|  |  |
| --- | --- |
|  | **ĐỒ ÁN 1. TÌM HIỂU VÀ LẬP TRÌNH LINUX KERNEL MODULE**  ***Nguyễn Phúc Minh - 1751081***  ***Lê Quốc Huy - 1751072***  **1. Mô tả yêu cầu:** |
|  | Mục tiêu hiểu về Linux kernel module và hệ thống quản lý file và device |
|  | trong linux, giao tiếp giữa tiến trình ở user space và kernel space. |
|  | ▪ Viết một module dùng để tạo ra số ngẫu nhiầuên. |
|  | ▪ Module này sẽ tạo một character device để cho phép các tiến trình ở |
|  | user space có thể open và read các số ngẫu nhiên |
|  |  |
|  | **2. Mức độ hoàn thành:** 100% |
|  |  |
|  | 3**. Nội dung Báo cáo**  Mục tiêu hiểu về Linux kernel module và hệ thống quản lý file và device  trong linux, giao tiếp giữa tiến trình ở user space và kernel space.  Gồm 3 mục:   1. Linux kernel module 2. Hệ thống quản lý file và device 3. Giao tiếp giữa tiến trình ở userspace và kernel space. |
|  |  |
|  |  |
|  | **1 Linux kernel module** |
|  |  |
|  | **Linux kernel module**: là một file với tên mở rộng là (.ko). Nó sẽ được lắp vào hoặc tháo ra khỏi kernel khi cần thiết. Chính vì vậy, nó còn có một tên gọi khác là loadable kernel module. |
|  | **Driver**: là một trong những kiểu loadable kernel module phổ biến . |
|  | **Loadable kernel module** mang lại 3 lợi ích: |
|  | - Giúp giảm kích thước kernel. Do đó, giảm sự lãng phí bộ nhớ và giảm thời gian khởi động hệ thống. |
|  | - Không phải biên dịch lại kernel khi thêm mới driver hoặc khi thay đổi driver. |
|  | - Không cần phải khởi động lại hệ thống khi thêm mới driver. Trong khi đối với Windows, mỗi khi cài thêm driver, ta phải khởi động lại hệ thống, điều này không thích hợp với các máy server. |
|  |  |
|  | ***loadable kernel module*** được chia làm 3 loại chính: |
|  | - device driver |
|  | - system call |
|  | - file system |
|  |  |
|  | Kernel module có thể được đưa vào trong kernel mộ*t cách tự động, hoặc thủ công.* |
|  |  |
|  | - ***Đối với trường hợp tự động,*** kernel kích hoạt tiến trình modprobe thông qua kmod hoặc udevd. Sau đó, modprobe sẽ đưa module cần thiết vào. |
|  | Quá trình diễn ra như sau: |
|  |  |
|  | *Bước 1:* Kernel kích hoạt tiến trình modprobe cùng với tham số truyền vào là tên của module (ví dụ xxx.ko). |
|  | *Bước 2:* Tiến trình modprobe kiểm tra file /lib/modules/<kernel-version>/modules.dep xem xxx.ko có phụ thuộc vào module nào khác không. Giả sử xxx.ko phụ thuộc vào module yyy.ko. |
|  | *Bước 3:* Tiến trình modprobe sẽ kích hoạt tiến trình insmod để đưa các module phụ thuộc vào trước (yyy.ko), rồi mới tới module cần thiết (xxx.ko). |
|  |  |
|  | Có 2 cách kernel *kích hoạt tiến trình modprobe:* |
|  |  |
|  | *Cách 1 là sử dụng kmod.* |
|  | Đây là một thành phần của Linux kernel, hoạt động trong kernel space. Khi một thành phần nào đó của kernel cần đưa một module vào trong kernel space, nó sẽ truyền tên module cho hàm request\_module của kmod. Hàm request\_module sẽ gọi hàm call\_usermodehelper\_setup để sinh ra tiến trình modprobe. Các bạn có thể tham khảo mã nguồn của kmod tại /kernel/kmod.c. |
|  | *Cách 2 là sử dụng udevd.* |
|  | Đây là một tiến trình hoạt động trong user space. Nếu một thiết bị cắm vào hệ thống máy tính, thì điện trở trên bus ngoại vi (ví dụ PCI bus hoặc USB bus) sẽ thay đổi và bộ điều khiển (controller) sẽ biết điều này. Khi đó, bus driver sẽ gửi một bản tin lên cho tiến trình udevd. Bản tin này chứa thông tin về thiết bị. Tiến trình udevd sẽ tra cứu file /lib/modules/<kernel-version>/modules.alias để tìm ra driver nào tương thích với thiết bị. Sau đó, udevd sinh ra tiến trình modprobe. |
|  |  |
|  | - ***Đối với trường hợp thủ công***, ta sẽ sử dụng lệnh insmod hoặc modprobe. Còn để đưa kernel module ra khỏi kernel space, ta sẽ sử dụng lệnh rmmod. |
|  |  |
|  | Linux kernel cũng giống như một thư viện giúp lập trình viên xây dựng các kernel module. Khi viết một kernel module, ta cần phải tham chiếu tới các file trong thư mục include/linux. Bất cứ một module nào cũng cần tham chiếu tới file <linux/module.h>. File này chứa 2 macro quan trọng, đó là module\_init và module\_exit. Hai macro này giúp xác định đâu là hàm khởi tạo module, đâu là hàm kết thúc module. Thông thường, ta nên đặt các macro \_\_init trước hàm khởi tạo, và macro \_\_exit trước hàm kết thúc để tiết kiệm bộ nhớ. |
|  | **2. Hệ thống quản lí file và device trong linux** |
|  | **2.1 Hệ thống quản lí file** |
|  | Hệ thống tập tin của Linux và Unix được tổ chức theo một hệ thống phân bậc tương tự cấu trúc của một cây phân cấp. Bậc cao nhất của hệ thống tập tin là thư mục gốc, được ký hiệu bằng gạch chéo “/” (root directory). |
|  | Nằm dưới thư mục gốc (/) có một loạt các thư mục quan trọng của hệ thống tập tin được công nhận ở tất cả các bản phân phối Linux khác nhau. Dưới đây là danh sách các thư mục thông thường được nhìn thấy dưới thư mục gốc (/): |
|  |  |
|  | có nhiệm vụ quản lý dữ liệu trên thiết bị lưu trữ (như ổ cứng, thẻ nhớ). Quản lý dữ liệu gồm các công việc: thêm, tìm kiếm, sửa, xóa dữ liệu. |
|  |  |
|  | 1. / – Root |
|  |  |
|  | - Mở từng tập tin và thư mục từ thư mục Root. |
|  |  |
|  | - Chỉ có Root user mới có quyền viết dưới thư mục này. |
|  |  |
|  | - Lưu ý rằng /root là thư mục gốc của Root user. |
|  |  |
|  | 2. /bin – User Binaries |
|  |  |
|  | - Chứa các tập tin thực thi nhị phân (binary executables). |
|  |  |
|  | - Lệnh Linux phổ biến sử dụng ở chế độ Singer-user mode nằm trong thư mục này. |
|  |  |
|  | - Tất cả user trên hệ thống nằm tại thư mục này đều có thể sử dụng lệnh. |
|  |  |
|  | - Ví dụ: ps, ls, ping, grep, cp. |
|  |  |
|  | 3. /sbin – System Binaries |
|  |  |
|  | - Cũng giống như /bin, /sbin cũng chứa tập tin thực thi nhị phân (binary executables). |
|  |  |
|  | - Lệnh Linux nằm trong thư mục này được sử dụng bởi Admin hệ thống, nhằm mục đích duy trì hệ thống. |
|  |  |
|  | - Ví dụ: iptables, reboot, fdisk, ifconfig, swapon. |
|  |  |
|  | 4. /etc – Configuration Files |
|  |  |
|  | - Chứa cấu hình các tập tin cấu hình của hệ thống, các tập tin lệnh để khởi động các dịch vụ của hệ thống…… |
|  |  |
|  | - Ngoài ra /etc còn chứa shell scripts startup và shutdown, sử dụng để chạy/ngừng các chương trình cá nhân. |
|  |  |
|  | - Ví dụ: /etc/resolv.conf, /etc/logrotate.conf. |
|  |  |
|  | 5. /dev – Files device |
|  |  |
|  | - Chứa các tập tin để nhận biết cho các thiết bị của hệ thống (device files). |
|  |  |
|  | - Bao gồm thiết bị đầu cuối, USB hoặc các thiết bị được gắn trên hệ thống. |
|  |  |
|  | - Ví dụ: /dev/tty1, /dev/usbmon0 |
|  |  |
|  | 6. /proc – Process Information |
|  |  |
|  | - Chưa các thông tin về System Process. |
|  |  |
|  | - Đây là hệ thống tập tin giả có chứa thông tin về các quá trình đang chạy. chẳng hạn như thư mục /proc/{pid} có chứa thông tin về quá trình đặc biệt của pid. |
|  |  |
|  | - Đây là một hệ thống tập tin ảo có thông tin về tài nguyên hệ thống. Chẳng hạn như /proc/uptime. |
|  |  |
|  | 7. /var – Variable Files |
|  |  |
|  | - Var là viết tắt của variable file, lưu lại tập tin ghi các số liệu biến đổi (variable files). |
|  |  |
|  | - Nội dung các tập tin được dự kiến sẽ tăng lên tại thư mục này. |
|  |  |
|  | - Bao gồm: hệ thống tập tin log (/var/log), các gói và các file dữ liệu (/var/lib), email (/var/mail), print queues (/var/spool); lock files (/var/lock); các file tạm thời cần khi reboot (/var/tmp). |
|  |  |
|  | 8. /tmp – Temporary Files (các tập tin tạm thời) |
|  |  |
|  | - Thư mục chứa các tập tin tạm thời được tạo bởi hệ thống và user. |
|  |  |
|  | - Các tập tin tạo thư mục này được xóa khi hệ thống được khởi động lại (reboot). |
|  |  |
|  | 9. /usr – User Programs |
|  |  |
|  | - Chứa các ứng dụng, thư viện, tài liệu và mã nguồn các chương trình thứ cấp. |
|  |  |
|  | - /usr/bin chứa các tập tin của các ứng dụng chính đã được cài đặt cho user. Nếu bạn không tìm thấy user binary tại thư mục /bin, bạn có thể tìm tại thư mục /usr/bin. Ví dụ như at, awk, cc, less, scp. |
|  |  |
|  | - /usr/sbin có chứa các tập tin ứng dụng cho Admin hệ thống. Nếu không tìm thấy hệ nhị phân tại /sbin, bạn có thể tìm tại /usr/sbin. Chẳng hạn như atd, cron, sshd, useradd, userdel. |
|  |  |
|  | - /usr/lib chứa thư viện /usr/bin và /usr/sbin. |
|  |  |
|  | - /usr/local chứa các chương trình user mà bạn cài đặt từ nguồn. |
|  |  |
|  | Chẳng hạn khi bạn cài đặt apache từ nguồn, apache nằm dưới /usr/local/apache2. |
|  |  |
|  | 10. /home – thư mục Home |
|  |  |
|  | - Thư mục chính lưu trữ các tập tin cá nhân của tất cả user. |
|  |  |
|  | - Ví dụ: /home/john, /home/nikita. |
|  |  |
|  | 11. /boot – Boot Loader Files |
|  |  |
|  | - Chứa các tập tin cấu hình cho quá trình khởi động hệ thống. |
|  |  |
|  | - Các file Kernel initrd, vmlinux, grub nằm trong /boot. |
|  |  |
|  | - Ví dụ: nitrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic. |
|  |  |
|  | 12. /lib – System Libraries |
|  |  |
|  | - Chứa các file thư viện hỗ trợ các thư mục nằm dưới /bin và /sbin. |
|  |  |
|  | - Tên file thư viện có thể là ld\* hoặc lib\*.so.\*. |
|  |  |
|  | - Ví dụ: ld-2.11.1.so, libncurses.so.5.7. |
|  |  |
|  | 13. /opt – Optional add-on Applications |
|  |  |
|  | - Opt là viết tắt của Optional (tùy chọn). |
|  |  |
|  | - Chứa các ứng dụng add-on từ các nhà cung cấp. |
|  |  |
|  | - Ứng dụng add-on được cài đặt dưới thư mục /opt/ hoặc thư mục /opt/ sub. |
|  |  |
|  | 14. /mnt – Mount Directory |
|  |  |
|  | - Gắn kết các thư mục hệ thống tạm thời (thư mục Temporary) nơi Sysadmins có thể gắn kết các file hệ thống. |
|  |  |
|  | 15. /media – Removable Media Devices |
|  |  |
|  | - Gắn kết các thư mục Temporary (thư mục tạm thời) được hệ thống tạo ra khi một thiết bị lưu động (removable media) được cắm vào như đĩa CDs, máy ảnh kỹ thuật số... |
|  |  |
|  | - Ví dụ: /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer. |
|  |  |
|  | 16. /srv – Service Data |
|  |  |
|  | - Svr viết tắt của service. |
|  |  |
|  | - Chứa các service của máy chủ cụ thể liên quan đến dữ liệu. |
|  |  |
|  | - Ví dụ: /srv/cvs chứa dữ liệu liên quan đến CVS. |
|  | **2.2 Hệ thống quản lí device** |
|  | Trong linux, đọc/ghi dữ liệu từ thiết bị cũng tương tự như đọc/ghi dữ liệu file thông thường:  \_ Gọi system call để tương tác device file.  \_ Kernel tạo 1 entry point của device driver tương ứng với device file. Mỗi entry point là 1 hàm của device driver. Device driver sẽ đăng kí các hàm này với kernel, thông thường là tương ứng 1-1 giữa system call của kernel với entry point của device driver.  \_ Cuối cùng, device driver sẽ hướng dẫn CPU giao tiếp với device.  Nên nhớ rằng device file chỉ đóng vai trò là 1 cổng thông tin cho device drivers, dữ liệu đi từ os/app -> device file -> device driver, cuối cùng đến thiết bị vật lí và ngược lại. |
|  |
|  |  |
|  | [*https://opensource.com/article/16/11/managing-devices-linux*](https://opensource.com/article/16/11/managing-devices-linux) |
|  | *https://vimentor.com/vi/lesson/linux-kernel-module* |
|  |  |
|  | **3 Giao tiếp giữa tiến trình ở user space và kernel space.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  | \_ **User space:** là nơi thực thi các ứng dụng, chương trình của người dùng cũng như chứa thư viện GNU C (glibc)  \_ **Kernel space:** là vùng không gian chứa linux kernel, có quyền truy cập không hạn chế vào bất kỳ địa chỉ của bộ nhớ và phần cứng. Cơ chế của kernel space ngăn người dùng truy cập trái phép tài nguyên của kernel để bảo vê kernel cũng như toàn bộ hệ thống. Vì vậy khi user gọi kernel hoặc điều khiển device driver đều thông qua các API, còn được gọi là system call. |

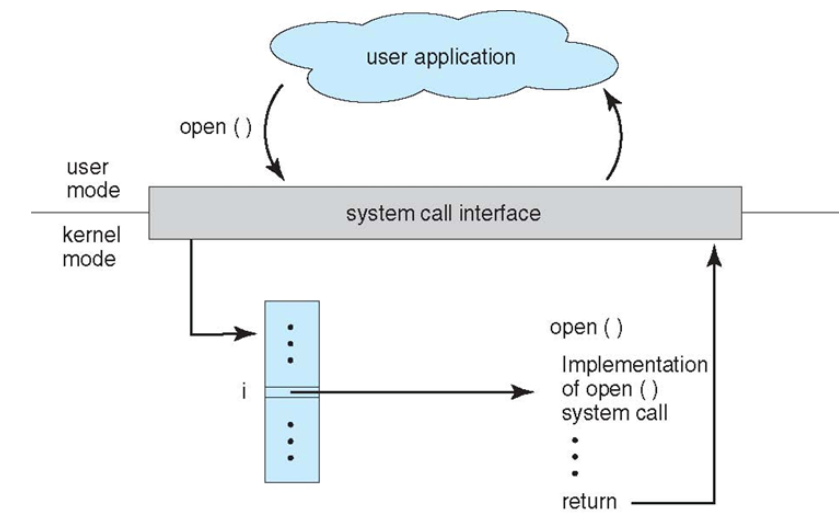
Ví dụ: khi ta sử dụng function thông dụng open() tương ứng system call open

\_ Chương trình của người dùng thiết lập các biến cho system calls. Ví dụ: tên file open,…

\_ Sau khi có đủ các biến, chương trình thực hiện system call instruction.

\_ Instruction đó dẫn đến exception: “event xảy ra buộc processor phải đến địa chỉ mới và thực thi đoạn code ngay tại đó”.

\_ Instructions tại địa chỉ mới lưu trữ trạng thái của người dùng, xác định loại system call (system call open), sau đó gọi xuống kernel để thực thi system call đó, cuối cùng trả về trạng thái cho người dùng và trả quyền truy cập lại cho người dùng.



[*https://stackoverflow.com/questions/24036214/how-does-a-system-call-work*](https://stackoverflow.com/questions/24036214/how-does-a-system-call-work)