / ghi DMA.



*Hình 2.2*



*Hình 2.3*

**6. Các kỹ thuật áp dụng trong quản lý thiết bị**

**6.1. Kỹ thuật vùng đệm**

### **6.1.1 Khái niệm và mục đích của vùng đệm**

* Vùng đệm(buffer) là một cùng nhớ trung gian dùng làm nơi lưu trữ thông tin tạm thời trong các thao tác vào/ra.

Để thực hiện một thao tác vào/ra, hệ thống cần phải thực hiện thao tác các bước sau:

1. Kích hoạt thiết bị.
2. Chờ thiết bị đạt trạng thái thích hợp.
3. Chờ thao tác vào/ra được thực hiện.

Việc chờ đợi các thiết bị đạt trạng thái thích hợp chiếm một thời gian khá lớn trong tổng thời gian thực hiện các thao tác vào/ra. Vì vậy, để đảm bảo tốc độ hoạt động chung của toàn hệ thống, thao tác vào /ra cần phải sử dụng vùng đệm nhằm mục đích:

1. Giảm số lượng các thao tác vào/ra vật lý.
2. Cho phép thực hiện song song các thao tác vào/ra với các xử lý thông tin khác nhau.
3. Cho phép thực hiện trước các phép nhập dữ liệu.

### **6.1.2.phân loại vùng đệm**

Thường chia làm 3 loại:

* Vùng đệm trung chuyển
* Vùng đệm xử lý
* Vùng đệm vòng tròn
* ***Vùng đệm trung chuyển***

Trong vùng đệm này hệ thống tổ chức hai vùng nhớ riêng biệt:

* Vùng nhớ vào: Chỉ dùng để nhập thông tin
* Vùng nhớ ra: Ghi thông tin

Trong chương trình ứng dụng, ngay sau khi khi mở file, thông tin sẽ được chuyển đến vào vùng nhớ. Khi gặp lệnh đọc(read) thông tin sẽ được chuyển tới các địa chỉ tương ứng nêu trong chương trình ứng dụng, mỗi giá trị sẽ được lưu trong hai nơi bộ nhớ. Sau khi giá trị cuối cùng của vùng đệm được lấy ra vùng đệm trở nên rỗng và hệ thống tổ chức nhập thông tin mới vào thời điểm sớm nhất có thể. Để giảm thời gian chờ đợi, hệ thống có thể tổ chức nhiều vùng đệm vào, khi hết thông tin ở một vùng đệm hệ thống sẽ chuyển sang vùng đệm kế tiếp.

Đối với vùng đệm ra, thông tin cũng được xử lý tương tự nhưng theo trình tự ngược lại. Lệnh ghi (write) không đọc trực tiếp thông tin mà thiết bị đưa ra. Khi một vùng đệm ra đầy hệ thống sẽ chuyển sang làm việc với vùng đệm kế tiếp đồng thời tổ chức đưa thông tin từ vùng đệm trước ra thiết bị .

**ƯU ĐIỂM**:

* Có hệ số song song cao,phổ dụng
* Cách thức tổ chức đơn giản

**NHƯỢC ĐIỂM** :

* Tốn bộ nhớ
* Kéo dài thời gian trao đổi thông tin ở bộ nhớ trong
* **Vùng đệm xử lý**

Ở đây các thông tin vào ra cùng được xử lí trong một vùng bộ nhớ. Trong trường hợp này lệnh đọc (read) xác định địa chỉ thông tin.

**ƯU ĐIỂM**:

* Tiết kiệm không gian nhớ
* Rút gọn thời gian trao đổi thông tin ở bộ nhớ trong:

**NHƯỢC ĐIỂM**:

* Tốc độ giải phóng vùng đệm chậm-> hệ số song song thấp hơn vùng đệm trung chuyển.
* Không phải thao tác trao đổi vào ra nào cũng có thể sử dụng được vùng đệm này
* Phương pháp tổ chức vùng đệm phức tạp
* **Vùng đệm vòng tròn**

Trong cách tổ chức này hệ thống làm việc với 3 vùng đệm:

* Một vùng đệm để đưa thông tin vào
* Một vùng đệm đưa thông tin ra.
* Một vùng đệm xử lý.

Sau một khoảng thời gian nhất định thì các chức năng của các vùng đệm trao đổi cho nhau vòng tròn tức là vùng đệm vào-> vùng xử lý->vùng đệm ra ->vùng đệm vào.

Loại vùng đệm này có thẻ gắn vào từng file cụ thể hoặc gắn vào toàn hệ thống.

Chế độ gắn file: Phương pháp tổ chức này đặc biệt thích hợp khi lỗi file có một kích thước bản ghi vật lí riêng vùng đệm được xây dựng khi nạp hệ thống và chưa gắn với một file cụ thể nào khi mở file một hoặc một số vùng đệm được gắn vào file và phục vụ cho sự truy nhập file đó.

Khi đóng file vùng đệm không bị xóa mà được trả về cho hệ thống như một tài nguyên chung.

**ƯU ĐIỂM:**

* Vùng đệm này sẽ đạt hiệu quả cao khi thời gian xử lý tương đương với thời gian vào
* Phương pháp tổ chức này tránh được việc phải thực hiện các thủ tục tạo vùng đệm nhiều lần .

**NHƯỢC ĐIỂM**:

* Có những thời điểm vùng đệm không được sử dụng hết gây lãng phí bộ nhớ
* Vùng đệm có thể trở thành tài nguyên găng khi nhiều file được mở đồng thời để giảm khả năng xảy ra cạnh tranh vùng đệm chúng ta có thể tăng số lượng vùng đệm ngay từ khi nạp hệ thống nhưng sẽ chiếm dụng nhiều bộ nhớ và làm tăng thời gian dịch vụ của hệ thống đặc biệt là dàn thông tin của vùng đệm.

**6.2. Kỹ thuật kết khối**

Là ghép nhiều bản ghi logic thành một bản ghi vật lý và việc trao đổi thông tin giữa các bộ phận.

Thường tồn tại các tổ chức kết khối như sau:

* Mỗi bản ghi vật lý chứa một số nguyên lần bản ghi logic và giá trị này là như nhau với mọi bản ghi vật lý(áp dụng khi cần phải lưu trữ hoặc sao chép các file có kích thước lớn nhưng không muốn sử dụng công cụ backup dữ liệu).
* Mỗi bản ghi vật lý chưa một số nguyên lần bản ghi logic nhưng số lương các bản ghi logic không giống nhau với những bản ghi quản lí khác nhau
* Bản ghi vật lý có độ dài cố định, không phụ thuộc vào các bản ghi logic. Vì vậy bản ghi vật lý không nhất thiết phải chứa một số nguyên lần các bản ghi logic.

Bản ghi vật lý chưa một phần bản ghi logic và vì vậy phải kết hợp nhiều bản ghi vật lý mới được một bản ghi logic.

Phương pháp kết khối được chọn tùy thuộc vào vấn đề cần giải quyết và phương thức hoạt động của thiết bị. còn được sử dụng như một biện pháp hạn chế việc truy nhập bất hợp lệ. Nếu không nêu đúng hệ số kết khối thì hệ thống sẽ không tiếp tục thực hiện các phép truy nhập thông tin sẽ bị giải mã sai lệnh vì hệ số kết khối đã nêu không hợp lý.

**ƯU ĐIỂM** **:** Giảm đáng kể số lần truy nhập vật lí là một ưu điểm lớn nhất của kĩ thuật này.

**NHƯỢC ĐIỂM:** Sẽ kéo theo chi phí bổ sung cần phải có bộ nhớ lưu trữ các chương trình phục vụ kết khối và mở khối, tốn thời gian xử lí bản ghi.

**6.3.Xử lý lỗi**

Bất kì một thành phần nào của hệ thống cũng có thể thực hiện công việc một cách không chuẩn. Tuy nhiên không có bộ phận nào lại bộc lộ nhiều sai sót trong hoạt động của các thiết bị vào/ra vì các thiết bị này luôn chịu ảnh hưởng của yếu tố môi trường và có nhiều chi tiết bị hao mòn trong quá trình sử dụng như: các bộ phận chuyển động bo mòn , độ nhiễm từ trên các đĩa kém.

Phương pháp chủ yếu thường áp dụng trong chống lỗi vào ra là giao trách nhiệm phát hiện lỗi cho hệ thống không phải cho người sử dụng. Vì nguyên nhân phát sinh ra lỗi là rất nhiều nên hệ thống phải thực hiện linh hoạt các phép kiểm tra thiết bị( cả phần cứng phần mềm).Các công đoạn kiểm tra được chú ý ngay từ công đoạn thiết kế chế tạo.

Để đảm bảo độ chính xác của thông tin lưu trữ, nhiều thiết bị tổ chức được lại nhiều thông tin sau khi ghi và so sánh kết quả của thông tin gốc hoặc so sánh kiểm tra tính được khi đọc với tổng kiểm tra tính được với thông tin gốc. Phương pháp này thường được áp dụng với các thiết bị có tốc độ nhanh như đĩa từ. việc so sánh và kiểm tra thông thường do các thiết bị điều khiển vào ra đảm nhận, sau đó mới thông báo lỗi cho hệ thống và chịu trách nhiệm thực hiện các tác động tương ứng.

Với mục đích tránh mọi sai sót (như cố gắng đọc đĩa từ khi chưa sẵn sàng). Trước và sau phép trao đổi vào ra hệ thống có những thao tác kiểm tra với đối kênh vào ra và phân tích kết quả xem có đủ điều kiện truy nhập thiết bị hay chưa.

Việc áp dụng các mã sửa sai giúp hệ thống khắc phục các lỗi dữ liệu thường gặp đặc biệt là thông tin lưu trữ dài hạn. Chính vì vậy tuy tốn nhiều thời gian và chi phí xây dựng nhưng mã sửa sai vẫn được áp dụng rộng rãi khi cần phải lưu trữ thông tin dài hạn.

**CHÚ Ý:**

* Hệ thống chỉ báo lỗi khi không tự khắc phục được và sẽ nêu phương án cho người sử dụng phương án tự giải quyết công việc.
* Việc kiểm tra và xử lý lỗi là một quá trình phức tạp liên quan chặt chẽ với đặc trưng của từng thiết bị cụ thể. Tuy nhiên mỗi thiết bị đều cung cấp một mã trở về (return code) cho hệ thống để các chương trình xử lý kết quả phân tích đánh giá.
* Để công việc phân tích đánh giá không chiếm dụng giờ CPU, ảnh hưởng tới tốc độ hoạt động của hệ thống thì thường các thiết bị có xu hướng cục bộ hóa sai sót (phân tích,xử lý,đánh giá… ngay tại thiết bị)

## **7. Cách truy xuất đĩa**

Cũng tương tự như Window, trong Linux cũng có khái niệm đường dẫn (path). Tuy nhiên, có 2 điểm cần lưu ý:

* Thứ nhất, sử dụng ký tự sổ trái (/) làm ký tự phân cách thư mục và tập
* tin.
* Thứ hai, không sử dụng ký tự ổ đĩa, mà dùng ký tự / ở đầu đường dẫn (thư mục gốc).

Ví dụ:

* /usr/local/dev
* /dev/hda

Khi khởi động hệ điều hành, Linux chỉ kết gắn cho phân vùng chính (nơi chứa nhân Linux) bằng ký tự “/” (thư mục gốc).

Các thông tin của phân vùng khác được Linux đặt trong thư mục /dev của phân vùng chính.

Như vậy mặc dù tất cả các file trong Linux đều được đặt trong cùng một cây thư mục, song chúng có thể được lưu trữ trên các bộ nhớ ngoài khác nhau như đĩa cứng hay CD-ROM. Mỗi thiết bị nhớ cũng có thể có hệ thống file (file system) khác nhau như: FAT, NTFS, ext2, …Để gắn một hệ thống file trên thiết bị lưu trữ vào cây thư mục chính ta dùng lệnh mount.

## **8. Các lệnh quản lý thiết bị ngoại vi**

### **8.1 Quản lý thiết bị lưu trữ trong Linux**

Linux có cách điều khiển các thiết bị rất khác biệt so với hệ điều hành khác. Sẽ không có các tên thiết bị lưu trữ vật lí như ổ A hay ổ C…, mà lúc đó, các thiết bị lưu trữ này sẽ trở thành 1 phần của hệ thống file cục bộ thông qua 1 thao tác được gọi là “*kết gắn- mount*”. Khi đang sử dụng thiết bị lưu trữ đó muốn tháo bỏ thì phải “*tháo bỏ kết gắn- unmount*” thiết bị.

### **8.1.1. Lệnh *mount* và *unmount***

* Lệnh *mount*:

Để sử dụng bất kì một thiết bị lưu trữ vật lí nào trên Linux đều cần đến lệnh mount. Điểm gắn kết là thư mục /mnt

Cú pháp lệnh:

*mount [tùy chọn] <file thiết bị> <thư mục>*

Lệnh này thông báo cho nhân hệ thống thực hiện việc kết gắn hệ thống file có trên file thiết bị vào thư mục (điểm kết gắn) là thư mục

Một số tùy chọn của lệnh mount:

* + t <kiểu>: xác định kiểu của thiết bị, xác định kiểu hệ thống file. Các kiểu hệ thống file hiện thời có trong file Linux/fs/filesystems.c
  + h: đưa ra trang trợ giúp
  + a: gắn kết tất cả các file hệ thống (thuộc kiểu được đưa ra) có trong file fstab
  + r: kết gắn hệ thống file chỉ có quyền đọc
  + w: kết gắn hệ thống file có quyền đọc, ghi.

…

Ví dụ:

* Cú pháp sử dụng đĩa mềm:

# mount –t msdos /dev/fd0 /mnt/floppy

* Cú pháp sử dụng đĩa CD:

#mount /dev/cdrom /mnt/cdrom

Thông thường, chỉ có người dùng root mới có quyền gắn kết các thiết bị, để mọi người khác có thể kết gắn đĩa mềm hoặc CD ROM chẳng hạn, hãy thực hiện như sau:

* Với tư cách người dùng root, hãy gõ các lệnh cấp cho người dùng các quyền truy nhập tới 2 thư mục là điểm kết gắn với 2 thiết bị đĩa mềm và CD ROM

*# chmod a+rữ /mnt/floppy; /mnt/cdrom*

* Cấp cho mọi người dùng quyền đọc và ghi đối với hai thư mục lưu trũ thiết bị hệ thống

*# chmod a+rw /dev/fd0; /dev/c*ú*drom*

Lưu ý: việc cho phép mọi người dùng có thể sử dụng lệnh mount được đối với thiết bị địa của mình là điều bất lợi vì nó có liên quan đến vấn đề bảo mật.

* Lệnh *unmount*

Lệnh unmount cho phép tháo bỏ gắn kết của một hệ thống file trên hệ thống file chính.

Cú pháp lệnh:

Unmount [tùy- chọn] <thiết-bị>

Cần chú ý rằng, không thể tháo bỏ gắn kết của một hệ thống file khi nó “bận”- tức là khi có một tiến trình đang hoạt động truy cập đến các file trên hệ thống file đó.

Một số tùy chọn lệnh:

* + n: loại bỏ các gắn kết mà không ghi vào thư mục /etc/mtab.
  + v: hiện các chế độ liên quan.
  + r: trong trường hợp loại bỏ gắn kết bị lỗi, tùy chọn này sẽ giúp tạo lại gắn kết với chế độ chỉ đọc.
  + a: tất cả các file hệ thống được hiển thị trong /etc/mtab đã được loại bỏ các gắn kết.

…

Ví dụ, khi không dùng đến đĩa mềm nữa, có thể dùng lệnh sau:

#unmount /mnt/f0

Tất cả các hệ thống file cần phải được mount trước khi truy nhập và phải được unmount khi đóng hệ thống.

Tuy nhiên Linux sẽ tự động mount một số thiết bị cho bạn khi khởi động và các thiết bị này cũng sẽ tự động được unmount khi đóng hệ thống

### **8.1.2. Định dạng đĩa trong Linux**

* Ổ đĩa cứng:

Ổ đĩa cứng phải được phân hoạch trước khi có thể định dạng và sử dụng nó. Tương tự như DOS, trong Linux có fdisk. Trong Linux có thể tạo ra các kiểu phân hoạch khác nhau, một phân hoạch được gắn với một chỉ số ( index: ID) để thông báo cho hệ điều hành biết kiểu phân hoạch của nó.

Dùng các lệnh sau:

# su

Passwd

# fdisk /dev/had

Command (m for help)

Lệnh trên báo cho fdisk biết sẽ làm việc với loại ổ đĩa nào. Nếu dùng đĩa kiểu SCSI, thì hda sẽ được thay bằng sda. Để xem các lệnh của fdisk, hãy gõ “m”

#fdisk /dev/had

Command (m for help): m

Ví dụ:

Command (m for help): p

Disk /dev/had: 64 heads, 63 sectors, 847 cylinders

Units= cylinders of 4032\* 512 bytes

Device Boot Start End Blocks Id System

/dev/hda1 \* 1 817 1647040+ 83 Linux

/dev/hda2 818 847 60480 82 Linux swap

Command (m for help) : q

#

Các cột trên đây có ý nghĩa như sau:

* + Cột Device có mục thiết bị dành cho phân vùng trong /dev, chẳng hạn /dev/hda1.
  + Cột Boot chỉ xem phân vùng đó có khả năng khởi động hay không, nếu có khả năng khởi động thì có dấu \*.
  + Cột Start, End chỉ ra số trụ (cylinder) đầu và cuối của phân vùng.
  + Cột Blocks chỉ ra dung lượng của phân vùng (là số lượng tính theo khối 1KB).
  + Cột Id và System: chỉ số Id và ý nghĩa của nó, ví dụ ID=82 có ý nghĩa là Linux swap.
* Định dạng mức thấp một đĩa mềm (lệnh fdformat)

Cú pháp lệnh định dạng mức thấp một đĩa mềm:

Fdformat [-n] thiết-bị

Lệnh *fdformat* thực hiện việc định dạng một đĩa mềm ở mức thấp. Tham số thiết-bị thường là một trong các loại sau ( chỉ định dạng với đĩa mềm, với chỉ số thiết bị là major=2- đây là chỉ số xác định kiểu của thiết bị, và minor- chỉ số xác định số của thiết bị trong trường hợp có nhiều thiết bị cùng loại):

/dev/fd0d360 (minor=4)

/dev/fd0h1200 (minor=8)

/de/fd0D360 (minor=12)

…

Tham số -n cho phép bỏ qua cá kiểm tra được thực hiện sau khi định dạng đĩa.

Các đĩa mềm, /dev/fd0 và dev/fd1 sẽ gặp lỗi nếu dùng lệnh format khi kiểu định dạng không chuẩn.

### **8.1.3. Quản lí đĩa**

* Xem dung lượng đĩa đã sử dụng bằng lệnh du

Linux cho phép người dùng xem thông tin về dung lượng đĩa đã được sử dụng bằng lệnh du với cú pháp:

du [tùy-chọn] [file]…

Lệnh du liệt kê kích thước ( tính theo kilo bytes) của mỗi file thuộc vào hệ thống file có chứa file được chỉ trong lệnh.

Một số tùy chọn là:

-a: liệt kê kích thước của tất cả các file có trong hệ thống file lưu trữ file.

-b,--bytes: hiển thị kích thước theo byte.

-c,--total: hiển thị cả tổng dung lượng được sử dụng trong hệ thống file.

-D,--dereference-args: không tính kích thước các file được liên kết đến nếu chúng nằm trên các thư mục khác.

-h,--human-readable: hiển thị kích thước các file kèm theo đơn vị tính (ví dụ: 1K, 234M, 2G…)

-k,--kilobytes: hiển thị kích thước tính theo kilobytes.

-m,--megabytes: tính kích thước của thư mục con.

-S,--separate-dí: không hiển thị kích thước của thư mục con.

-s: đưa ra kích thước của hệ thống file có lưu trữ file.

-x,--one-file system: bỏ qua các thư mục trên các hệ thống file khác.

--help: hiển thị trang trợ giúp và thoát.

Cần lưu ý rằng, lệnh du không cho phép có nhiều có nhiều tùy chọn trên cùng một dòng lệnh.

Ví dụ: Lệnh sau cho biết kích cỡ của các file trong thư mục /usr/doc/test:

# du /usr/doc/test

28 ./TODO/1.0\_to\_1.5

24 ./TODO/lib++

16 ./TODO/unreleased

12 ./TODO/unstable

144 ./TODO

44 ./code

160 ./languages

56 ./licenses

Nhìn vào màn hình có thể biết được kích thước của file . ./TODO/1.0\_to\_1.5 là 28KB, file ./TODO/lib++ là 24KB,… và kích thước của thư mục hiện thời là 532KB

* Kiểm tra dung lượng đĩa trống với lệnh *df:*

Cú pháp lệnh:

Df [tùy-chọn] [file]…

Lệnh này hiển thị dung lượng đĩa còn trống trên hệ thống file chứa file. Nếu không có tham số file, thì lệnh này hiển thị dung lượng đĩa còn trống trên tất cả các hệ thống được kết nối.

Một số tùy chọn là:

* -a,--all: bao gồm cả các file hệ thống có dung lượng là 0 block.
* --block-size= cỡ: thiết lập lại độ lớn của khối là cỡ byte.
* -k,--kilobytes: hiển thị dung lượng tính theo kilobytes.
* -l,--local: giới hạn danh sách các file cục bộ trong hệ thống.
* -m,--megabytes: hiển thị dung lượng tính theo megabytes.
* -t,--type=kiểu: giới hạn danh sách các file của hệ thống thuộc kiểu
* -T,--print-type: hiển thị các kiểu file hệ thống.
* --help: đưa ra trang trợ giúp và thoát.

Để chỉ ra được dung lượng đĩa còn trống trong Linux không phải là điều dễ làm. Người dùng có thể sử dụng lệnh df để làm được điều này, tuy nhiên kết quả của lệnh này chỉ cho biết dung lượng đĩa đã sử dụng và dung lượng đĩa còn trống của từng hệ thống file.

### **8.2. Cổng nối tiếp và modem**

Các cổng nối tiếp, ví dụ COM1 trong DOS hoặc Windows, được Linux xem là /dev/ttyS0 và /dev/ttyS dành cho dữ liệu đến và /dev/cua dành cho dữ liệu đi.

Để xác lập trong nhân kiểu Modem hiện có, sử dụng chương trình setserial.

Ví dụ: #setserial b/dev/cua0 irq 15 autoconfig

Lệnh trên sẽ đặt lại ngắt dùng cho dữ liệu của COM1 là IRQ15. Sau khi đã xác lập modem, có thể sử dụng một số tiện ích để quay số như minicom,seyon…

### **8.3. Cổng song song và máy in**

Để sử dụng máy in, phải đảm bảo có phần hỗ trợ máy in trong nhân và đảm bảo IRQ máy in bình thường chưa có thiết bị khác sử dụng.

Nếu máy in nằm trên cổng song song đầu tiên, với tư cách siêu người dùng, có thể nhập lệnh dưới đây

### **8.3.1. Khởi tạo và thiết lập máy in trong lpd**

Hầu hết các máy in trên hệ thống Linux đều được điều khiển bởi một chương trình chạy ngầm được gọi là daemon lpd: chương trình này được khởi động cùng hệ thống. Trong suốt quá trình khởi động, daemon lpd sẽ đọc file /etc/printcap để nhận dạng các phần được áp dụng cho bất kỳ máy in nào gắn vào hệ thống.

Ví dụ:

#LOCAL djet500

lp|dj|deskjet:\

:sd=/var/spool/lpd/dj:\

:mx#0:\

:lp=/dev/lp0\

:sh:

### **8.3.2. Các lệnh in ấn cơ bản**

In một file với lệnh lpr

Với tư cách người sử dụng root, có thể dùng lệnh cat thực hiện được phương pháp đó:

# cat thesis.txt >/dev/lp

Linux cho phép sử dụng các lệnh lpr, lprm, lpq để in ấn văn bản mà không đòi hỏi vấn đề bảo mật quá chặt chẽ

Ví dụ: # lpr -160 thesis.txt

### **8.4. Sound card**

Hầu hết sound card chính đều được Linux hỗ trợ. Thường thì phần hỗ trợ được cài đặt trong nhân. Linux cho phép dùng lệnh cat để làm việc với các dạng file audio

Dạng file đặc biệt của nó: /dev/audio, /dev/mide, /dev/dsp (Digital signal processor)

Có thể thu một file nhạc và nghe một file nhạc theo định dạng .au một cách trực tiếp, mà không cần phải thông qua một chương trình nghe nhạc nào khác, ví dụ:

#cat /dev/audio > file.au

Để phát lại dữ liệu phát lại dữ liệu có thể dùng:

# cat file.au >/dev/audio

# **Chương 3: Kết luận**

Qua quá trình Nghiên cứu tìm hiểu về quản lý thiết bị ngoại vi trong hệ điều hành Linux. Chúng ta biết được các loại thiết bị ngoại vi, các yêu cầu, nguyên tắc tổ chức và quản lý thiết bị ngoại vi, các lệnh để quản lý thiết bị, tầm quan trọng của việc quản lý thiết bị ngoại vi trong hệ điều hành Linux.