



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»

ЛАБОРАТОРНАЯ РАБОТА №9

Дисциплина Операционные системы

Тема Обработчики прерываний

Студент Набиев Ф.М.

Группа ИУ7-63Б

Оценка (баллы)

Преподаватель Рязанова Н.Ю.

Москва, 2020 г.

1 Тасклеты

1.1 Реализация

```
1 #include <linux/module.h>
2 #include <linux/kernel.h>
3 #include <linux/init.h>
4 #include <linux/interrupt.h>
5
6 #define HANDLED_IRQ 1
7
8 MODULE_LICENSE("GPL");
9 MODULE_AUTHOR("Faris Nabiev");
10
11 static int __init my_tasklet_init(void);
12 static void __exit my_tasklet_exit(void);
13
14 module_init(my_tasklet_init);
15 module_exit(my_tasklet_exit);
16
17 static int dev_id;
18 const char *tasklet_data = "tasklet data";
19
20 // Функция обработки нижней половины
21 void tasklet_handler(unsigned long data);
22
23 // Регистрация тасклета
24 DECLARE_TASKLET(my_tasklet, tasklet_handler, (unsigned long)&tasklet_data);
25
26 void tasklet_handler(unsigned long data)
27 {
28     printk(KERN_INFO "tasklet: state -- %d, count -- %d, data -- %s\n",
29             my_tasklet.state, atomic_read(&my_tasklet.count),
30             (const char *)(&my_tasklet.data));
31 }
32
33 // Обработчик прерывания
34 static irqreturn_t ihandler(int irq, void *dev_id)
35 {
36     // Проверка, что произошло именно нужное прерывание
37     if (irq == HANDLED_IRQ)
38     {
39         printk(KERN_INFO "tasklet: scheduled\n");
40         // Добавление тасклета в очередь
41         tasklet_schedule(&my_tasklet);
42
43         return IRQ_HANDLED; // Прерывание обработано
```

```

44     }
45     else
46         return IRQ_NONE; // Прерывание не обработано
47 }
48
49 // Инициализация модуля
50 static int __init my_tasklet_init(void)
51 {
52     // Совместное использование линии IRQ с другими устройствами
53     if (request_irq(HANDLED_IRQ, ihandler, IRQF_SHARED,
54                     "ihandler", &dev_id))
55     {
56         printk(KERN_ERR "tasklet: Error on request_irq\n");
57
58         return -1;
59     }
60
61     printk(KERN_INFO "tasklet: Module loaded!\n");
62
63     return 0;
64 }
65
66 // Выгрузка модуля
67 static void __exit my_tasklet_exit(void)
68 {
69     // Удаление тасклета
70     tasklet_disable(&my_tasklet);
71     tasklet_kill(&my_tasklet);
72     // Освобождение линии прерывания
73     free_irq(HANDLED_IRQ, &dev_id);
74
75     printk(KERN_INFO "tasklet: Module unloaded!\n");
76 }

```

1.2 Результаты работы

Рис. 1.1 – Сборка

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
# make
make -C /lib/modules/5.5.0-0.bpo.2-amd64/build M=/home/faris/Documents/Repositories
make[1]: Entering directory '/usr/src/linux-headers-5.5.0-0.bpo.2-amd64'
  CC [M]  /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01/t
Building modules, stage 2.
MODPOST 1 modules
  CC [M]  /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01/t
  LD [M]  /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01/t
make[1]: Leaving directory '/usr/src/linux-headers-5.5.0-0.bpo.2-amd64'

~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
#
```

Рис. 1.2 – Загрузка модуля ядра, вывод буфера сообщений ядра, проверка списка загруженных модулей

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
# sudo insmod ./tasklet.ko

~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
# sudo dmesg | tail -20
[90753.228321] tasklet: Module loaded!
[90753.325509] tasklet: scheduled
[90753.325542] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.454039] tasklet: scheduled
[90753.454071] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.454148] tasklet: scheduled
[90753.454182] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.578394] tasklet: scheduled
[90753.578424] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.578502] tasklet: scheduled
[90753.578506] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.652057] tasklet: scheduled
[90753.652153] tasklet: scheduled
[90753.652169] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.753807] tasklet: scheduled
[90753.753845] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.753930] tasklet: scheduled
[90753.753963] tasklet: state -- 2, count -- 0, data -- tasklet data
[90753.837404] tasklet: scheduled
[90753.837426] tasklet: state -- 2, count -- 0, data -- tasklet data

~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
# lsmod | grep tasklet
tasklet                16384  0
```

Рис. 1.3 – Просмотр содержимого /proc/interrupts

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?
# cat /proc/interrupts
```

| | CPU0 | CPU1 | CPU2 | CPU3 | CPU4 | CPU5 | CPU6 | CPU7 | | | |
|-----|---------|-------|------|------|------|------|------|------|------------|-------------|-----------------|
| 0: | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 2-edge | timer |
| 1: | 187644 | 0 | 0 | 0 | 0 | 0 | 0 | 1140 | IR-IO-APIC | 1-edge | i8042, ihandler |
| 8: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 8-edge | rtc0 |
| 9: | 1485226 | 52386 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 9-fastetoi | acpi |
| 14: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 14-fastetoi | INT3450:00 |

Рис. 1.4 – Выгрузка модуля ядра, вывод буфера сообщений ядра

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?  
# sudo rmmmod tasklet  
  
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task01 master ?  
# sudo dmesg | tail -5  
[90258.290646] tasklet: scheduled  
[90258.290647] tasklet: state -- 2, count -- 0, data -- tasklet data  
[90261.121080] tasklet: scheduled  
[90261.121100] tasklet: state -- 2, count -- 0, data -- tasklet data  
[90261.140862] tasklet: Module unloaded!
```

2 Очереди работ

2.1 Реализация

```
1 #include <linux/init.h>
2 #include <linux/module.h>
3 #include <linux/kernel.h>
4 #include <linux/interrupt.h>
5 #include <linux/workqueue.h>
6
7 #define HANDLED_IRQ 1
8
9 MODULE_LICENSE("GPL");
10 MODULE_AUTHOR("Faris Nabiev");
11
12 static int __init my_workqueue_init(void);
13 static void __exit my_workqueue_exit(void);
14
15 module_init(my_workqueue_init)
16 module_exit(my_workqueue_exit)
17
18 static int dev_id;
19 static int irq_call_cnt = 0;
20 // Очередь работ
21 static struct workqueue_struct *my_workqueue_struct;
22
23 // Функция обработки нижней половины
24 void my_workqueue_function(struct work_struct *work)
25 {
26     printk(KERN_INFO "workqueue: counter %d\n", ++irq_call_cnt);
27 }
28
29 DECLARE_WORK(my_workqueue, my_workqueue_function);
30
31 // Обработчик прерывания
32 static irqreturn_t ihandler(int irq, void *dev_id)
33 {
34     // Проверка, что произошло именно нужное прерывание
35     if (irq == HANDLED_IRQ)
36     {
37         queue_work(my_workqueue_struct, &my_workqueue);
38         printk(KERN_INFO "workqueue: in ihandler\n");
39
40         return IRQ_HANDLED; // Прерывание обработано
41     }
42     else
43         return IRQ_NONE; // Прерывание не обработано
```

```

44 }
45
46 static int __init my_workqueue_init(void)
47 {
48     // Инициализация модуля
49     if (request_irq(HANDLED_IRQ, ihandler,
50                     IRQF_SHARED, "ihandler", &dev_id))
51     {
52         printk(KERN_ERR "workqueue: Error on handler registering\n");
53
54         return -1;
55     }
56
57     // создание очереди работ
58     my_workqueue_struct = create_workqueue("my_workqueue");
59
60     if (!my_workqueue_struct)
61     {
62         free_irq(HANDLED_IRQ, &dev_id);
63         printk(KERN_ERR "workqueue: Error on workqueue creation\n");
64
65         return -ENOMEM;
66     }
67
68     printk(KERN_INFO "workqueue: workqueue created\n");
69     printk(KERN_INFO "workqueue: Module loaded!\n");
70
71     return 0;
72 }
73
74 // Выгрузка модуля
75 static void __exit my_workqueue_exit(void)
76 {
77     // Удаление очереди работ
78     flush_workqueue(my_workqueue_struct);
79     destroy_workqueue(my_workqueue_struct);
80     // Освобождение линии прерывания
81     free_irq(HANDLED_IRQ, &dev_id);
82
83     printk(KERN_INFO "workqueue: Module unloaded\n");
84 }

```

2.2 Результаты работы

Рис. 2.1 – Сборка

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?
# make
make -C /lib/modules/5.5.0-0.bpo.2-amd64/build M=/home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02
make[1]: Entering directory '/usr/src/linux-headers-5.5.0-0.bpo.2-amd64'
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02/workqueue.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02/workqueue.ko
LD [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02/workqueue.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.5.0-0.bpo.2-amd64'
```

Рис. 2.2 – Загрузка модуля ядра, вывод буфера сообщений ядра, проверка списка загруженных модулей

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?
# sudo insmod workqueue.ko

~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?
# sudo dmesg | tail -20
[91175.704739] workqueue: workqueue created
[91175.704740] workqueue: Module loaded!
[91175.744271] workqueue: in ihandler
[91175.744321] workqueue: counter 1
[91175.901603] workqueue: in ihandler
[91175.901713] workqueue: in ihandler
[91175.901736] workqueue: counter 2
[91176.020603] workqueue: in ihandler
[91176.020699] workqueue: in ihandler
[91176.020740] workqueue: counter 3
[91176.072574] workqueue: in ihandler
[91176.072613] workqueue: counter 4
[91176.072702] workqueue: in ihandler
[91176.072710] workqueue: counter 5
[91176.170081] workqueue: in ihandler
[91176.170095] workqueue: counter 6
[91176.170177] workqueue: in ihandler
[91176.170184] workqueue: counter 7
[91176.240341] workqueue: in ihandler
[91176.240358] workqueue: counter 8

~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?
# lsmod | grep workqueue
workqueue_                16384  0
```

Рис. 2.3 – Просмотр содержимого /proc/interrupts

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?
# cat /proc/interrupts
```

| | CPU0 | CPU1 | CPU2 | CPU3 | CPU4 | CPU5 | CPU6 | CPU7 | | | |
|-----|---------|-------|------|------|------|------|------|------|------------|------------|------------------|
| 0: | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 2-edge | timer |
| 1: | 192843 | 0 | 0 | 0 | 0 | 0 | 0 | 1140 | IR-IO-APIC | 1-edge | i8042, ihandler |
| 8: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 8-edge | rtc0 |
| 9: | 1501185 | 52386 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 9-fasteoi | acpi |
| 14: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 14-fasteoi | INT3450:00 |
| 16: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IR-IO-APIC | 16-fasteoi | idma64.0, i801_s |

Рис. 2.4 – Выгрузка модуля ядра, вывод буфера сообщений ядра

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_19/task02 master ?  
# sudo dmesg | tail -5  
[91248.256481] workqueue: in ihandler  
[91248.256504] workqueue: counter 284  
[91251.424290] workqueue: in ihandler  
[91251.424334] workqueue: counter 285  
[91251.460982] workqueue: Module unloaded
```