ГУАП

Кафедра № 63

ОТЧЁТ
ЗАЩИЩЁН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ст.преп.		М.Д. Поляк
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ О КУРСОВОЙ РАБОТЕ

Добавление защиты от несанкционированного запуска операционной системы.

по ОПЕРАЦИОННЫМ СИСТЕМАМ

РАБОТУ ВЫПОЛНИЛ:

СТУДЕНТ ГР. 4331		А.С. Алимов
	подпись, дата	инициалы, фамилия

Санкт-Петербург, 2016

1 Цель работы

Знакомство с устройством ядра ОС Linux. Получение опыта разработки драйвера устройства.

2 Задание

Необходимо внести изменения в процесс загрузки ядра Linux, добавив проверку наличия подключенного через интерфейс USB flash-накопителя с заданным серийным номером. Если в процессе загрузки операционной системы нужный flash-накопитель подключен к одному из портов USB, то операционная система успешно загружается в штатном режиме. Если flash-накопитель с нужным серийным номером отсутствует, отобразить на экране предупреждение и таймер с обратным отсчетом (30 секунд), загрузка операционной системы при этом приостанавливается. По истечении обратного отсчета таймера должно происходить автоматическое выключение компьютера. При подключении к любому из USB-портов нужного flash-накопителя во время обратного отсчета таймера, таймер должен останавливаться, после чего операционная система должна загружаться в штатном режиме.

3 Техническая документация

Действия, выполняемые при добавлении Linux-драйвера, а также следующие специальные действия с флэш-накопителем:

Шаг 1: Собираем драйвер (имя_файла.ko) с помощью запуска команды make.

Шаг 2: Копируем файл в папку /lib/modules/версия ядра/имя_файла.ko.

Шаг 3: Добавляем драйвер в автозагрузку с помощью команды depmod имя_файла.ko.

Шаг 4: Отключаем флеш-устройство при загрузке системы

Шаг 5: Перезгружаем систему

Шаг 6: Вставляем USB флэш-накопитель с заданным номером чтобы разблокировать систему

Скриншот отсчета времени до перезагрузки системы

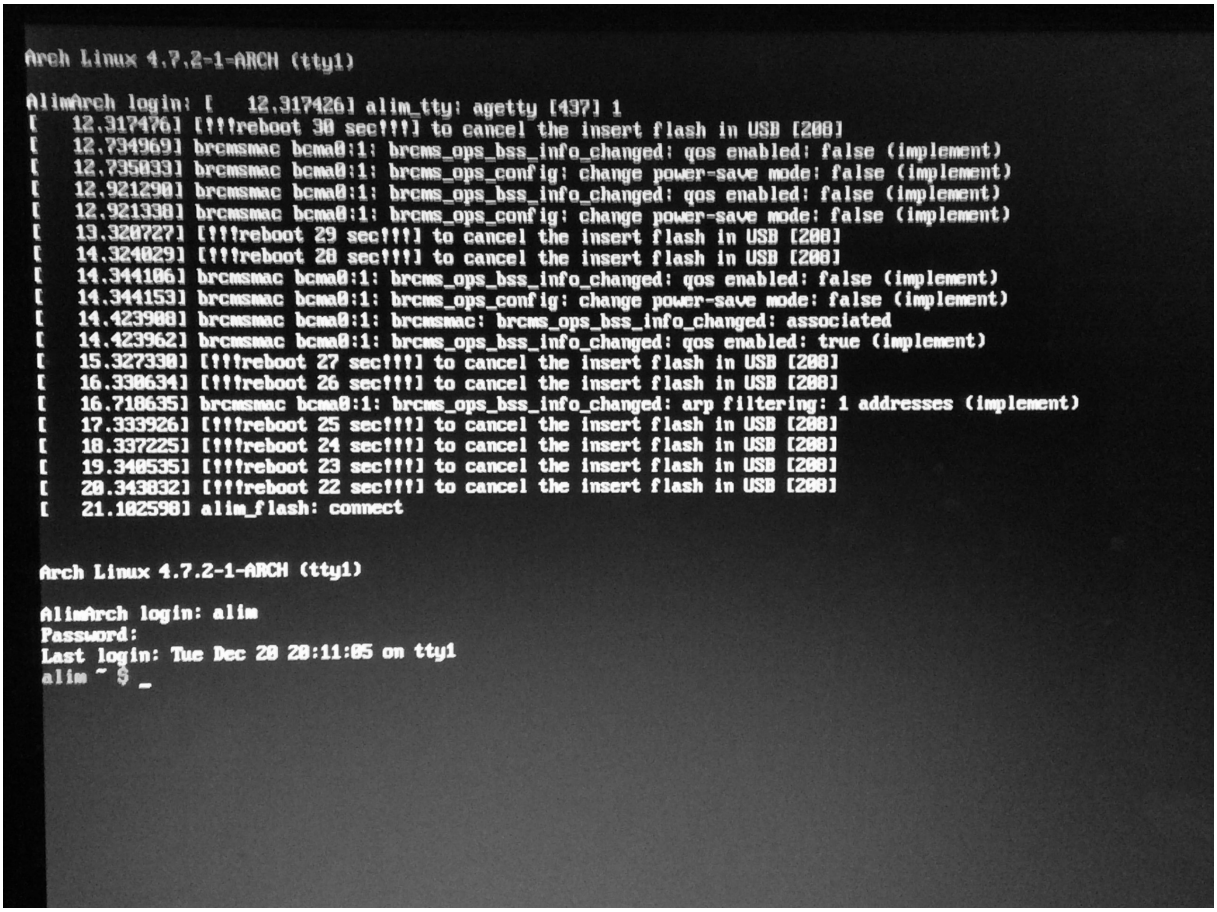

Рис. 1: Лог модуля

Выгружаем драйвер с помощью команды rmmod имя_файла.ko.

4 Вывод

Проделав работу я ознакомился с ядром ОС Linux и получил опыт разработки драйвера USB устройства.

5 Приложение

```
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/usb.h>
#include <linux/sched.h>
#include <linux/kthread.h>
#include <linux/delay.h>
#include <linux/reboot.h>

struct task_struct *tAgetty;
struct task_struct *tTimer;
```

```
struct task_struct *task;
bool stopThread = true;
bool isTimer = true;

static int thread_timer( void * data)
{
int i = 30;

while(stopThread)
{
printk(KERN_ERR "[!!!reboot %i sec!!!] to cancel the insert flash in USB [
ssleep( 1 );
if (i <= 0) kernel_restart(NULL);
i--;
}
return 0;
}

static int thread_agetty_uninterrupyible( void * data)
{
// поток таймера
tTimer = kthread_create( thread_timer, NULL, "thread_sleep" );

// основной цикл потока
while(stopThread)
{
for_each_process(task)
{
if (strcmp(task->comm, "agetty") == 0 && task->state == TASK_INTERRUPTIBLE
{
ssleep(1);
printk(KERN_ERR "alim_tty: %s [%d] %u \n", task->comm , task->pid, (u32)ta
task->state = TASK_UNINTERRUPTIBLE;
// запускаем поток таймера
if (!IS_ERR(tTimer) && isTimer)
{
isTimer = false;
printk(KERN_INFO "alim_thread: %s start\n", tTimer->comm);
wake_up_process(tTimer);
}
}
}
}
return 0;
```

```
}

static int pen_probe(struct usb_interface *interface, const struct usb_dev
{
stopThread = false;
printk(KERN_ERR "alim_flash: connect\n");

for_each_process(task)
{
if (strcmp(task->comm, "agetty") == 0 && task->state == TASK_UNINTERRUPTIB
{
printk(KERN_INFO "alim_flash: %s [%d] %u \n", task->comm , task->pid, (u32
task->state = TASK_INTERRUPTIBLE;
}
}

return 0;
}

static void pen_disconnect(struct usb_interface *interface)
{
printk(KERN_ERR "alim_flash: disconnect\n");
}

static struct usb_device_id pen_table[] =
{
{ USB_DEVICE(0x0C76, 0x0005) },
{ USB_DEVICE(0x8564, 0x1000) },
    {} /* Terminating entry */
};
MODULE_DEVICE_TABLE (usb, pen_table);

static struct usb_driver pen_driver =
{
.name = "alim_driver",
.probe = pen_probe,
.disconnect = pen_disconnect,
.id_table = pen_table,
};

static int __init pen_init(void)
{
printk("alim_init: start\n");
```

```
// поток блокирования tty
tAgetty = kthread_create( thread_agetty_uninterrupyible, NULL, "agetty_uni

if (!IS_ERR(tAgetty))
{
printk(KERN_INFO "alim_thread: %s start\n", tAgetty->comm);
wake_up_process(tAgetty);
}
else
{
printk(KERN_ERR "alim_thread: agetty_uninterrupyible error\n");
WARN_ON(1);
}

return usb_register(&pen_driver);
}

static void __exit pen_exit(void)
{
usb_deregister(&pen_driver);
}

module_init(pen_init);
module_exit(pen_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Alimov Alexandr Sergeevich<a1imov233@gmail.com>");
MODULE_DESCRIPTION("USB Driver");
```