MÔN: HỆ ĐIỀU HÀNH

BÁO CÁO

***ĐỒ ÁN 02: KERNEL MODULE &***

***HOOK SYSTEM CALL***

1. **Thành viên:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ & TÊN** | **MSSV** | **PHÂN CÔNG** |
| 1 | Nguyễn Khánh Hoàng |  | Phần 1 |
| 2 | Nguyễn Hữu Vinh | 1712206 | Phần 2 |

1. **Phần 1:**
2. **Phần 2:**
3. **Mô tả đồ án**:

Viết chương trình hook vào một system call

* System call **Open**: Ghi vào dmesg tên tiến trình mở file và tên file được mở
* System call **Write**: Ghi vào dmesg tên tiến trình ghi file, tên file bị ghi và số byte được ghi

1. **Mức độ hoàn thành:**

* Hook system call Open: hoàn thành
  + Ghi vào dmesg tiến trình mở file, tên file được mở
* Hook system call Write: hoàn thành
  + Ghi vào dmesg tiến trình ghi file, tên file bị ghi, số byte được ghi

1. **Tổ chức:**

* hookSyscall.c

Chứa mã nguồn chương trình hook system call Open & Write. Bao gồm các hàm:

* **asmlinkage int (\*original\_open) (const char \*pathname, int flags);**
* **asmlinkage ssize\_t (\*original\_write)(unsigned int fd, const void \*buf, size\_t count);**

Đóng vai trò các hàm tạm thời, dùng để lưu lại địa chỉ mặc định trong system call table của hàm Open và Write

* **asmlinkage int new\_open(const char \*pathname, int flags)**

Hàm thay thế system call Open

* **asmlinkage ssize\_t new\_write(unsigned int fd, const void \*buf, size\_t count)**

Hàm thay thế system call Write

* **int make\_rw(unsigned long address)**

Tắt thuộc tính read-only của system call table

* **int make\_ro(unsigned long address)**

Bật thuộc tính read-only của system call table

* **static int \_\_init entry\_point(void)**

Hàm khởi tạo khi hook syscall

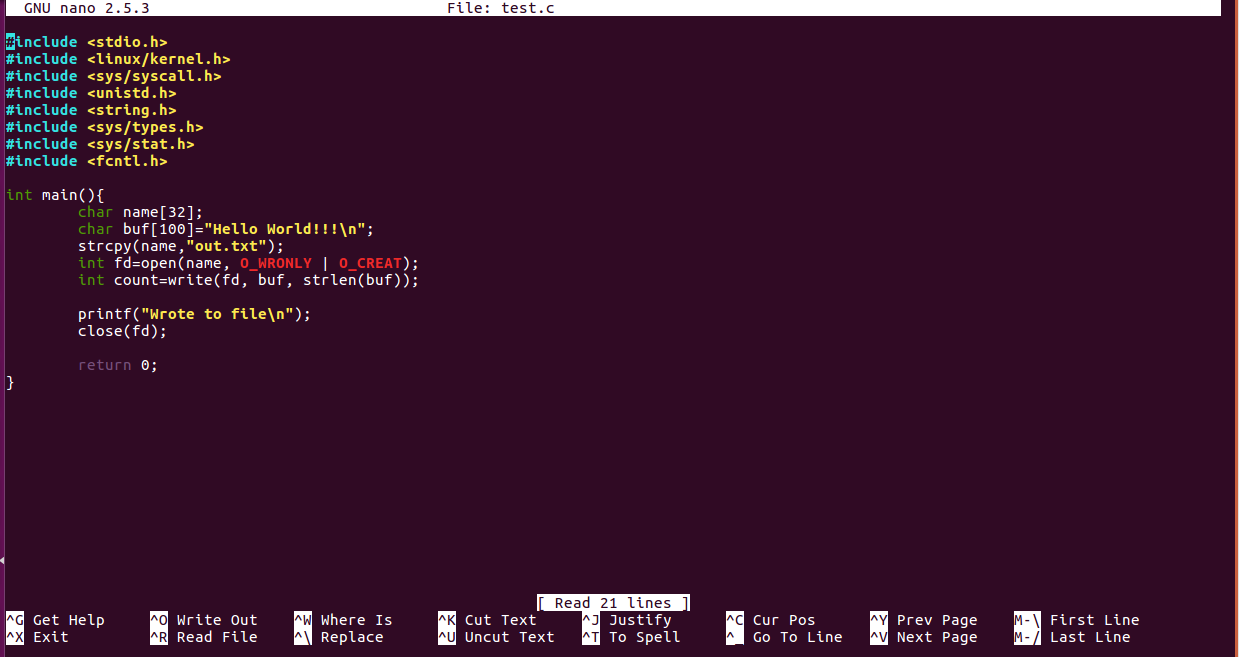
* **static void \_\_exit exit\_point(void)**

Hàm thoát khi gỡ hook syscall

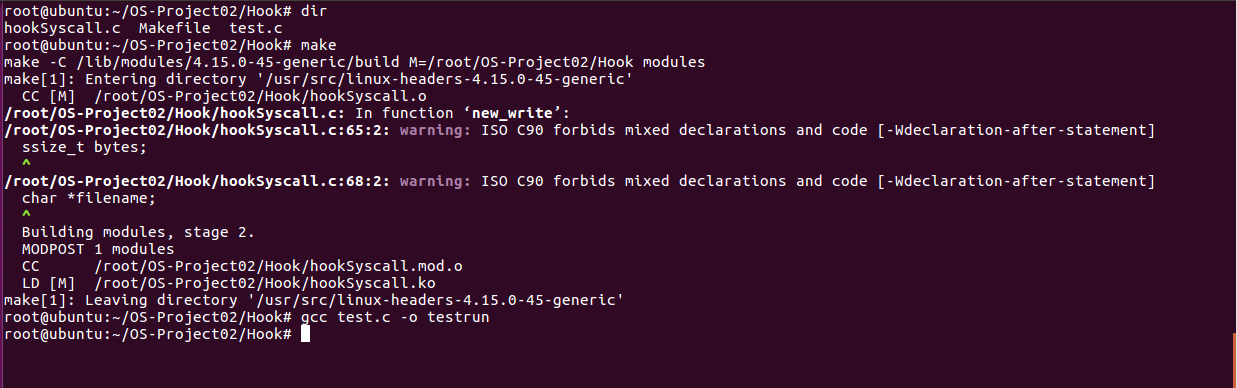
* Makefile

File qui định để thực thi lệnh make

1. **Xử lí:**
2. Tìm địa chỉ system call table
3. Lưu địa chỉ mặc định của hàm Open và Write vào các hàm tạm thời
4. Tắt thuộc tính read-only của bảng system call table
5. Gán địa chỉ hàm thay thế các syscall vào vị trí hàm Open và Write vào bảng
6. Gọi lại hàm mặc định trong hàm thay thế đồng thời thêm các xử lí in ra dmesg
7. Gán địa chỉ ban đầu của hàm Open và Write vào bảng
8. Bật thuộc tính read-only của bảng system call table
9. **Ảnh minh họa:**

Chương trình test.c:

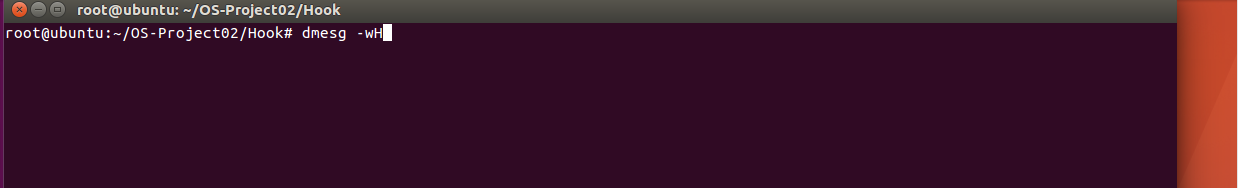
Tiến hành compile test.c sang testrun và compile chương trình hook



Gắn hook module và chạy testrun để kiểm tra



Dùng lệnh dmesg để kiểm tra kết quả in ra





Gỡ hook module và kết thúc



1. **Nguồn tham khảo:**

* https://github.com/GreenCandlePrinter/syscall-hook/blob/master/syscall-hook.c
* https://stackoverflow.com/questions/2103315/linux-kernel-system-call-hooking-example
* https://stackoverflow.com/questions/8250078/how-can-i-get-a-filename-from-a-file-descriptor-inside-a-kernel-module