HƯỚNG DẪN PROJECT 2

System call & Hook

lvlong@fit.hcmus.edu.vn

Systemcall là gì

- Giúp người dùng có thể tương tác với các tiện ích / dịch vụ của Hệ Điều Hành thông qua giao diện, tập lệnh, hoặc câu lệnh
- Systemcall được viết bằng C / C++ hoặc bằng assembly (Hợp ngữ)

Hoạt động của system call

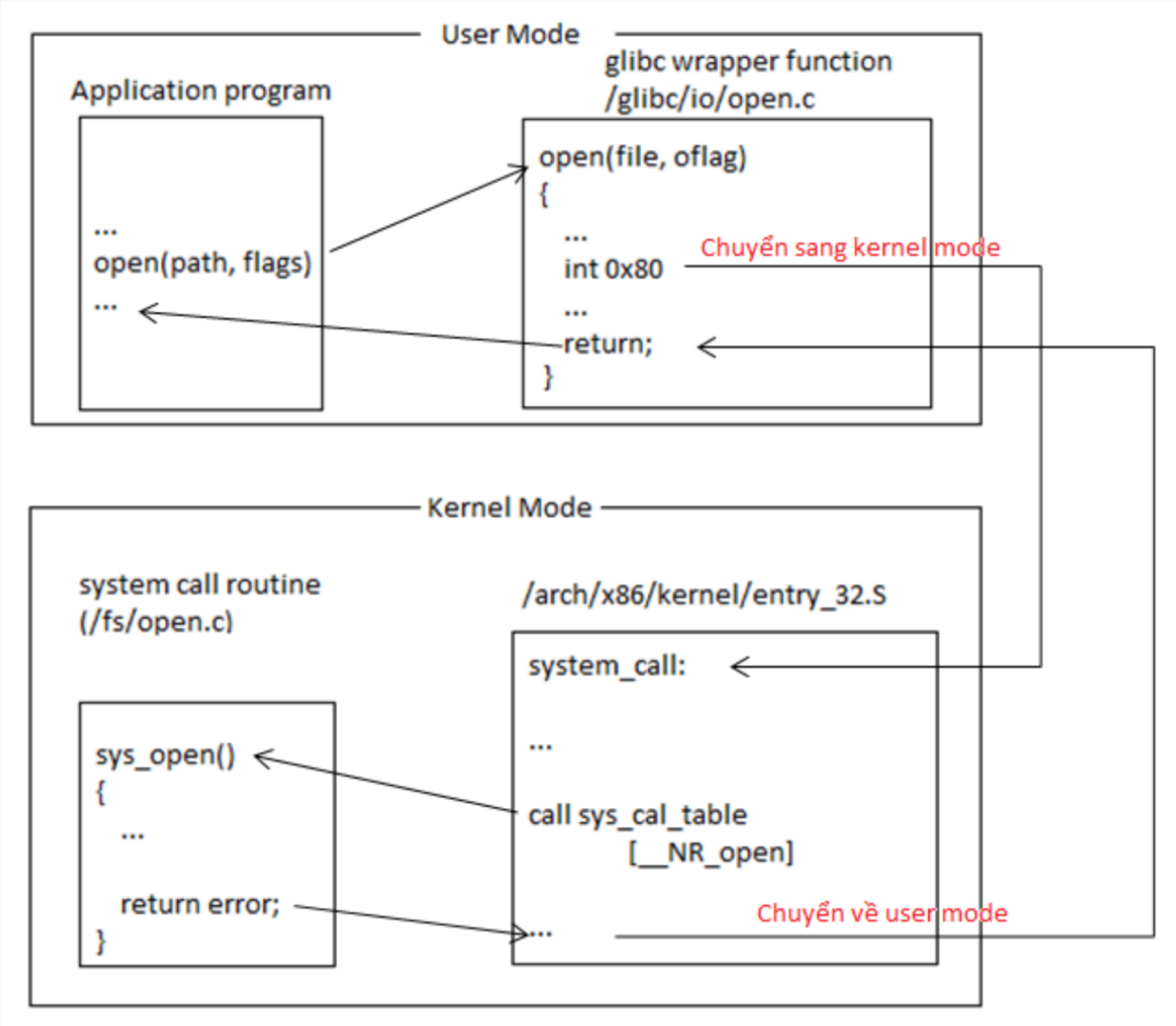
- Không gian bộ nhớ của Hệ Điều Hành được chia thành 2 phần:
  - *User space: quản lý dữ liệu của các tiến trình người dùng*
  - *Kernel space: quản lý dữ liệu của các tiến trình hệ thống*

→ Có 2 chế độ thực thi:
  - *User mode: thực thi các câu lệnh của các tiến trình người dùng*
  - *Kernel mode: thực thi các câu lệnh của tiên trình hệ thống*

==> System call là cửa ngõ vào kernel cho phép tiến trình người dùng yêu cầu kernel thực thi một vài tác vụ cho mình.

Ví dụ

Trong hệ thống linux người dùng có thể gọi đến hàm sys_open() là 1 system call thực thi việc mở một file trong hệ thống file system và trả về id file cho người dung.



Di chuyển dữ liệu giữa user space và kernel space

- Hệ điều hành cung cấp 2 hàm:
  - *copy_from_user*
  - *copy_to_user*

CÁCH VIẾT MỘT SYSTEM CALL

B1. Cài đặt các gói cần thiết cho việc biên dịch kernel

sudo apt-get install libncurses5-dev libncursesw5-dev

sudo apt-get install libssl-dev

sudo apt-get install libelf-dev

sudo apt-get install bison

sudo apt-get install flex

sudo apt-get install bc

sudo apt-get install perl

B2. Download kernel

- Kiểm tra phiên bản kernel hiện tại của hệ thống: uname –r
- Download gói kernel phù hợp tại:

https://mirrors.edge.kernel.org/pub/linux/kernel/

Lưu ý: Không thực hiện trên kernel có sẵn của hệ thống

- Trong ví dụ này sử dụng gói: linux-3.13.tar.xz

wget https://mirrors.edge.kernel.org/pub/linux/kernel/v3.x/linux-3.13.tar.xz

- Giải nén vào thư mục: /usr/src/

sudo tar –xvf linux-3.13.tar.xz –C /usr/src/

B4. Chỉnh sửa lại Makefile của kernel

cd /usr/src/linux-13.3

sudo gedit Makefile

Sử dung cộng cụ Tìm đến dòng

core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/

Bổ sung them khoảng trắng và hello/ vào cuối dòng

core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/ hello/

B5. Định nghĩa mã cho systemcall mới

```
cd /urs/src/linux-3.13/arch/x86/entry/syscalls/
```

- Chỉnh sửa tập tin syscall_32.tbl (nếu máy 64-bit thì là syscall_64.tbl)

```
sudo gedit  syscall_32.tbl
```

- Thêm 1 dòng vào cuối file

```
400 i386 hello sys_hello
```

Mỗi system call sẽ có 1 mã riêng. Trong ví dụ này lựa chọn mã 400 vì chưa được dung. Cần nhớ mã này để sử dung khi gọi hàm systemcall này.

B6. Định nghĩa nguyên hàm trong system call header file

cd /urs/src/linux-3.13/include/linux

sudo gedit syscalls.h

- Thêm dòng asmlinkage long sys_hello(void); vào trước dòng **#endif** cuối cùng

B7. Biên dịch lại kernel

```
cd /usr/src/linux-13.3
sudo make -j 4
sudo make modules_install -j 4
sudo make install -j 4
```

Hoặc

```
sudo make -j 4 && sudo make modules_install -j 4 && sudo make install -j 4
```

Chạy xong reboot. Khi khởi động vào Ubuntu advance option: chọn kernel linux13.3 để sử dụng

Bước 8. Test systemcall

- Viết 1 chương trình C đơn giản gọi đến system call sys_hello

cd /Desktop/

gedit test.c

```
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>
int main()
{
    long int nmt = syscall(400);
    printf("System call sys_hello returned %ld\n", nmt);
    return 0;
}
```

400 là mã của system call đã định nghĩa ở B5.

B9. Biên dịch và xem kết quả

gcc test.c -o test

./test

- *Để xem kết quả tiến trình gọi lệnh dmesg*