**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

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

****

**BÁO CÁO BTL THUỘC HỌC PHẦN:**

**NGUYÊN LÍ HỆ ĐIỀU HÀNH**

**ĐỀ TÀI:**

**NGUYÊN CỨU TÌM HIỂU VỀ QUẢN LÝ THIẾT BỊ NGOẠI VI TRONG HỆ ĐIỀU HÀNH LINUX**

|  |  |  |
| --- | --- | --- |
| GVHD: | ThS. Nguyễn Thanh Hải | |
| Nhóm - Lớp: | 8 - 20222IT6025003 |  |
| Thành viên: | Trần Xuân Đức | 2021604681 |
|  | Trần Đức Hạnh | 2021605015 |
|  | Vũ Thị Huế | 2021601676 |
|  | Bùi Huy Nam | 2021603337 |
|  | Thân Ngọc Thiện | 2021602775 |

***Hà Nội, 2023***

**MỤC LỤC**

[LỜI CẢM ƠN 5](#_Toc138269792)

[LỜI NÓI ĐẦU 6](#_Toc138269793)

[CHƯƠNG I: TỔNG QUAN VỀ HĐH LINUX 7](#_Toc138269794)

[1.1 Linux là gì? 7](#_Toc138269795)

[1.2 Các phiên bản của hệ điều hành linux 8](#_Toc138269796)

[1.3 Ưu nhược điểm của Linux 9](#_Toc138269797)

[1.3.1 Ưu điểm: 9](#_Toc138269798)

[1.3.2 Nhược điểm: 10](#_Toc138269799)

[1.4 Dòng lệnh trên Linux 10](#_Toc138269800)

[1.5 So sánh HĐH Windows và Linux 11](#_Toc138269801)

[CHƯƠNG II: QUẢN LÝ THIẾT BỊ NGOẠI VI TRONG HĐH LINUX 12](#_Toc138269802)

[2.1. Tìm hiểu về thiết bị ngoại vi 12](#_Toc138269803)

[2.1.1 Thiết bị ngoại vi là gì? 12](#_Toc138269804)

[2.1.1.1 Khái niệm 12](#_Toc138269805)

[2.1.1.2 Phân loại 12](#_Toc138269806)

[2.1.1.3. Một số thiết bị ngoại vi phổ biến 13](#_Toc138269807)

[2.1.2 Nhận biết cổng và khe cắm trong máy tính 14](#_Toc138269808)

[2.1.2.1 Khái niệm cổng và chức năng chính 14](#_Toc138269809)

[2.1.2.2 Cổng nối tiếp và cổng song song 14](#_Toc138269810)

[2.1.2.3. Các cổng phổ biến. Công dụng của từng loại cổng, tốc độ truyền dữ liệu 14](#_Toc138269811)

[2.1.2.4. Khe cắm PCI, ISA 18](#_Toc138269812)

[2.2 Yêu cầu và nguyên tắc xử lý thiết bị ngoại vi 20](#_Toc138269813)

[2.2.1. Yêu cầu của quản lý thiết bị 20](#_Toc138269814)

[2.2.2 Nguyên tắc tổ chức và quản lý thiết bị 20](#_Toc138269815)

[2.2.3 Bộ điều khiển DMA 21](#_Toc138269816)

[2.2.3.1 Khái niệm DMA 21](#_Toc138269817)

[2.2.3.2 Hoạt động DMA cơ bản. 21](#_Toc138269818)

[CHƯƠNG 3: CÁC KỸ THUẬT TRONG QUẢN LÝ THIẾT BỊ NGOẠI VI TRONG LINUX. 23](#_Toc138269819)

[3.1 Trình quản lý thiết bị trong hệ điều hành Linux 23](#_Toc138269820)

[3.2. Tệp thiết bị (Device file) 24](#_Toc138269821)

[3.3 Cách thức hoạt động của thiết bị ngoại vi 25](#_Toc138269822)

[3.3.2. Character Device 27](#_Toc138269823)

[3.3.3. Network Device 27](#_Toc138269824)

[3.4. Các kĩ thuật trong quản lý thiết bị 28](#_Toc138269825)

[3.4.1. Kĩ thuật vùng đệm 28](#_Toc138269826)

[3.4.1.1. Khái niệm và mục đích vùng đệm 28](#_Toc138269827)

[3.4.1.2. Phân loại vùng đệm 28](#_Toc138269828)

[3.4.2. Kĩ thuật kết khối 31](#_Toc138269829)

[3.4.3. Xử lí lỗi 32](#_Toc138269830)

[3.4.4. SPOOL 33](#_Toc138269831)

[3.5. Quản lý Driver modules 35](#_Toc138269832)

[3.5. Quy ước đặt tên 37](#_Toc138269833)

[3.6 Cách truy xuất đĩa 38](#_Toc138269834)

[3.7 Các lệnh quản lý thiết bị ngoại vi. 38](#_Toc138269835)

[KẾT LUẬN 43](#_Toc138269836)

[TÀI LIỆU THAM KHẢO 44](#_Toc138269837)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Một số thiết bị ngoại vi phổ biến 13](#_Toc138110522)

[Hình 2.2: Cổng kết nối USB-Type A 15](#_Toc138110523)

[Hình 2.3: Cổng kết nối USB TYPE C 15](#_Toc138110524)

[Hình 2.4: Cổng giao thức Thunderbolt 3 16](#_Toc138110525)

[Hình 2.5: Cổng kết nối HDMI 16](#_Toc138110526)

[Hình 2.6: Cổng kết nối VGA 17](#_Toc138110527)

[Hình 2.7: Cổng kết nối LAN RJ-45 18](#_Toc138110528)

[Hình 2.8: Ke cắm PCI, ISA 19](#_Toc138110529)

[Hình 3.1: Minh họa cách linux làm việc với thiết bị 23](#_Toc138110530)

[Hình 3.2: Hiển thị thông tin về các tệp và thiết bị trong thư mục /dev 25](#_Toc138110531)

[Hình 3.3: Danh sách những mô-đun đang được tải 36](#_Toc138110532)

# **LỜI CẢM ƠN**

"Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Công Nghiệp Hà Nội đã đưa môn học **Nguyên lí hệ điều hành** vào trương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – ThS. Nguyễn Thanh Hải đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Nguyên lí hệ điều hành của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Từ những điều thầy truyền tải, chúng em đã có đủ kiến thức và vận dụng chúng vào bài tập lớn.

Bộ môn Nguyên lí hệ điều hành là môn học thú vị, vô cùng bổ ích. Tuy nhiên, có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhât định. Do đó, trong quá trình hoàn thành bài tập lớn, chúng em khó có thể tránh khỏi những thiếu sót. Chúng em rất mong muốn nhận được những góp ý đến từ thầy để bài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!”

**Nhóm sinh viên thực hiện đề tài**

# **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 hànhlâu 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 chi là một nhân 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 nhên, còn khá nhiều rắc rối và bất cập khiến HĐH miễn phí này chưa thể thay thế hoàn toàn Window là nó khá rắc rối khi cài đặt, cực hình với những dòng lệnh, không thể sử dụng tất cả những ứng đụng 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 được chia sẻ với các bạn những gì chúng tôi biết về cách quản lý thiết bị ngoại vi trong hệ điều hành mã nguồn mở này.

# **CHƯƠNG I: TỔNG QUAN VỀ HĐH LINUX**

## **1.1 Linux là gì?**

***Linux*** là một hệ điều hành máy tính được phát triển từ năm 1991, dựa trên hệ điều hành Unix và viết bằng ngôn ngữ C. Đây là hệ điều hành mã nguồn mở, 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[1-3].

**Linux** bao gồm cả các công nghệ cũ và mới. Nhìn từ góc độ kỹ thuật, Linux chỉ là một nhân 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, phần lớn người dùng đều coi “*Linux*” như một hệ thống hoàn chỉnh gồm nhân hệ điều hành kèm theo các trình ứng dụng khác: một môi trường làm việc và phát triển đầy đủ bao gồm trình dịch, các hệ soạn thảo, giao diện đồ họa, xử lý văn bản, … Cho tới phiên bản *Linux RedHat 6.1*, Ubuntu 10 trở đi, … Linux đã trở thành một hệ điều hành đầy đủ cho thương mại, giáo dục hoặc người dùng cá nhân.

- Cấu trúc của HĐH Linux:

* **Kernel:** Hay còn được gọi là phần Nhân, là phần quan trọng và được ví như trái tim của HĐH Linux. Phần kernel quan trọng nhất của máy tính có nhiệm vụ chứa các module, thư viện để quản lý và giao tiếp với phần cứng và các ứng dụng.
* **Shell:** Shell là một chương trình có chức năng thực thi các lệnh (command) từ người dùng hoặc từ các ứng dụng yêu cầu– tiện ích yêu cầu chuyển đến cho Kernel xử lý. Shell được coi là cầu nối để kết nối Kernel và Application, phiên dịch các tập lệnh từ Application gửi đến Kernel để thực thi.
* **Applications:** Là các ứng dụng và tiện ích mà người dùng cài đặt trên Server. Ví dụ: ftp, samba, Proxy,…

## **1.2 Các phiên bản của hệ điều hành linux**

- Như ta đã biết Linux là hệ điều hành mã nguồn mở. Về mặt nguyên tắc hệ điều hành cũng là một software, nhưng đây là một software đặc biệt được dùng để quản lý, điều phối các tài nguyên (resource) của hệ thống (bao gồm cả hardware và các software khác). Linux còn được gọi là Open Source Unix (OSU), Unix-like Kernel, hay Clone of the UNIX operating system [4-5].

- Linux có nguyên lý hoạt động tương tự hệ điều hành Unix (Unix-like). Mặc dù Linux không phải là Unix nhưng người ta vẫn xem Linux như là phiên bản Unix trên PC (PC version of Unix OS). Do là Unix-like, Linux có đầy đủ tất cả các đặc tính của Unix (fully functional). Ngoài ra nó còn hỗ trợ thêm một số tính năng mà trên Unix không có, như long file name (tên file có ký tự space “ ”). Hiện tại có nhiều hãng, nhiều tổ chức, nhiều nhóm khác nhau cùng phát triển Linux. Tất cả các phiên bản (release). Linux đều có chung phần kernel (phần nhân của hệ điều hành) và hầu hết các tính năng đặc trưng, tuy nhiên các tool (công cụ) và utility (tiện ích) có đôi chút dị biệt [6].

- Có rất nhiều các ứng dụng cho Linux, tuy nhiên hầu hết các ứng dụng cho Linux hiện có đều là các ứng dụng mang tính chuyên dụng. Các ứng dụng được viết trên Linux đều có thể hoạt động trên các hệ thống UNIX (có thể cần phải compile lại). Các release hiện nay gồm có [7]:

* **RedHat Linux (Fedora Core):** Là phiên bản khá phổ biến. Cung cấp khá nhiều tool và utility để hỗ trợ user (người sử dụng) từ các thao tác setup (cài đặt) đến config (cấu hình) hệ thống.
* **Mandrake Linux:** Một dòng khác thoát thai từ RedHat Linux, tương thích hoàn toàn với RedHat. Thường có nhiều phần mềm mới đang ở giai đoạn thử nghiệm.
* **Mandrake Linux**: Một dòng khác thoát thai từ RedHat Linux, tương thích hoàn toàn với RedHat. Thường có nhiều phần mềm mới đang ở giai đoạn thử nghiệm.
* **Slackware Linux:** Đây là một trong phiên bản Linux lâu đời. Hỗ trợ các dịch vụ mạng rất mạnh, tuy nhiên việc setup và config đòi hỏi user có kiến thức tốt về hệ điều hành này.
* **S.u.S.E. Linux:** Do hãng S.u.S.E (Đức) phát hành, khá phổ biến tại Âu châu, nhưng không được phổ biến tại các nước khác. Có các công cụ riêng để hỗ trợ setup và config tương đối dễ sử dụng.
* **Free BSD Linux:** Được phát triển bởi Đại Học Berkeley, đây không phải là phiên bản thương mại, do vậy ít được phổ biến. Có rất nhiều tiện ích dành cho việc phát triển hệ thống và lập trình. Hỗ trợ đầy đủ các shell trên Unix.
* **Corel Linux:** Phát triển bởi hãng Corel, dễ setup, có graphic interface (giao diện đồ họa) khá giống Windows NT kể các tool và utility. Tuy nhiên các config tool chưa hoạt động tốt.
* **Open Linux:** Do hãng Caldera phát triển, dễ cài đặt cũng như sử dụng. Giao diện KDE. Thích hợp cho người sử dụng tại gia đình.
* Và còn rất nhiều release khác như Turbo Linux, Linux PPC, Debian Linux, Infomagic Linux, Softlanding Linux System Release (SLS)
* V.V....

- Ngoài ra, hiện nay còn có một dòng Linux gọi là Live-CD Linux (chạy trực tiếp trên CD – không cần cài đặt) như Ubuntu, Knoppix, thích hợp với các beginner Linux.

- Các thông tin và tài nguyên (resource) của Linux có thể tìm thấy ở khắp nơi trên Internet và hầu hết đều free. Thêm vào đó có khá nhiều các trình ứng dụng cũng như tiện ích dành cho Linux dễ dàng được tìm thấy trên Internet.

## **1.3 Ưu nhược điểm của Linux [8]**

### **1.3.1 Ưu điểm:**

**- Hệ điều hành Linux miễn phí***:* Với Linux bạn không phải mất phí để mua bản quyền Linux để bắt đầu quá trình sử dụng. Mà bạn hoàn toàn sử dụng một cách miễn phí với tất cả các chức năng của hệ điều hành này. Ngoài ra, được hỗ trợ các ứng dụng văn phòng OpenOffice và LibreOffice.

**- Tính linh hoạt:** Khi sử dụng Linux, người dùng có thể linh hoạt trong việc chỉnh sửa hệ điều hành theo nhu cầu của mình. Là một ưu điểm rất hữu ích trong quá trình sử dụng của các lập trình viên.

**- Tính bảo mật cao:** Tính bảo mật của Linux là cực cao nên tất cả các phần mềm mã độc, virus,… đều không thể hoat động trên Linux. Vì thế bạn hãy yên tâm tải Linux và sử dụng một cách thoải mái.

**- Linux hỗ trợ cho máy cấu hình yếu:** Với Linux dù máy tính của bạn có cấu hình yếu nhưng Linux vẫn hỗ trợ cập nhật, nâng cấp và hỗ trợ liên tục và thường xuyên trong khi sử dụng. Hoạt động của Linux cũng vô cùng ổn định trên các máy tính yếu.

### **1.3.2 Nhược điểm:**

Một vài hạn chế của Linux như:

+ Số ứng dụng được trên Linux còn hạn chế.

+ Một số nhà sản xuất driver không phát triển và hỗ trợ cho Linux.

+ Khó tiếp cận và làm quen khi bạn đã quá quen thuộc với Windows thì khi chuyển sang Linux bạn cần một thời gian để có thể làm quen được hệ điều hành này.

## **1.4 Dòng lệnh trên Linux**

Dòng lệnh trên Linux là quyền truy cập trực tiếp của bạn vào máy tính. Đó là nơi bạn yêu cầu phần mềm thực hiện các hành động mà giao diện người dùng đồ họa (GUI) không thể thực hiện.

Các câu lệnh trong Linux có sẵn, mã nguồn mở hoặc độc quyền. Nhưng nó thường được liên kết với Linux. Bởi vì cả hai dòng lệnh và phần mềm nguồn mở. Cung cấp cho người dùng quyền truy cập không hạn chế vào máy tính của họ.

## **1.5 So sánh HĐH Windows và Linux**

|  |  |  |
| --- | --- | --- |
|  | **Linux** | **Windows** |
| *Cấu trúc file* | Cây dữ liệu | Thư mục |
| *Registry* | Không có | Có |
| *Trình quản lý gói* | Quản lý gói (Package Manager) | \*.exe |
| *Giao diện* | Giao diện thay đổi được | Gắn liền với hệ thống |
| *Tài khoản và quyển user* | Rugular, Administrator (root), service | Administrator, Standard, Child và Guest |
| *Một số thiết lập điều khiển khác* | Phù hợp cho công việc: lướt web, chat, … | Phù hợp với mọi nhu cầu. |

Bảng 1.1: So sánh hệ điều hành Window và Linux

# **CHƯƠNG II: QUẢN LÝ THIẾT BỊ NGOẠI VI TRONG HĐH LINUX**

## **2.1. Tìm hiểu về thiết bị ngoại vi**

### **2.1.1 Thiết bị ngoại vi là gì?**

#### **2.1.1.1 Khái niệm**

- 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 (IO) hoặc mở rộng khả năng lưu trữ (như một dạng bộ nhớ phụ).

- Các thiết bị ngoại vi:

+ Màn hình.

+ Ổ đĩa mềm.

+ Ổ đĩa quang (CD, DVD).

+ Ổ đĩa cứng.

+ USB, thẻ nhớ,...

+ Bàn phím.

+ Chuột.

+ Loa, tai nghe.

#### **2.1.1.2 Phân loại**

**Thiết bị nhập – input**: Bao gồm đa số các phần cứng, nó cho phép bạn nhập dữ liệu, chương trình, các lệnh và những phản hồi từ các bạn vào máy tính như chuột máy tính, bàn phím, ổ đĩa CD,….

**Thiết bị xuất – output**: Bao gồm tất cả các nhân cung cấp những chức năng truyền thông tin tới có quý khách hoặc thực hiện các công tác giải mã thông tin dữ liệu mà khách hàng với thể hiểu được như USB, máy in, màn hình, ổ cứng, máy fax, máy chiếu, loa,…

#### **2.1.1.3. Một số thiết bị ngoại vi phổ biến [9]**

A picture containing camera, electronic device

Description automatically generated

Hình 2.1: Một số thiết bị ngoại vi phổ biến

- **USB (Universal Serial Bus):** là một dạng bộ nhớ mở rộng của các vật dụng số cầm tay, sử dụng khoa học flash để ghi lại dữ liệu.

- **Máy in:** là vật dụng sử dụng để xuất văn bản và hình ảnh ra thiết bị lưu trữ vật lý như giấy, tấm phim.

- **Bàn phím:** là thiết bị có những kí tự được khắc hoặc in trên phím, mỗi lần ấn phím sẽ có 1 ký hiệu hiện ra.

- **Chuột:** là vật dụng ngoại vi tiêu dùng để điều khiển và khiến việc trực tiếp mang máy tính qua con trỏ chuột trên màn hình hiển thị của máy tính.

- **Loa:** là thiết bị phát ra âm thanh có tích hợp sẵn mạch công suất phục vụ cho việc giao tiếp và tiêu khiển.

- **Micro:** Micro máy tính là trang bị tích hợp cảm biến để thực hiện việc chuyển đổi âm thanh sang dấu hiệu điện.

- **Webcam (**webcamera): một chiếc máy quay phim khoa học số được chế tác riêng cho máy tính xách tay cá nhân để người dùng mang thể thực hành cuộc gọi video, gửi thư bằng hình ảnh hoặc tải ảnh lên trang web cho mọi người cộng xem.

### **2.1.2 Nhận biết cổng và khe cắm trong máy tính**

#### **2.1.2.1 Khái niệm cổng và chức năng chính**

*Cổng kết nối của máy tính* là một giao diện hoặc một điểm kết nối giữa máy vi tính và các thiết bị ngoại vi của nó. Một số thiết bị ngoại vi phổ biến là chuột, bàn phím, màn hình hoặc bộ hiển thị, máy in, loa, ổ đĩa flash, v.v.

*Cổng máy tính* còn được gọi là cổng giao tiếp vì nó chịu trách nhiệm giao tiếp giữa máy tính và thiết bị ngoại vi của nó.

*Chức năng chính* của cổng máy tính là hoạt động như một điểm gắn kết, nơi cáp từ thiết bị ngoại vi có thể được cắm vào và cho phép dữ liệu truyền từ và đến thiết bị.

#### **2.1.2.2 Cổng nối tiếp và cổng song song**

Trong máy tính, các cổng giao tiếp có thể được chia thành hai loại dựa trên loại hoặc giao thức được sử dụng để giao tiếp, bao gồm:

- *Cổng nối tiếp*: Cổng nối tiếp là một giao diện mà qua đó các thiết bị ngoại vi có thể được kết nối bằng giao thức nối tiếp liên quan đến việc truyền dữ liệu từng bit một trên một đường truyền thông duy nhất. Loại cổng nối tiếp phổ biến nhất là cổng kết nối D-Subminiature hoặc D-sub mang tín hiệu RS-232.

- *Cổng song song*: Cổng song song là một giao diện mà qua đó giao tiếp giữa máy tính và thiết bị ngoại vi của nó theo cách song song tức là dữ liệu được truyền vào hoặc ra song song bằng cách sử dụng nhiều hơn một đường dây hoặc dây giao tiếp. Cổng máy in là một ví dụ về cổng song song.

#### **2.1.2.3. Các cổng phổ biến. Công dụng của từng loại cổng, tốc độ truyền dữ liệu**

- **Cổng kết nối USB-Type A**:

A close-up of a usb cable

Description automatically generated with medium confidence

Hình 2.2: Cổng kết nối USB-Type A

+ Đây là loại cổng kết nối được sử dụng phổ biến nhất, cho phép kết nối điện thoại với máy tính và với hầu hết các thiết bị ngoại vi hiện nay như chuột máy tính, bộ nhớ USB, cáp USB điện thoại và nhiều phụ kiện khác.

+ Trong đó, cổng kết nối USB 2.0 cho tốc độ truyền dữ liệu đạt 60 MB/s. Trong khi với cổng USB 3.0 sở hữu tốc độ truyền dữ liệu gấp 10 lần cổng USB 2.0 với, tương đương khoảng 600-625 MB/s.

+ Vì vậy hầu hết các dòng máy tính xách tay trang bị cổng SSD hoặc các ổ cứng ngoài có tốc độ cao đều sử dụng cổng USB 3.0 để đảm bảo tốc độ kết nối nhanh chóng.

- **Cổng kết nối USB-Type C**:

A close-up of a usb type-c device

Description automatically generated with medium confidence

Hình 2.3: Cổng kết nối USB TYPE C

+ USB-Type C là loại cổng kết nối thế hệ mới được sử dụng thay thế cho kết USB-Type A đã được trang bị cho nhiều dòng máy tính xách tay mới hiện nay. Thậm chí loại kết nối này giờ đây còn được trang bị trên cả các dòng điện thoại smartphone thông minh.

+ USB-Type C có ưu điểm nhỏ gọn và dễ dàng thao tác rút - kết nối. Chuẩn USB type C còn hỗ trợ cả chuẩn USB 3.1 với tốc độ truyền tải dữ liệu tối đa lên tới hơn 1.2GB/s – gấp 2 lần chuẩn USB 3.0 trước đó.

- **Cổng giao thức Thunderbolt 3**:

A close-up of a usb cable

Description automatically generated with low confidence

Hình 2.4: Cổng giao thức Thunderbolt 3

+ Thunderbolt 3 sử dụng cổng kết nối USB-Type C cho khả năng kết nối với ổ cứng và card đồ họa rời, từ đó giúp tăng khả năng xử lý hình ảnh, hỗ trợ tăng độ phân giải màn hình lên đến 5k, cùng một số thiết bị ngoại vi thông dụng khác.

+ Thunderbolt 3 được đánh giá là một trong các cổng giao tiếp của máy tính với giao thức kết nối có tốc độ truyền tải nhanh nhất hiện nay, đạt gần 5GB/s, gấp 4 lần cổng USB 3.1.

- **Cổng kết nối HDMI**:

Close-up of a cable plugged into a computer

Description automatically generated with medium confidence

Hình 2.5: Cổng kết nối HDMI

+ HDMI là một trong các loại cổng kết nối trên máy tính quen thuộc nhất với những người thường xuyên dùng máy tính để trình chiếu hoặc kết nối các thiết bị ngoại vi trình chiếu khác.

+ Cổng HDMI cho phép kết nối máy tính với các thiết bị trình chiếu để truyền tải hình ảnh và âm thanh chất lượng cao. Cổng kết nối này được sử dụng rất phổ biến trên máy tính xách tay hoặc TV, máy chiếu.

+ Cổng HDMI cho phép xuất hình ảnh với độ phân giải lên đến 4K, hỗ trợ 8 kênh 24bit ở tốc độ 192GHz, truyền tải được công nghệ âm thanh tiên tiến như Dolby TrueHD hay DTS-HD Master Audio hỗ trợ trình chiếu.

- **Cổng kết nối VGA**:



Hình 2.6: Cổng kết nối VGA

+ VGA thuộc nhóm các loại cổng kết nối xuất hình ảnh. Mặc dù loại kết nối này được ra đời từ rất lâu rồi nhưng đến nay vẫn được sử dụng phổ biến trên các dòng máy tính xách tay, TV và máy trình chiếu.

+ Một cáp VGA với 15 chân kết nối. Cáp VGA có thể được sử dụng với nhiều độ phân giải. VGA hỗ trợ độ phân giải dao động từ 640 × 400px @ 70 Hz (24 MHz băng thông tín hiệu) đến 1280 × 1024px @ 85 Hz (160 MHz) và lên đến 2048 × 1536px @ 85 Hz(388 MHz).

+ Hiện nay, VGA hỗ trợ nhiều tiêu chuẩn độ phân giải hình ảnh 640x400 pixels và 1280x1024 pixels, thậm chí hiện nay có thể lên đến 1920x1080 pixels Full HD, 2048x1536 pixels.

- **Cổng kết nối LAN RJ-45**:



Hình 2.7: Cổng kết nối LAN RJ-45

+ LAN RJ-45 là loại cổng hỗ trợ kết nối Internet không thể thiếu trên cả máy tính, máy tính xách tay và PC. Đây là một trong các cổng kết nối trên máy tính xách tay được sử dụng để kết nối Internet với tốc độ kết nối ổn định và đường truyền mạnh hơn so với sử dụng kết nối wifi.

+ Ngoài nhược điểm phải kết nối mạng dây khá rắc rối thì cổng LAN RJ-45 hỗ trợ kết nối Ethernet là rất cần thiết và không thể thiếu được trên máy tính PC hay máy tính xách tay.

+ Công nghệ Ethernet mới nhất được gọi là Gigabit Ethernet và hỗ trợ tốc độ truyền dữ liệu trên 10 Gigabit mỗi giây.

#### **2.1.2.4. Khe cắm PCI, ISA**

- **Khe cắm ISA**:

+ Khe cắm ISA là một trong những khe cắm xuất hiện đầu tiên trên Mainboard và lâu đời nhất hiện nay, còn có tên gọi khác là Industry Standard Architecture. Tốc độ truyền dữ liệu giữa các thiết bị ngoại vi của ISA rất kém.

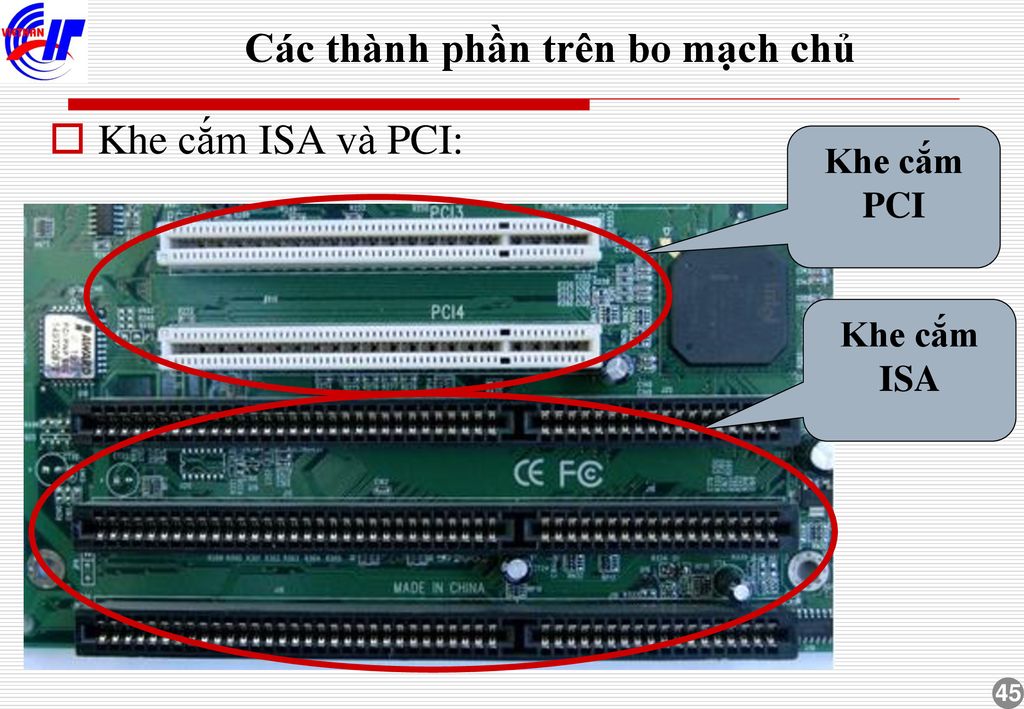
+ Khe cắm này sử dụng phương pháp truyền dữ liệu song song, nên phù hợp với chuẩn dành cho các Mainboard trên các máy tính đời cũ, cho phép chúng thực hiện các giao tiếp với tất cả các loại card mở rộng. Do sử dụng phương pháp này, người dùng muốn tăng dữ liệu truyền, thì phải cần có kích thước lớn.

+ Khe cắm ISA sở hữu **8 bit** và **16 bit**, nên phù hợp với các mainboard hỗ trợ CPU thế hệ cũ như **Pentium 3** có tốc độ xử lý **933 MHz**. Khe cắm có thông số tối đa khá nhỏ nên trong một giây chỉ truyền dung lượng tối đa là **66 MB/s**.

- **Khe cắm PCI**:

+ Khe cắm PCI là sự phát triển thay thế cho khe cắm ISA, nên cũng áp dụng phương pháp truyền song song, nhưng được nâng cấp hơn. Chuẩn kết nối này có 120 chân giúp bạn giao tiếp dữ liệu và kết nối các thiết bị ngoại vi như card mạng, card âm thanh, màn hình... với mainboard.

+ Tuy nhiên, nhược điểm của khe cắm PCI là chia sẻ băng thông. Khi bạn sử dụng nhiều card mở rộng cùng lúc, thì băng thông sẽ chia sẻ đều cho tất cả các card mở rộng được cắm trên mainboard, nên làm mất đi hiệu năng hoạt động.



Hình 2.8: Ke cắm PCI, ISA

## **2.2 Yêu cầu và nguyên tắc xử lý thiết bị ngoại vi**

### **2.2.1. Yêu cầu của quản lý thiết bị [10]**

- Chức năng của các thiết bị ngoại vi là đảm nhiệm việc truyền thông tin qua lại giữa 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 (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.

- 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.

### **2.2.2 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 quản lý có thể kết nối 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 là 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 chuyền 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 phái 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 dạng hoạt động (ngôn ngữ kênh thực chất là các trình điều khiển kênh).

### **2.2.3 Bộ điều khiển DMA**

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

#### **2.2.3.1 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 [11].

#### **2.2.3.2 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 vi xử lý (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. Để điều khiển quá trình đọc DMA ta cần hai tín hiệu hoạt động MEMR (đọc bộ nhớ) và IOW (ghi I/O). Để 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 chi bộ nhớ và tín hiệu chọn thiết bị I/O 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 [12].

# **CHƯƠNG 3: CÁC KỸ THUẬT TRONG QUẢN LÝ THIẾT BỊ NGOẠI VI TRONG LINUX.**

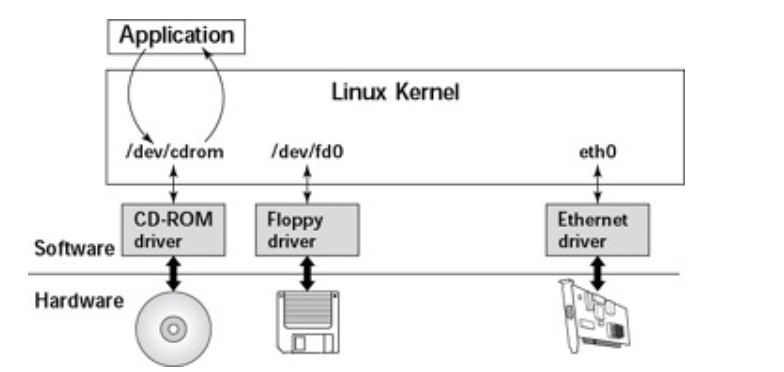
## **3.1 Trình quản lý thiết bị trong hệ điều hành Linux**

Thiết bị ngoại vi được xem như một file chứa trong thư mục /dev có các quyền truy cập giống như file và thư mục. Tất cả các tập tin trong /dev cho phép chương trình truy xuất các thiết bị của kernel Linux. Khả năng xử lý mọi thiết bị này như một tệp thông qua việc sử dụng trình điều khiển thiết bị (device driver) [13].

Device driver là một chương trình đặc biệt kiểm soát một loại phần cứng cụ thể cho phép thiết bị phần cứng giao tiếp với hệ điều hành của máy tính.

Khi Linux Kernel ghi dữ liệu vào thiết bị, trình điều khiển thiết bị làm bất cứ điều gì phù hợp với thiết bị đó. Ví dụ, khi linux ghi dữ liệu vào ổ đĩa mềm, floppy device driver sẽ đặt dữ liệu đó vào môi trường vật lý của đĩa mềm. Mặt khác, nếu linux kernel ghi dữ liệu vào thiết bị cổng song song, trình điều khiển cổng song song sẽ gửi dữ liệu đến máy in được kết nối với cổng song song. Bất kỳ ứng dụng nào cũng có thể truy cập một thiết bị bằng cách mở tệp cụ thể cho thiết bị đó.

Hình bên dưới minh họa cách linux làm việc với thiết bị ngoại vi:



Hình 3.1: Minh họa cách linux làm việc với thiết bị

* 1. **Tệp thiết bị (Device file)**

Như hình 3.1 cho thấy, các ứng dụng có thể truy cập một thiết bị như thể đó là một tệp. Các tệp này là các tệp đặc biệt được gọi là tệp thiết bị và tệp này xuất hiện trong thư mục /dev trong hệ thống tệp của Linux. Nếu sử dụng lệnh ls-1/dev trên cửa sổ terminal để xem danh sách các tệp trong thư mục /dev, người dùng sẽ thấy vài nghìn tệp. Điều này không có nghĩa là hệ thống có vài nghìn thiết bị khác nhau bởi vì[14]:

* Một device driver có thể ứng với một hoặc nhiều device file.
* Một device driver có thể điều khiển một hoặc nhiều thiết bị.
* Một device file có thể được sử dụng bởi nhiều tiến trình.
* Một tiến trình có thể cần dùng nhiều device file.

Đó là lý do tại sao số lượng tệp thiết bị lớn.

Vì là một loại file, nên hoạt động tương tác với device file cũng tương tự như với các file thông thường. Các tương tác này bao gồm mở file (open), đóng file (close), ghi dữ liệu vào file (write), đọc dữ liệu từ file (read). Tuy nhiên, kết quả của các tương tác này thường không giống với kết quả tương tác với file thông thường. Đối với file thông thường, nếu người dùng ghi dữ liệu vào là A, ngay sau đó đọc dữ liệu từ file ra, thì vẫn là A. Những điều này có thể không đúng với device file.

Ví dụ audio device file: Khi người dùng ghi dữ liệu vào file này thì dữ liệu được đưa tới loa, còn khi người dùng đọc dữ liệu từ file này, thì dữ liệu đó đến từ microphone.

Vì vậy, làm thế nào để Linux biết device driver nào tương ứng với device file nào và device driver phải điều khiển thiết bị nào khi đang điều khiển nhiều thiết bị?

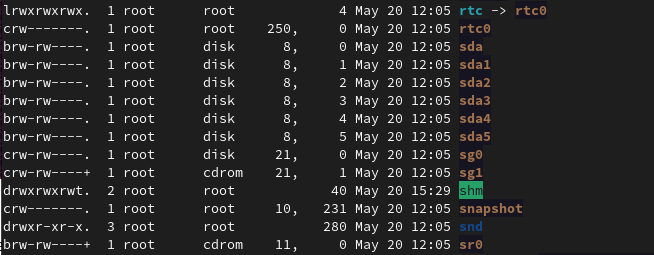
Câu trả lời là nằm ở Device Number. Device Number là bộ gồm 2 số gọi là major number (Số hiệu chính) và minor number (Số hiệu phụ).

* Major number giúp linux nhận biết device driver nào tương ứng với device file.
* Minor number giúp device driver nhận biết nó sẽ phải điều khiển thiết bị nào, nếu như device driver đó đang điều khiển nhiều thiết bị.

Mỗi tệp thiết bị được ánh xạ tới một trình điều khiển thiết bị cụ thể thông qua các số hiệu này. Để xem ví dụ về các số hiệu chính và phụ, hãy nhập lệnh sau vào cửa sổ terminal:

ls -l /dev

Bạn sẽ thấy một dòng đầu ra tương tự như sau:



Hình 3.2: Hiển thị thông tin về các tệp và thiết bị trong thư mục /dev

## **3.3 Cách thức hoạt động của thiết bị ngoại vi**

Có 3 loại thiết bị trong hệ điều hành linux:

* Thiết bị hoạt động ở chế độ khối (block device).
* Thiết bị hoạt động ở chế độ kí tự (character device)
* Thiết bị mạng (network device)

**3.3.1. Block Device[15]**

Khi nhập câu lệnh của 1 tệp thiết bị vào cửa sổ terminal cung cấp một manh mối quan trọng. Đối với thiết bị /dev/hda, chữ cái đầu tiên là b, chỉ ra rằng /dev/hda là một thiết bị hoạt động ở chế độ khối.

Ví dụ hãy nhập lệnh sau vào cửa sổ terminal:

ls -l /dev/hda

Bạn sẽ thấy một dòng đầu ra tương tự như sau:

brw-rw---- 1 root disk 3, 0 Feb 16 14:50 /dev/hda

* + Khối là đơn vị thông tin (dữ liệu) có kích thước cố định (512B, 1KB, 4K, …).
  + Các thiết bị hoạt động ở chế độ khối có 4 tính chất:
* Truy xuất ngẫu nhiên (random access) vào một khối n**i** bất kỳ.
* Sử dụng vùng nhớ đệm ở cấp độ nhân (kernel level).
* Có thể tạo một hệ thống file trên thiết bị này.
* Các thiết bị hoạt động ở chế độ khối được xem như một dải các khối từ 0 đến N – 1.

Ví dụ: đĩa cứng, đĩa mềm, băng từ, …

* Truy xuất thông tin từ các thiết bị gồm 2 bước:

+ Bước 1: Đọc

Chuyển khối dữ liệu từ thiết bị sang bộ nhớ hệ thống (caching).

Sao chép khối dữ liệu từ bộ nhớ hệ thống sang bộ nhớ chương trình.

+ Bước 2: Ghi

Tương tự như đọc, đôi khi kèm theo các cơ chế đồng bộ hóa.

### **3.3.2. Character Device [16]**

Nếu chữ cái đầu tiên trong danh sách của tệp thiết bị là c, khi nhập câu lệnh của 1 tệp thiết bị vào cửa sổ terminal thì thiết bị là một thiết bị hoạt động ở chế độ ký tự.

Ví dụ nhập lệnh sau vào cửa sổ terminal:

ls -l /dev/ttyS0

Bạn sẽ thấy một dòng đầu ra tương tự như sau:

crw-rw---- 1 root uucp 4, 64 Feb 16 14:50 /dev/ttyS0

Các thiết bị hoạt động theo chế độ ký tự thì nó sẽ trao đổi bất kỳ lượng thông tin nào, truy xuất tuần tự, không sử dụng bộ nhớ đệm, và không có hệ thống file trên thiết bị.

Ví dụ: máy in, socket, …

Truy xuất gần như truy xuất file bình thường

- Hai chế độ đọc:

* Theo dạng (line): dữ liệu sẽ được truyền đi khi một ký tự đặc biệt được phát ra (ví dụ: ký tự xuống dòng) cho phép điều khiển luồng dữ liệu, ngắt.
* Theo từng ký tự (raw): đọc từng ký tự một.

### **3.3.3. Network Device**

Các thiết bị mạng, chẳng hạn như các kết nối Ethernet và dial-up point-to-point protocol (PPP), có phần đặc biệt ở chỗ chúng không có tệp tương ứng với thiết bị. Thay vào đó, kernel sử dụng một tên đặc biệt cho thiết bị. Ví dụ, các thiết bị Ethernet được đặt tên là ETH0 cho thiết bị Ethernet đầu tiên, ETH1 cho thiết bị thứ hai, v.v. Các kết nối PPP được đặt tên là PPP0, PPP1, v.v. Vì các thiết bị mạng không được ánh xạ vào các tệp thiết bị, không có tệp tương ứng với các thiết bị này trong thư mục /dev.

## **3.4. Các kĩ thuật trong quản lý thiết bị [10]**

### **3.4.1. Kĩ thuật vùng đệm**

#### **3.4.1.1. Khái niệm và mục đích vùng đệm**

Vùng đệm (buffer) là một vùng 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:

* Kích hoạt thiết bị.
* Chờ thiết bị đạt trạng thái thích hợp.
* 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:

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

#### **3.4.1.2. Phân loại vùng đệm**

Thường chia làm 3 loại:

* + Vùng đệm trung chuyển
  + Vùng đệm xử lí
  + Vùng đệm vòng tròn

**a. 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 thông 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 lí 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

**b. Vùng đệm xử lí**

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 chứ không cần cung cấp giá trị thông tin như vùng đệm trung chuyển.

**ƯU ĐIỂM:**

* Tiết kiệm không gian nhớ
* Rút ngọn 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

**c. Vùng đệm vòng tròn**

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

* Một vùng đệm để đưa thông tin vào
* Một vùng đệm đưa thông tin ra.
* Một vùng đêm xứ lí.

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 lí -> vùng đệm ra -> vùng đệm vào.

Loại vùng đệm này có thẻ gắn vào từng file cụ thể hoặc gắn vào toàn hệ  
thống. Trong chế độ gắn file, vùng đệm được xây dựng khi mở file, xóa khi xuất file và chỉ phục vụ riêng file đó. Phương pháp tổ chức này đặc biệt thích hợp khi mỗi 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 vầ phục vụ cho sự truy nhập file đó. Khi đóng file, vùng đệm không bị xóa mà được trả về cho hệ thông như một tài nguyên chung.

**ƯU ĐIỂM:**

* Vùng đệm này sẽ đạt hiểu quả cao khi thời gian xử lí 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 gan dịch vụ của hệ thống đặc biệt là dàn thông tin của vùng đệm.

### **3.4.2. Kĩ thuật kết khối**

Là ghép nhiều bản ghi logic thành một bản ghi vật lí và việc trao đổi  
thông tin giữa các bộ phận được tiến hành theo bản ghi vật lí.

Thường tồn tại các tổ chức kết khối như sau:

* + Mỗi bản ghi vật lí chứa một số nguyên lần bản ghi logic và giá trị này là  
    như nhau với mọi bản ghi vật lí (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 lí 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 lí có độ dài cố định, không phụ thuộc vào các bản ghi logic.Vì vậy bản ghi vật lí không nhất thiết phải chứa một số nguyên lần các bản ghi logic.
  + Bản ghi vật lí chưa một phần bản ghi logic và vì vậy phải kết hợp nhiều  
    bản ghi vật lí 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. Nó còn được sử dụng như một biện pháp hạn chế việc tuy nhập bất hợp lệ. Nếu không nêu đúng hệ số kết khối thì hệ thông sẽ không tiếp tục thực hiện cấc phép truy nhập thông tin sẽ bị giải mã sai lệnh vì hệ số kết khối đã nêu không hợp lí.

**ƯU ĐIỂM:** Giảm đáng kể số lần truy nhập vật lý.

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

### **3.4.3. Xử lí lỗi**

Bất kì một thành phần nào của hệ thống cũng có thể thưc hiện công việc  
một cách không chuẩn. Tuy nhiên không có bộ phận nào lại bộc lộ nhiều sa 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 bọ mòn, độ nhiễm từ trên các đĩa kém. Phương pháp chử 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 nhanhnhư đĩa từ. việc so sánh và kiểm tra thông thường do các thiết bi điều khiển vào  
ra đảm nhận, sau đó mới thông báo lỗi cho hệ thống và chịu trách nhiệm thực  
hiên các tác dộng tương ứng.

Với mục đích tránh mọi sai sót (như cố gắng đọc đĩa từ khi chưa sẵn  
sàng). Trước va 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 sử 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 bi)

### **3.4.4. SPOOL**

Thông thường, các thiết bị vào/ra được xem xét như những công cụ kỹ thuật để nhận các chương trình kênh và dữ liệu, đồng thời là nơi gửi các mã trạng thái cho hệ thống phân tích.

Nhưng trên thực tế, mọi chương trình và dữ liệu của nó đếu hoạt động hoàn toàn tương tự như thiết bị vào/ra có thực. Như vậy, có thể dùng tiến trình để mô phỏng hoạt động vào/ra và ngược lại, mọi thiết bị có thể coi như các tiến trình.

Trên thực tế, trong nhiều trường hợp, hệ thống đã mô phỏng hoạt động vào/ra bằng con đường chương trình. Các chương trình này có thể hoạt động song song và tuân thủ theo nguyên tắc quản lý tiến trình.

Việc mô phỏng thiết bị ngoại vi đã làm xuất hiện thiết bị ảo. Mỗi thiết bị ngoại vi cộng với chương trình mô phỏng tương ứng sẽ tạo ra thiết bị hoàn toàn khác trong tay người sử dụng.

Ngoài mục đích mô phỏng thiết bị, thiết bị ảo còn có hai ứng dụng khác

* Mô phỏng quá trình điều khiển và quản lý thiết bị mới đang chế tạo hoặc chưa có điều kiện lắp đặt.
* Tạo ra các SPOOL (Simultaneous Pedipheral Operations On Line – hệ thống mô phỏng các phép trao đổi ngoại vi trong chế độ trực tiếp).

Nhiệm vụ của SPOOL là tạo ra hiệu ứng sử dụng song song các thiết bị chỉ được phép khai thác trong chế độ tuần tự. Kỹ thuật SPOOL mô phỏng các thiết bị này bằng các thiết bị ảo và cung cấp cho các tiến trình có yêu cầu. Các tiến trình sẽ gửi thông tin của mình ra thiết bị ảo giống như đối với thiết bị thật và vào thời điểm thích hợp, thông tin từ thiết bị ảo sẽ được chuyển sang thiết bị thật.

*Ví dụ:* Máy in là một thiết bị chỉ có thể hoạt động trong chế độ tuần tự. Khi có nhiều tiến trình cùng có nhu cầu sử dụng máy in thì hệ thống không thể cấp phát nó cho tất cả các tiến trình có nhu cầu. Đối với trường hợp này, hệ thống sẽ mô phỏng acsc máy in ảo và cung cấp cho mỗi tiến trình có nhu cầu một máy ảo. Các tiến trình sẽ gửi thông tinh của mình ra máy ảo như máy in thật. Như vậy, các tiến trình có thể hoạt động song song mà không cần cấp hàng chờ đợi tài nguyên máy in.

SPOOL được sử dụng rộng rãi để thay thế nhiều thiết bị không có khả năng sử dụng chung để nâng cao khả năng hoạt động song song của các tiến trình. Ngoài ra đối với các thiết bị phụ thuộc tốc độ thông tin đầu vào, các tiến trình sẽ nhận được những SPOOL thích hợp để đảm bảo hoạt động bình thường.

Kỹ thuật SPOOL hoạt động bằng cách sử dụng một vùng nhớ đệm trung gian (spool) để lưu trữ dữ liệu in hoặc xử lý trước khi chuyển đến thiết bị tương ứng. Dưới đây là cách thức hoạt động chi tiết của kỹ thuật SPOOL:

* *Đưa vào hàng đợi:* Khi người dùng yêu cầu một tác vụ in hoặc xử lý, dữ liệu tương ứng được đưa vào hàng đợi của kỹ thuật SPOOL. Thay vì gửi dữ liệu trực tiếp đến thiết bị in hoặc xử lý, nó được lưu trữ trong vùng nhớ đệm trung gian.
* *Giải nén và xử lý:* Khi một tác vụ đã được đưa vào hàng đợi, kỹ thuật SPOOL tiến hành giải nén và xử lý tác vụ đó. Nếu đó là tác vụ in, dữ liệu in được giải nén để tạo ra các trang in. Nếu đó là tác vụ xử lý, dữ liệu được trích xuất và chuyển đến chương trình xử lý tương ứng.
* *Quản lý thiết bị:* Kỹ thuật SPOOL theo dõi trạng thái của các thiết bị in hoặc xử lý. Khi một thiết bị trở thành khả dụng, kỹ thuật SPOOL chuyển tiếp dữ liệu từ vùng nhớ đệm trung gian đến thiết bị. Điều này đảm bảo rằng dữ liệu chỉ được gửi đi khi thiết bị đã sẵn sàng nhận nó.
* *Đa nhiệm:* Kỹ thuật SPOOL cho phép xử lý đa nhiệm, tức là có thể có nhiều tác vụ đang diễn ra cùng một lúc. Khi một tác vụ đã hoàn thành, kỹ thuật SPOOL tự động lấy tác vụ tiếp theo từ hàng đợi và tiếp tục xử lý. Điều này cho phép người dùng tiếp tục làm việc mà không cần chờ đợi tác vụ hoàn thành.
* *Ghi log và báo cáo:* Kỹ thuật SPOOL thường cung cấp chức năng ghi log và báo cáo về tiến trình in và xử lý. Người dùng có thể kiểm tra trạng thái của các tác vụ, xem thông báo lỗi và theo dõi quá trình thực hiện.

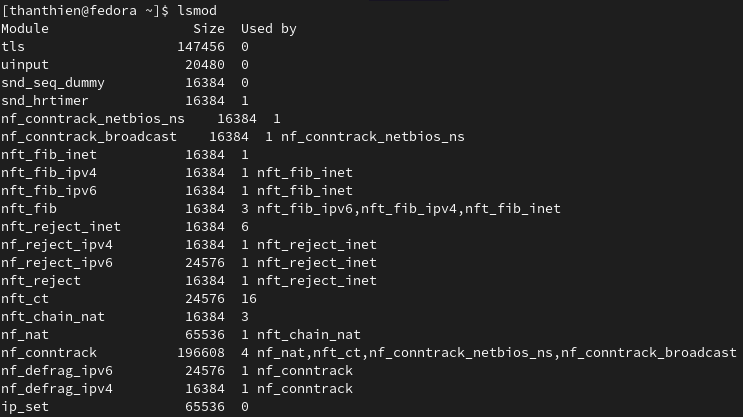
## **3.5. Quản lý Driver modules**

Để sử dụng bất kỳ thiết bị nào, Linux kernel phải chứa driver. Nếu driver code được liên kết vào kernel như một chương trình ở dạng một tệp lớn, thì việc thêm driver mới có nghĩa là xây dựng lại kernel bằng driver code mới. Xây dựng lại kernel có nghĩa là bạn phải khởi động lại PC bằng kernel mới trước khi bạn có thể sử dụng device driver mới. kernel Linux sử dụng một thiết kế mô -đun loại bỏ tất cả các rắc rối khởi động lại. Trình điều khiển thiết bị Linux có thể được tạo dưới dạng các mô-đun mà kernel có thể tải và gỡ bỏ driver mà không phải khởi động lại PC.

***Các câu lệnh quản lý Driver modules***

Bạn phải đăng nhập với quyền root để sử dụng một số lệnh này.

|  |  |
| --- | --- |
| ***Lệnh*** | ***Chức năng*** |
| insmod | Chèn một mô -đun vào kernel |
| rmmod | Loại bỏ một mô -đun khỏi kernel |
| epmod | Xác định sự phụ thuộc lẫn nhau giữa các mô -đun |
| ksyms | Hiển thị danh sách các ký hiệu cùng với tên của mô -đun xác định biểu tượng |
| lsmod | Danh sách tất cả các mô -đun hiện đang được tải |
| modinfo | Hiển thị thông tin về mô -đun kernel |
| modprobe | Chèn hoặc loại bỏ một mô -đun hoặc một tập hợp các mô-đun một cách thông minh. Ví dụ: nếu mô-đun A yêu cầu B, thì modprobe sẽ tự động tải B khi được yêu cầu tải B. |



Hình 3.3: Danh sách những mô-đun đang được tải

## **3.5. Quy ước đặt tên**

Linux xây dựng cơ chế truy xuất đến tất cả các loại đĩa và thiết bị đều ở dạng tập tin (tập tin thiết bị) và lưu trong thư mục */dev* [18].

***Linux quy ước đặt tên như sau:***

- Ổ đĩa mềm: *fd*

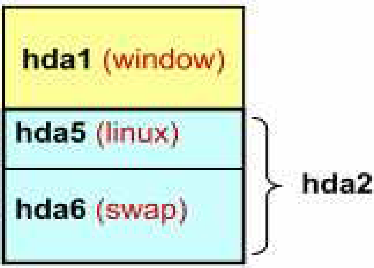
- Ổ đĩa cứng vật lý thứ nhất: *hda*

- Ổ đĩa cứng vật lý thứ hai: *hdb*

- …

Nếu đĩa cứng theo tiêu chuẩn SCSI thì gọi là: *sda, sdb,…* Các thiết bị USB, Linux xem như là thiết bị SCSI (ví dụ nếu máy có một đĩa cứng SCSI thì USB sẽ là *sdb1*).Các phân vùng (partitions) được đánh số sau tên đĩa. Ví dụ: *hda1, hda2,sda1, sdb1 (ổ A), fd1 (ổ B) …* Các phân vùng chính (*primary*) hoặc phân vùng mở rộng (*extended*) đượcđánh số từ 1 -> 4 và các phân vùng logic (nằm trong phân vùng mở rộng) đánh số từ 5 trở đi.

Ví dụ phân vùng đĩa cứng IDE:



Giải thích:

- *hda1*: phân vùng chính.

- *hda2*: phân vùng mở rộng.

- *hda5*: phân vùng logic.

- *hda6*: phân vùng logic.

***Chú ý:*** nếu khi cài đặt Linux mà trước đó đã cài Window, thì Linux sẽ tự động cài đặt vào các phân vùng mở rộng.

## **3.6 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*** [19].

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

Chú ý là lệnh ***mount***chỉ thực hiện được nếu bạn là người dùng có quyền cao nhất *(****root)****.*

***\* Gắn kết một thiết bị lưu trữ***

Cú pháp lệnh: mount [tùy-chọn] <file-thiết-bị> <thư-mục>

* Lệnh mount được dùng để thông báo cho nhân hệ thống biết là tồn tại một hệ thống file nào đó muốn kết nối vào hệ thống file chính tại một điềm gắn nào đó (mount-point).
* Điểm gắn thường là một thư mục của hệ thống file chính và có thể truy cập dễ dàng.
* Điểm gắn kết là thư mục con của mnt.

Ví dụ 1: Mount và sử dụng đĩa mềm [20]:

*# mount /dev/fd0 /mnt/floppy*

Trong lệnh trên, hệ thống sẽ kết nối đĩa mềm *fd0* vào cây thư mục tại điểm nối là */mnt/floppy*. Từ đó bạn có thể vào */mnt/floppy* để truy nhập nội dung ổ đĩa A.

Ví dụ 2: Mount và sử dụng ổ CD:

*# mount /dev/cdrom /mnt/cdrom*

Ví dụ 3: Mount và sử dụng USB:

*# mount /dev/sdb1 /mnt/usb*

* Các tuỳ chọn của lệnh mount
* -t<kiểu > : xác định kiểu của thiết bị, ngoài ra cũng được sử dụng để xác định kiều hệ thống file.
* -a : gắn kết tất cả các file hệ thống (thuộc kiểu được đưa ra) có trong tệp tin fstab (đây là file lưu trữ thông tin về trạng thái của các file hệ thống).
* -n : gắn kết hệ thống file mà không ghi vào file /etc/mtab (đây là file lưu trữ thông tin về các file hệ thống hiện có trên hệ thống).
* -r : kết gắn hệ thống file chỉ có quyền đọc.
* -w : kết gắn hệ thống file có quyền ghi.
* -L nhãn : kết gắn phân vùng được chỉ ra bởi nhãn.
* -U uuid : kết gắn phân vùng được xác định bởi uuid. Hai tùy chọn này chỉ thực hiện được khi file proc/partitions tồn tại (đây là file lưu trữ thông tin về các phân vùng trên hệ thống)

***\* Hủy bỏ gắn kết một thiết bị lưu trữ*** [19]

Cú pháp lệnh: umount [tùy-chọn] <thiết-bị>

* Tháo bỏ kết gắn của hệ thống file có trên thiết bị ra khỏi hệ thống file chính.
* Không thể tháo bỏ kết gắn của một hệ thống file khi nó “bận” - tức là khi có một quá trình đang hoạt động truy cập đến các file trên hệ thống file đó.
  + - Tuỳ chọn
* -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 chi đọc.
* -a : tất cả các file hệ thống được hiển thị trong /etc/mtab đã được loại bỏ các gắn kết.
* -f : bắt buộc phải tháo bỏ các gắn kết.
* Ví dụ: Gỡ kết nối với đĩa mềm

*# umount /dev/fd0 /mnt/floppy*

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

***\* Xây dựng một hệ thống file trên Linux*** [21]

Cú pháp lệnh: mkfs [tùy-chọn] <hệ-thống-file> [khối]

* Lệnh mkfs được sử dụng để tạo một hệ thống file trên thiết bị
* hệ-thống-file hoặc là tên thiết bị (ví dụ /dev/hdai, /dev/sdb2) hoặc là điềm kết nối file hệ thống (ví dụ /, /usr, /home); khối là số khối được sử dụng cho hệ thống file.
* -t kiểu: tùy chọn này xác định kiểu file hệ thống được xây dựng. Nếu không có tùy chọn này, kiểu file hệ thống mặc định sẽ được sử dụng (hiện tại là kiểu ext2).
* -c: kiểm tra thiết bị để tìm ra các khối hỏng trước khi xây dựng hệ thống file.
* Ví dụ: mkfs /dev/hda2

***\* Xem dung lượng đĩa đã sử dụng***

Cú pháp lệnh: du [tùy-chọn] [file] …

* Các tuỳ chọn:
* -a : liệt kê kích thước của tất cả các file
* -b, --bytes : hiển thị kích thước theo byte.
* -c, --total : hiển thị cả tổng dung lượng được sử dụng trong hệ thống file.
* -, --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.
* -l, --count-links : tính kích thước các file nhiều lần nếu được liên kết cứng.
* -m, --megabytes : tính kích thước theo megabytes.
* -S, --separate-dirs : 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.
* Ví dụ: # du /usr/doc/test

***\* Kiểm tra dung lượng đĩa trống***

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 file được kết nối.
* Các tùy chọn
  + -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 có 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=kiều : giới hạn danh sách các file hệ thống thuộc kiều.
  + -T, --print-type : hiển thị các kiều của file hệ thống.
  + --help: đưa ra trang trợ giúp và thoát.
* Ví dụ:

# df /mnt/floppy

Kết quả: Filesystem 1k-blocks Used Available Use% Mounted on

/dev/fd0 1423 249 1174 18% /mnt/floppy

# **KẾT LUẬN**

       Trên cơ sở nhìn nhận một cách tổng quan hệ điều hành, một cách khách quan các ưu nhược điểm, và quan trọng hơn là hiểu rõ hơn về cách quản lý thiết bị ngoại vi của HĐH Linux cũng như xem xét xu hướng phát triển tin học ở nước ta có thể thấy: Đối với người dùng thông thường việc chuyển từ Windows sang Linux trong ngày một ngày hai là chưa thể. Tuy nhiên đối với những người làm tin học, đặc biệt là đối với sinh viên chúng ta, việc tìm hiểu và nghiên cứu Linux và phần mềm mã nguồn mở là một điều kiện rất tốt để nâng cao hiểu biết của mình. Linux dẫu sao vẫn là một hệ điều hành rất có giá trị: chi phí thấp, linh hoạt, ổn đinh, và bảo mật cao. Bản thân em nói riêng, các thành viên nhóm của em nói chung tin rằng, trong tương lai gần, Linux sẽ dần dần trở thành một trong những hệ điều hành hàng đầu trên thế giới.

**TÀI LIỆU THAM KHẢO**

[1]. Steven Vaughan-Nichols. (April 13, 2011). *Twenty Years of Linux according to Linus Torvalds*.

[2]. The Info Valley. (Apr 11, 2020). *What Is Linux: An Overview of the Linux Operating System.* (Truy cập: 15/05/2023). “[medium.com/@theinfovalley097/what-is-linux-an-overview-of-the-linux-operating-system-77bc7421c7e5”](mailto:medium.com/@theinfovalley097/what-is-linux-an-overview-of-the-linux-operating-system-77bc7421c7e5).

[3]. Linus Benedict Torvalds. (05/10/1991). *Free minix-like kernel sources for 386-AT*. (Truy cập: 15/05/2023). “Groups.google.com/g/comp.os.minix/c/4995SivOl9o/m/GwqLJlPSlCEJ.”

[4]. Trần Nguyễn Khoa. (16/07/2012). *Giới thiệu cơ bản về hệ điều hành Linux*.

[5]. James Stanger, Patrick T. Lane. (August 19, 2001). *Hack Proofing Linux: A Guide to Open Source Security*.

[6]. Trần Thị Thanh Hằng. (02/07/2010). *Linux là gì?.*

[7]. Nguyễn Hoàng Giang, Nguyễn Trí Dũng, Vũ Văn Hùng, Bùi Xuân Điệp, Lưu Minh Quân. (2022). *Nghiên cứu tìm hiểu về quản lý bộ nhớ ngoài trong hệ điều hành Linux*.

[8]. Nguyễn Hưng. (19/02/2021). *Linux là gì? Linux hoạt động như thế nào?*. (Truy cập: 16/05/2023). “vietnix.vn/linux-la-gi/”.

[9]. Marta Drozd. (2019). *PHẦN II CÁC BỘ PHẬN CỦA MÁY TÍNH.* (Truy câp: 16/05/2023). “slideplayer.com/slide/14112328/”.

[10]. Nguyễn Thanh Hải, Nguyễn Tuấn Tú, Trần Thanh Huân. (2016). *Giáo trình nguyên lý Hệ điều hành*. H. Đại học sư phạm Hà Nội.

[11]. Vương Quốc Dũng. *Giáo trình Kiến trúc máy tính*. Thống kê, 2020.

[12]. TS. Nguyễn Văn Minh Trí, ThS. Lâm Tăng Đức, Các sinh viên lớp 03SK. (29/12/2018). *Ghép nối và điều khiển thiết bị ngoại vi*.

[13]. Phạm Nguyên Khang, Đỗ Thanh Nghị. (04/2012). *Giáo trình hệ điều hành Linux*.

[14]. Nguyễn Tiến Đạt. (2019). *Linux Device Driver - Giới thiệu Character Driver*.

[15]. eTutorials.org. Part IV: Managing Red Hat Linux, Chapter 20: Advanced System Administration, Managing Devices. “etutorials.org/Linux+systems/red+hat+linux+9+professional+secrets/Part+IV+Managing+Red+Hat+Linux/Chapter+20+Advanced+System+Administration/Managing+Devices/”

[16]. Linux Documentation Project Guides. (05/08/2006). *The Linux Kernel Module Programming Guide*.

[17]. Chu Thành Hưng. (21/07/2019). *Tìm hiểu Linux Kernel - Tùy biến cho phù hợp với hệ thống.* (Truy cập: 15/05/2023). “viblo.asia/p/tim-hieu-linux-kernel-tuy-bien-cho-phu-hop-voi-he-thong-Ljy5VYoklra"

[18]. Hà Quang Thụy; Nguyễn Trí Thành. "Giáo trình Hệ điều hành Unix- Linux". Giáo dục Việt Nam, 2011.

[19]. Lê Thị Thúy Nga, Trần Minh Tuấn. (10/2005). *Giáo trình Thực hành Linux.*

[20]. Đặng Thu Hiền. (22/12/2016). *Bài giảng Hệ điều hành UNIX-Linux - Chương 6: Quản lý thiết bị ngoại vi, các lệnh truyền thông.*

[21]. Đặng Thu Hiền. (22/12/2016). *Bài giảng Hệ điều hành UNIX-Linux - Chương 7: Các lệnh quản lý thiết bị ngoại vi.*