

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



MÔN HỌC: HỆ ĐIỀU HÀNH

ASSIGNMENT 1

System Call

Danh sách thành viên:

1. Nguyễn Trung Tính
2. Nguyễn Nhật Tân
3. Đặng Văn Dũng
4. Cao Đăng Dũng

MSSV

1713521
1713074
1710853
1710849



Mục lục

1	Thêm syscall mới	2
1.1	Chuẩn bị Linux Kernel	2
1.2	File configuration	3
1.3	Dùng Kernel Module để thử nghiệm	4

Các bước thực hiện thêm một system call mới để lấy thông tin lịch trình (schedule) của một process bất kì.

1 Thêm syscall mới

1.1 Chuẩn bị Linux Kernel

Tải Ubuntu image bản 12.04 cho VirtualBox và khởi động máy ảo. Sau đó update và cài đặt các gói package cần thiết, chủ yếu là *build-essential* và *kernel-package*.

```
$ sudo apt-get install build-essential
$ sudo apt-get install kernel-package
```

Câu hỏi: Tại sao phải cài đặt *kernel-package*?

Trả lời: Cài đặt *kernel-package* để giúp cho việc cá nhân hóa kernel dễ dàng hơn, nó cũng giúp cho việc compile kernel thuận tiện hơn bằng các script thực hiện tự động.

Tạo thư mục để build kernel và tải bản linux-4.4.56 về.

```
$ mkdir ~\kernelbuild
$ cd ~\kernelbuild
$ wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.56.tar.xz
```

Câu hỏi: Tại sao phải dùng kernel source từ <http://www.kernel.org>?, ta có thể biên dịch trực tiếp bằng kernel cục bộ có trong OS hay không?

Trả lời: Ta hoàn toàn có thể biên dịch trực tiếp bằng kernel có trong OS, nhưng với điều kiện là đủ can đảm và đảm bảo rằng hệ thống đều đã back up khi có lỗi xảy ra. Thay vào đó nếu ta muốn kiểm tra xem có lỗi không và sửa lỗi thì ta nên dùng kernel source tải từ <http://www.kernel.org> sẽ an toàn hơn nhiều.

Cài đặt *openssl package*? và giải nén kernel:

```
$ sudo apt-get install openssl libssl-dev
$ tar -xvJf linux-4.4.56.tar.xz
$ wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.56.tar.xz
```



1.2 File configuration

Copy file `/boot/config` sang thư mục `linux-4.4.56`

```
$ cp /boot/config -x.x.x-generic ~\kernelbuild/linux-4.4.56/.config
```

Cài đặt package `libncurses5-dev`:

```
$ sudo apt-get install libncurses5-dev
```

Trong thư mục thư mục `linux-4.4.56`, mở kernel configuration:

```
$ make nconfig
```

Chọn General setup -> Local version - append to kernel release , sau đó nhập `.1713521 //MSS` vào và lưu lại.

1.3 Dùng Kernel Module để thử nghiệm

Dùng Kernel Module để kiểm tra các bước tìm task struct của process. Hiện thực file test.c để kiểm tra system call như sau:

```
#include <linux/module.h> // included for all kernel modules
#include <linux/kernel.h> // included for KERN_INFO
#include <linux/init.h> // included for __init and __exit macros
#include <linux/proc_fs.h>
#include <linux/sched.h>
#include <linux/pid_namespace.h>
#include <linux/pid.h>
static int pid;
struct task_struct *task;
static int __init procsched_init(void)
{
    printk(KERN_INFO "Starting kernel module!\n");
    for_each_process(task){
        if((int)task->pid==pid){
            printk(KERN_INFO "%d\n",task->pid);
            printk(KERN_INFO "%lu\n",task->sched_info.pcount);
            printk(KERN_INFO "%llu\n",task->sched_info.run_delay);
            printk(KERN_INFO "%llu\n",task->sched_info.last_arrival);
            printk(KERN_INFO "%llu\n",task->sched_info.last_queued);
            return 0;
        }
    }
    return 1;
}
static void __exit procsched_cleanup(void)
{
    printk(KERN_INFO "Cleaning up module.\n");
}
MODULE_LICENSE("GPL");
module_init(procsched_init);
module_exit(procsched_cleanup);
module_param(pid, int, 0);
```



Tài liệu