

Đồ án 2 hệ điều hành - Mô tả tổ chức/thiết kế của đồ án

MSSV: 1512363

Họ và tên: Nguyễn Hoàng Văn Nhã

Môi trường: Ubuntu 14.04

Linux kernel: 4.4.0-142-generic

Các package yêu cầu: make, gcc

Phần 1. Linux kernel module dùng để tạo ra số ngẫu nhiên.

1.1 Phần module.

- rand_module là một linux kernel module dùng để tạo ra số ngẫu nhiên
- Hàm `init_random` sẽ đăng ký số hiệu module, tên thiết bị, lớp thiết bị để cho các process ở userspace có thể dùng khi lắp module vào.
- Hàm `exit_random` sẽ hủy các đăng ký ở trên khi tháo module ra.
- rand_module định nghĩa một file_operations bao gồm các thao tác lên file thiết bị mà ta đã định nghĩa như: open, read, write, release.
- trong hàm `rand_dev_read` (tức read trong file_operations) sẽ trả về số ngẫu nhiên khi userspace process yêu gọi đến module này.

1.2 Phần test.

- file random_test.c có hàm main gọi đến sys_call open với địa chỉ file thiết bị được định nghĩa trong rand_module.c
- sau đó gọi sys_call read sys_call và hiển thị số ngẫu nhiên nhận được từ file device.

Phần 2. Chương trình hook vào một system call.

Hook vào sys_call là một linux kernel module có chức năng thay thế các sys_call bằng hàm tùy chỉnh của mình.

1. Hàm `entry_point` (module_init)

- Gọi hàm `get_sys_call_table` để tìm địa chỉ của sys_call_table.
- Sau khi có sys_call_table ta lưu lại địa chỉ cũ và thay thế địa chỉ của `__NR_write` và `__NR_open` bằng hàm của ta.
- Gọi hàm `allow_write` và `disable_write` để cho phép và dừng cho phép ghi vào sys_call_table.

2. Hàm `exit_point` (module_exit)

- Gọi hàm `allow_write` và `disable_write` để cho phép và dừng cho phép ghi
- Khôi phục lại địa chỉ của `__NR_write` và `__NR_open` theo mặc định.

3. Hàm `get_sys_call_table`

Lấy địa chỉ của `sys_call_table`

4. Hàm `new_open`

- In ra dmesg thông tin process mở file và tên file.
- Gọi đến sys_call open mặc định để thực thi yêu cầu.

5. Hàm `new_write`

- Gọi đến sys_call write mặc định để ghi và đếm số bytes ghi được.
- In ra tên process và số bytes đã ghi.