**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**Môn: Hệ điều hành**

**Đồ án**

**TÌM HIỂU**

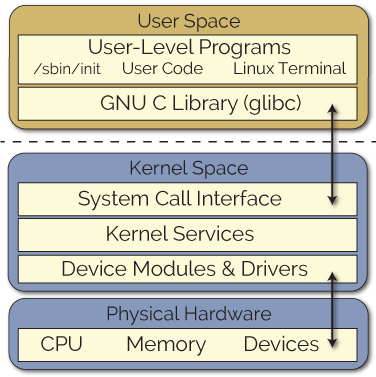
**HỆ ĐIỀU HÀNH LINUX**

**Sinh viên: 1612543 - Phạm Anh Quốc**

09 / 2018

1. **Linux Kernel Module**

Linux là một hệ điều hành mã nguồn mở và miễn phí, được sử dụng khá phổ biến bởi nhiều lập trình viên trên thế giới. Linux, cũng như nhiều hệ điều hành khác, có thể được cài đặt trên nhiều thiết bị phần cứng khác nhau. Để đảm bảo Linux có thể hoạt động và tương thích với nhiều hệ thống phần cứng, một thành phần không thể thiếu trong nhân của Linux là các Module, còn gọi là Linux Kernel Module



Kernel Module là những đoạn code nhị phân được biên dịch và chèn trực tiếp vào phần nhân của hệ điều hành – phần nằm giữa tầng ứng dụng (user-space) bên trên và phần cứng bên dưới. Những Kernel module nằm ở mức logic gần với phần cứng nhất, được cấp đặc quyền truy cập cao nhất (ring 0), vì vậy những kernel module có thể truy cập mọi thứ của hệ thống với tốc độ rất cao.

Với vị trí và khả năng truy cập như vậy, kernel module cùng với các driver, đóng vai trò quan trọng trong sự kết nối giữa tầng ứng dụng và phần cứng. Mỗi module sẽ tương ứng với từng phần cứng khác nhau, cung cấp các hàm chức năng của mỗi thành phần ở tầng vật lý. Nhờ đó, các thành phần trong nhân hệ điều hành có thể hiểu và kết nối được với các phần cứng mà không cần biết từng phần cứng hoạt động như thế nào. Chính nhờ đặc điểm này, mà hệ điều hành Linux có độ linh hoạt cao với nhiều hệ thống phần cứng khác nhau. Nếu không có các module cũng như driver, thì nhân hệ điều hành phải có mã nguồn rất lớn tương thích với tất cả các loại phần cứng mà nó muốn sử dụng. Một thuộc tính quan trọng khác của các Kernel Module là khả năng load và unload khi hệ điều hành đang chạy. Khả năng này cho ứng dụng có thể thêm các module mà không cần phải khởi động lại hệ điều hành.

1. **Hệ thống quản lý file và device trong linux**

Trong Linux có nhiều loại file khác nhau:

* File người dùng: là những file dữ liệu do người dùng tạo ra thông qua các chương trình ứng dụng tương ứng.
* File hệ thống: là những file lưu trữ những thông tin của hệ thống
* File thực thi: là những file chứa những lệnh, chương trình để hệ điều hành thực hiện
* File thiết bị: là những file định danh và đại diện cho một thiết bị, các chương trình ứng dụng có thể làm việc với các thiết bị thông qua file này.
* File liên kết: là những file chứa thông tin tham chiếu đến những file khác

Hệ thống file của linux gồm các thành phần:

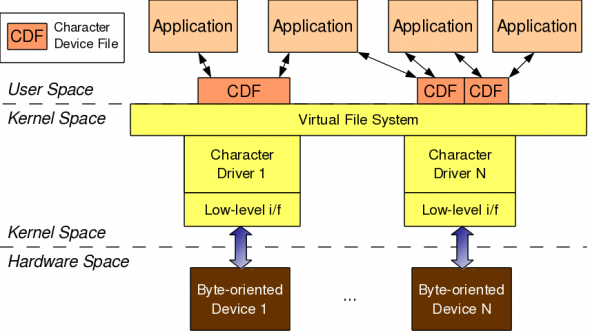
* Boot block: Thường chứa đoạn mã để nạp hệ điều hành lúc máy khởi động.
* Super block: khối nhớ lưu trữ các thông tin của hệ thống file như: kích thước hệ thống file, số khối còn trống, danh sách khối trống, số inode còn trống, danh sách inode còn trống, trường khóa của danh sách khối và inode.
* Inode table: bảng chứa các inode của các file.
* Data block: Vùng lưu trữ phần dữ liệu của các file.

Linux hiển thị cho người dùng hệ thống file dưới dạng cây thư mục với thư mục gốc là root (/), chứa các thư mục con và các file chứa trong các thư mục. Hệ điều hành sẽ quản lý các file thông qua tên file và inode của nó. Mỗi file là một khối dữ liệu được lưu liên tục hoặc không liên tục trên bộ nhớ, và được liên kết với một inode chứa các thông tin về file gọi là metadata. Metadata của file mà inode lưu trữ bao gồm các thông tin: kích thước file, device ID, user ID, group ID, kiểu file và các quyền truy cập, timestamps (các thời điểm file bị thay đổi), số hardlink liên kết đến file, và quan trọng nhất là con trỏ trỏ đến vùng nhớ lưu file. Inode không chứa tên file và tên thư mục.

Khi một file được tạo, nó sẽ được gán cho một file name và nột inode (với inode number duy nhất trong hệ thống). Cặp file name – inode number được lưu thành các mục. Khi người dùng cần truy cập đến một file, hệ điều hành sẽ tìm đến inode tương ứng và thông qua con trỏ vùng nhớ để đến được vị trí file lưu trong bộ nhớ.

Ngoài quản lý file, hệ điều hành còn có hệ thống quản lý các device, thông qua device file, device driver và device thật sự.

* Device file là các file logic, không phải là một thiết bị thật sự, nó nằm ở không gian người dùng, đại diện cho một device nằm ở không gian phần cứng. Khi ứng dụng cần thao tác với một thiết bị, thì ứng dụng đó sẽ làm việc thông qua device file tương ứng.
* Device driver: là các thành phần nằm ở kernel của hệ điều hành, có vai trò tương tự như kernel module đã nói ở trên, giúp liên kết giữa device file và thiết bị phần cứng thực sự. Khi các device file tiếp nhận yêu cầu từ ứng dụng, Virtual file system trong kernel diễn giải và gửi thông tin đến device driver tương ứng.
* Device: là thiết bị phần cứng, mỗi device sẽ có một device driver riêng nằm trong tầng nhân.



1. **Giao tiếp giữa các tiến trình ở user-space và code kernel-space**

Các tiến trình ở User space và Kernel space giao tiếp với nhau thông qua các hệ thống file ảo (virtual file system) và các lệnh gọi hệ thống (system call). Tầng nhân sẽ cung cấp các hàm gọi hệ thống như các API cho các ứng dụng ở user space sử dụng. Ngược lại, khi ứng dụng dùng các hàm này, VFS sẽ đưa các lệnh đến các lớp tương ứng bên dưới.

**Tài liệu tham khảo**

1. <http://derekmolloy.ie/writing-a-linux-kernel-module-part-1-introduction/>
2. <https://blog.sourcerer.io/writing-a-simple-linux-kernel-module-d9dc3762c234>
3. <https://sites.google.com/site/embedded247/ddcourse/device-drivers-phan-1-co-ban-ve-driver-tren-linux>
4. Pablo Neira Ayuso, Rafael M. Gasca1 and Laurent Lefevre – Communicating between the kernel and user-space in Linux using Netlink sockets