



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ

«Информатика и системы управления»

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии»

## ОТЧЁТ

По лабораторной работе №9

По курсу: «Операционные системы»

Тема: «Обработчики прерываний»

Студент: Керимов А. Ш.

Группа: ИУ7-64Б

Преподаватель: Рязанова Н. Ю.

Москва

2020

# Тасклет

## Задание.

- Написать загружаемый модуль ядра, в котором зарегистрировать обработчик аппаратного прерывания с флагом `IRQF_SHARED`.
- Инициализировать тасклет.
- В обработчике прерывания запланировать тасклет на выполнение.
- Вывести информацию о таскете, используя или `printk()`, или `seq_file interface`.

Листинг 1: tasklet/my\_tasklet.c

```
1 #include <linux/kernel.h>
2 #include <linux/module.h>
3 #include <linux/interrupt.h>
4 #include <asm/io.h>
5
6 MODULE_LICENSE("GPL");
7 MODULE_AUTHOR("Kerimov A. IU7-64b");
8 MODULE_DESCRIPTION("tasklet");
9
10 #define IRQ 1 /* keyboard irq */
11 static int dev_id;
12
13 char my_tasklet_data[] = "KEYBOARD INTERRUPT";
14
15 void my_tasklet_function(unsigned long data)
16 {
17     int scancode = inb(0x60);
18     if (scancode < 103) {
19         printk(KERN_INFO "my_tasklet: %s: scancode: %d\n", (char *)data, scancode);
20     }
21 }
22
23 DECLARE_TASKLET(my_tasklet, my_tasklet_function, (unsigned long) &my_tasklet_data);
24
25 irqreturn_t my_irq_handler(int irq, void *dev)
26 {
27     if (irq == IRQ) {
28         tasklet_schedule(&my_tasklet);
29         return IRQ_HANDLED;
30     }
31     return IRQ_NONE;
32 }
33
34 static int __init my_tasklet_init(void)
35 {
36     int ret = request_irq(IRQ, my_irq_handler, IRQF_SHARED, "my_irq_handler", &dev_id);
37     if (ret) {
38         printk(KERN_ERR "my_tasklet: my_irq_handler wasn't register\n");
39     } else {
40         printk(KERN_INFO "my_tasklet: module loaded\n");
41     }
42     return ret;
43 }
44
45 static void __exit my_tasklet_exit(void)
46 {
47     tasklet_kill(&my_tasklet);
48     free_irq(IRQ, &dev_id);
49     printk(KERN_DEBUG "my_tasklet: module unloaded\n");
50 }
51
52 module_init(my_tasklet_init)
53 module_exit(my_tasklet_exit)
```

```
user@lenovo: ~/bmstu/OS/sem06/lab09/tasklet
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$ sudo insmod my_tasklet.ko
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$ dmesg
[ 9998.373587] my_tasklet: module loaded
[10000.662055] my_tasklet: KEYBOARD INTERRUPT: scancode: 32
[10000.742620] my_tasklet: KEYBOARD INTERRUPT: scancode: 50
[10000.810491] my_tasklet: KEYBOARD INTERRUPT: scancode: 18
[10000.995668] my_tasklet: KEYBOARD INTERRUPT: scancode: 31
[10001.173746] my_tasklet: KEYBOARD INTERRUPT: scancode: 34
[10001.305500] my_tasklet: KEYBOARD INTERRUPT: scancode: 28
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$ cat /proc/interrupts | grep my_
1:        6575      14219      0          0      IR-IO-APIC  1-edge      i8042, my_irq_handler
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$ sudo rmmod my_tasklet
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$ dmesg | tail -1
[10022.950708] my_tasklet: module unloaded
user@lenovo:~/bmstu/OS/sem06/lab09/tasklet$
```

Рис. 1: Демонстрация работы программы

## Очередь работ

### Задание.

- Написать загружаемый модуль ядра, в котором зарегистрировать обработчик аппаратного прерывания с флагом `IRQF_SHARED`.
- Инициализировать очередь работ.
- В обработчике прерывания запланировать тасклет на выполнение.
- Вывести информацию о тасклете, используя или `printk()`, или `seq_file interface`.

Листинг 2: `workqueue/my_workqueue.c`

```
1 #include <linux/kernel.h>
2 #include <linux/module.h>
3 #include <linux/interrupt.h>
4 #include <linux/workqueue.h>
5 #include <asm/io.h>
6
7 MODULE_LICENSE("GPL");
8 MODULE_AUTHOR("Kerimov A. IU7-64b");
9 MODULE_DESCRIPTION("workqueue");
10
11 #define IRQ 1 /* keyboard interrupt */
12 static int dev_id;
13
14 static struct workqueue_struct *my_wq;
15
16 void my_workqueue_function(struct work_struct *work)
17 {
18     const int scancode = inb(0x60);
19     if (scancode < 103) {
20         printk(KERN_INFO "my_workqueue: KEYBOARD INTERRUPT: scancode: %d\n", scancode);
21     }
22 }
```

```

23
24 DECLARE_WORK(workname, my_workqueue_function);
25
26 irqreturn_t my_irq_handler(int irq, void *dev)
27 {
28     if (irq == IRQ) {
29         queue_work(my_wq, &workname);
30         return IRQ_HANDLED;
31     }
32     return IRQ_NONE;
33 }
34
35 static int __init my_workqueue_init(void)
36 {
37     int ret = request_irq(IRQ, my_irq_handler, IRQF_SHARED, "my_irq_handler", &dev_id);
38     if (ret) {
39         printk(KERN_ERR "my_tasklet: my_irq_handler wasn't registered\n");
40         return ret;
41     }
42
43     if (!(my_wq = create_workqueue("my_queue"))) {
44         free_irq(IRQ, &dev_id);
45         printk(KERN_INFO "my_workqueue: workqueue wasn't created");
46         return -ENOMEM;
47     }
48
49     printk(KERN_INFO "my_workqueue: module loaded\n");
50     return 0;
51 }
52
53 static void __exit my_workqueue_exit(void)
54 {
55     flush_workqueue(my_wq);
56     destroy_workqueue(my_wq);
57     free_irq(IRQ, &dev_id);
58     printk(KERN_INFO "my_workqueue: module unloaded\n");
59 }
60
61 module_init(my_workqueue_init)
62 module_exit(my_workqueue_exit)

```

```

user@lenovo: ~/bmstu/OS/sem06/lab09/workqueue
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$ sudo insmod my_workqueue.ko
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$ dmesg
[10083.159584] my_workqueue: module loaded
[10084.431474] my_workqueue: KEYBOARD INTERRUPT: scancode: 32
[10084.507011] my_workqueue: KEYBOARD INTERRUPT: scancode: 50
[10084.601564] my_workqueue: KEYBOARD INTERRUPT: scancode: 18
[10084.749197] my_workqueue: KEYBOARD INTERRUPT: scancode: 31
[10085.031454] my_workqueue: KEYBOARD INTERRUPT: scancode: 34
[10085.111087] my_workqueue: KEYBOARD INTERRUPT: scancode: 28
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$ cat /proc/interrupts | grep my_
1:        6875      14219      0          0  IR-IO-APIC  1-edge      i8042, my_irq_handler
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$ sudo rmmod my_workqueue
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$ dmesg | tail -1
[10125.110947] my_workqueue: module unloaded
user@lenovo:~/bmstu/OS/sem06/lab09/workqueue$

```

Рис. 2: Демонстрация работы программы