**Khái quát Linux Kernel Module:**

Nhân [Linux](https://gocit.vn/bai-viet/tag/linux/) ([kernel](https://gocit.vn/bai-viet/tag/kernel/)) là lớp phần mềm nằm giữa phần cứng và các chương trình ứng dụng chạy trên một máy tính. Nói một cách chặt chẽ theo đúng nghĩa khoa học máy tính, danh từ “[Linux](https://gocit.vn/bai-viet/tag/linux/)” chỉ dùng để chỉ phần nhân mà Linus Torvalds đã viết đầu những năm 90.

Linux được thiết kế để làm việc với hàng tỉ thiết bị. Nhưng ta không thể đưa tất cả các driver vào trong kernel được, vì sẽ làm cho kích thước kernel rất lớn. Giải pháp cho vấn đề này đó là: thiết kế các driver dưới dạng module tách rời với kernel. Trong quá trình hoạt động, driver nào cần thiết sẽ được lắp vào kernel, còn driver nào không cần thiết sẽ bị tháo ra khỏi kernel (dynamic loading).

Các nhân Linux đời đầu có cấu trúc đơn khối (monolithic): các bộ phận của nhân liên kết cố định với nhau tạo nên một file chạy lớn.

Nhân Linux hiện nay có cấu trúc module: mỗi module chức năng chỉ được tải vào nhân khi cần đến. Do đó nhân có kích thước nhỏ và có thể tải hoặc thay thế các module trong một nhân đang chạy mà không cần phải reboot.

**Quản lý file và device trên Linux (đã xin phép thầy cho nộp sau):**

**User space và Kernel space:**

Bộ nhớ RAM chứa các lệnh/dữ liệu dạng nhị phân của Linux kernel và các tiến trình. RAM được chia làm 2 miền (hình 4).

Kernel space là vùng không gian chứa các lệnh và dữ liệu của kernel. Nó chạy trong phần dành riêng của bộ nhớ. Vai trò của không gian hạt nhân là quản lý các ứng dụng / quy trình đang chạy trong không gian người dùng. Nó có thể truy cập tất cả các bộ nhớ. Nếu một quá trình thực hiện một lời gọi hệ thống (system call), một lệnh ngắt phần mềm được gửi đến kernel, sau đó gửi một trình xử lý ngắt thích hợp.

User space là vùng không gian chứa các lệnh và dữ liệu của các tiến trình. Nó được thiết lập các vị trí nơi các tiến trình của người dùng bình thường chạy. Các tiến trình này không thể truy cập không gian hạt nhân trực tiếp.

A screenshot of a cell phone

Description automatically generated

CPU có 2 chế độ thực thi (hình 4):

* Khi CPU thực thi các lệnh của kernel, thì nó hoạt động ở chế độ kernel mode. Khi ở chế độ này, CPU sẽ thực hiện bất cứ lệnh nào trong tập lệnh của nó, và CPU có thể truy cập bất cứ địa chỉ nào trong không gian địa chỉ.
* Khi CPU thực thi các lệnh của tiến trình, thì nó hoạt động ở chế độ user mode. Khi ở chế độ này, CPU chỉ thực hiện một phần tập lệnh của nó, và CPU cũng chỉ được phép truy cập một phần không gian địa chỉ.

Để dễ hình dung hãy nghĩ về hệ thống máy tính như một ngôi nhà của một gia đình trong đó không gian hạt nhân là danh sách các công việc mà cha mẹ chịu trách nhiệm và người dùng là bất kỳ đứa con nào. Vì vậy, những đứa trẻ không được can thiệp vào những gì cha mẹ làm như trả hóa đơn tiền điện, chúng biết rằng cha mẹ sẽ giữ mái ấm trong tình trạng tốt nhất có thể và những đứa trẻ không cần phải lo lắng về điều gì. Trẻ em làm những việc như bài tập về nhà hay xem ti vi ( với không gian người dung) cha mẹ (hạt nhân) giám sát và cha mẹ cũng đưa ra thời gian giới nghiêm như hạt nhân tạo ra những hạn chế đối với tài nguyên có sẵn mà người dùng có thể sử dụng để thực hiện công việc của nó.

Để hiểu rõ hơn, ta xét tiến trình 1 trong hình 4. Tiến trình này gồm nhiều lệnh nhị phân, tương ứng với 2 lệnh C. CPU sẽ lần lượt lấy các lệnh này ra và thực thi. Lệnh thứ nhất, "a = 5 + 10", là một lệnh tính toán, sẽ được CPU thực thi ở chế độ user mode. Lệnh thứ hai, "printf("%d", a)", là một lệnh vào/ra. Hàm "printf" sẽ gọi system call "write" để yêu cầu Linux kernel in thông tin ra màn hình. Khi đó, CPU sẽ chuyển sang chế độ kernel mode để thực thi các lệnh của Linux kernel.

A close up of a map

Description automatically generated

Tham khảo nguồn :

* <https://vimentor.com/vi/lesson/gioi-thieu-ve-linux-kernel-1>
* <https://www.quora.com/What-is-the-difference-between-user-space-and-the-kernel-space>
* <https://gocit.vn/bai-viet/linux-kernel-lam-viec-nhu-the-nao/>
* <https://bom.to/NDJWGC>
* <https://freetuts.net/quan-ly-file-tren-linux-422.html>
* <https://sites.google.com/site/embedded247/ddcourse/device-drivers-phan-1-co-ban-ve-driver-tren-linux>
* https://blogd.net/linux/tong-quan-ve-filesystem-tren-linux/