#include <linux/module.h>

#include <linux/interrupt.h>

#include <linux/kernel.h>

#include <linux/init.h>

#include <linux/io.h>

static int irq = 1; // Klaviatura üçün IRQ nömrəsi

static char \*devname = "keyboard\_handler"; // Cihazın adı

// Klaviatura interrupt handler funksiyası

irqreturn\_t keyboard\_isr(int irq, void \*dev\_id) {

unsigned char scancode;

scancode = inb(0x60); // Klaviaturanın portundan scan kodunu oxu

printk(KERN\_INFO "Keyboard interrupt: Scan code: 0x%02x\n", scancode);

return IRQ\_HANDLED;

}

// Modul yükləndikdə işləyəcək funksiya

static int \_\_init keyboard\_init(void) {

// IRQ1 üçün handler qeydiyyatdan keçir

if (request\_irq(irq, keyboard\_isr, IRQF\_SHARED, devname, (void \*)keyboard\_isr)) {

printk(KERN\_ERR "Failed to request IRQ %d for keyboard\n", irq);

return -EIO;

}

printk(KERN\_INFO "Keyboard interrupt handler registered on IRQ %d\n", irq);

return 0;

}

// Modul silindikdə işləyəcək funksiya

static void \_\_exit keyboard\_exit(void) {

free\_irq(irq, (void \*)keyboard\_isr); // IRQ azad et

printk(KERN\_INFO "Keyboard interrupt handler removed from IRQ %d\n", irq);

}

module\_init(keyboard\_init); // Modulun giriş nöqtəsi

module\_exit(keyboard\_exit); // Modulun çıxış nöqtəsi

MODULE\_LICENSE("GPL");

MODULE\_DESCRIPTION("Simple Keyboard Interrupt Handler");

MODULE\_AUTHOR("Farid Karimli");

// ================= TIMER INTERRUPT MODULE ====================

#include <linux/module.h>

#include <linux/timer.h>

#include <linux/jiffies.h>

static struct timer\_list my\_timer; // Timer obyekti

// Timer interrupt callback funksiyası

void my\_timer\_callback(struct timer\_list \*t) {

printk(KERN\_INFO "Timer interrupt occurred at jiffies: %lu\n", jiffies);

mod\_timer(&my\_timer, jiffies + msecs\_to\_jiffies(1000)); // Timeri 1 saniyə sonra yenidən qur

}

// Modul yüklənəndə işləyəcək funksiya

static int \_\_init timer\_init(void) {

timer\_setup(&my\_timer, my\_timer\_callback, 0); // Timeri qur

mod\_timer(&my\_timer, jiffies + msecs\_to\_jiffies(1000)); // 1 saniyəlik interval təyin et

printk(KERN\_INFO "Timer module loaded.\n");

return 0;

}

// Modul çıxarılanda işləyəcək funksiya

static void \_\_exit timer\_exit(void) {

del\_timer(&my\_timer); // Timeri sil

printk(KERN\_INFO "Timer module unloaded.\n");

}

module\_init(timer\_init);

module\_exit(timer\_exit);

MODULE\_LICENSE("GPL");

MODULE\_DESCRIPTION("Simple Timer Interrupt Example");

MODULE\_AUTHOR("Farid Karimli");