



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»

ОТЧЕТ

к лабораторной работе №9

По курсу: «Операционные системы»

На тему: «Обработчики прерываний»

Студентка ИУ7-65Б
Оберган Т.М

Преподаватель
Рязанова Н.Ю.

2020 г.

Оглавление

Часть 1 Тасклет.....	3
Листинг	3
Результат работы программы.....	5
Часть 2 Очередь работ	6
Листинг	6
Результат работы программы.....	7

Часть 1 Тасклет

Листинг

Содержимое makefile:

```
ifneq ($(KERNELRELEASE),)
    obj-m := tasklet.o
else
    CURRENT = $(shell uname -r)
    KDIR = /lib/modules/$(CURRENT)/build
    PWD = $(shell pwd)

default:
    $(MAKE) -C $(KDIR) M=$(PWD) modules
    make clean

clean:
    @rm -f *.o *.cmd *.flags *.mod.c *.order
    @rm -f *.*.cmd ~~ *.~~ TODO.*
    @rm -fR .tmp*
    @rm -rf .tmp_versions

disclean: clean
    @rm *.ko *.symvers
endif
```

Содержимое tasklet.c:

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/interrupt.h>

#define SHARED_IRQ 1

MODULE_LICENSE("Dual BSD/GPL");
MODULE_AUTHOR("Obergan T.M");

static int my_dev_id;
char tasklet_data[] = "LAB9 tasklet data";

void tasklet_handler(unsigned long data);

DECLARE_TASKLET(my_tasklet, tasklet_handler, (unsigned long) &tasklet_data);

void tasklet_handler(unsigned long data)
{
    printk(KERN_INFO "LAB9 Tasklet: state - %ld, count - %d, data - %s\n",
           my_tasklet.state, my_tasklet.count, my_tasklet.data);
}
```

```

// Обработчик прерывания
static irqreturn_t my_interrupt(int irq, void *dev_id)
{
    if (irq == SHARED_IRQ)
    {
        printk(KERN_INFO "LAB9 Tasklet scheduled\n");
        tasklet_schedule(&my_tasklet);
        return IRQ_HANDLED;
    }
    else
        return IRQ_NONE;
}

static int __init my_tasklet_init(void)
{
    if (request_irq(SHARED_IRQ, my_interrupt, IRQF_SHARED, "my_interrupt",
&my_dev_id))
    {
        printk(KERN_ERR "LAB9 Error on request_irq\n");
        return -1;
    }

    printk(KERN_INFO "Module loaded!\n");
    return 0;
}

static void __exit my_tasklet_exit(void)
{
    tasklet_kill(&my_tasklet);
    free_irq(SHARED_IRQ, &my_dev_id); // Освобождение линии прерывания
    printk(KERN_INFO "LAB9 Module unloaded!\n");
}

module_init(my_tasklet_init);
module_exit(my_tasklet_exit);

```

Результат работы программы

Загрузка модуля при помощи команды insmod:

```
[winterpuma@winterpuma ~]$ sudo insmod tasklet.ko
[sudo] password for winterpuma:
[winterpuma@winterpuma ~]$ lsmod | grep tasklet
tasklet                16384  0
[winterpuma@winterpuma ~]$ sudo dmesg | grep LAB9
[ 430.374305] LAB9 Tasklet scheduled
[ 430.374309] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 433.746846] LAB9 Tasklet scheduled
[ 433.746850] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 433.866244] LAB9 Tasklet scheduled
[ 433.866248] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 433.923517] LAB9 Tasklet scheduled
[ 433.923520] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 434.081206] LAB9 Tasklet scheduled
```

Просмотр содержимого файла /proc/interrupts:

```
[winterpuma@winterpuma ~]$ 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:         0         0         0         0         0         0         0     671 IR-IO-APIC  1-edge     i8042, my_interrupt
8:         0         0         0         0         0         1         0      0 IR-IO-APIC  8-edge     rtc0
9:         0         6         0         0         0         0         0      0 IR-IO-APIC  9-fasteoi  acpi
14:        0         0         1         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, i2c_designer
17:        0         0         0         0         502        0         0      0 IR-IO-APIC 17-fasteoi idma64.1, i2c_designer
20:        0         0         0         0         0         0         0      0 IR-IO-APIC 20-fasteoi idma64.2
120:       0         0         0         0         0         0         0      0 DWAR-MST  0-edge     dmar0
```

Выгрузка модуля при помощи команды rmmod:

```
[winterpuma@winterpuma ~]$ sudo rmmod tasklet
[winterpuma@winterpuma ~]$ sudo dmesg | grep LAB9 | tail -4
[ 542.124564] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 543.020419] LAB9 Tasklet scheduled
[ 543.020422] LAB9 Tasklet: state - 2, count - 0, data - LAB9 tasklet data
[ 543.031547] LAB9 Module unloaded!
[winterpuma@winterpuma ~]$
```

Часть 2 Очередь работ

Листинг

Содержимое my_workqueue.c:

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/interrupt.h>
#include <linux/workqueue.h>

#define SHARED_IRQ 1

MODULE_LICENSE("Dual BSD/GPL");
MODULE_AUTHOR("Obergan T.M");

static int my_dev_id;
static int irq_call_n = 0;
struct workqueue_struct *wq;

void my_workqueue_function(struct work_struct *work)
{
    printk(KERN_INFO "LAB9 workqueue: counter %d\n", ++irq_call_n);
}

DECLARE_WORK(my_workqueue, my_workqueue_function);

static irqreturn_t my_interrupt(int irq, void *dev_id)
{
    if (irq == SHARED_IRQ)
    {
        queue_work(wq, &my_workqueue);
        printk(KERN_INFO "LAB9 workqueue in my_interrupt\n");
        return IRQ_HANDLED;
    }
    else
        return IRQ_NONE;
}

static int __init my_workqueue_init(void)
{
    if (request_irq(SHARED_IRQ, my_interrupt, IRQF_SHARED, "my_interrupt",
&my_dev_id))
    {
        printk(KERN_ERR "LAB9 error: can't register handler\n");
        return -1;
    }

    wq = create_workqueue("workqueue");
    if (wq)
        printk(KERN_INFO "LAB9 Workqueue created!\n");
    else
    {
        free_irq(SHARED_IRQ, &my_dev_id);
        printk(KERN_ERR "LAB9 error: can't create workqueue\n");
        return -ENOMEM;
    }

    printk(KERN_INFO "LAB9 Module loaded!\n");

    return 0;
}
```

```
static void __exit my_workqueue_exit(void)
{
    flush_workqueue(wq);
    destroy_workqueue(wq);
    free_irq(SHARED_IRQ, &my_dev_id);
    printk(KERN_INFO "LAB9 Module unloaded\n");
}

module_init(my_workqueue_init)
module_exit(my_workqueue_exit)
```

Результат работы программы

Загрузка модуля при помощи команды insmod:

```
[winterpuma@winterpuma ~]$ sudo insmod my_workqueue.ko
[winterpuma@winterpuma ~]$ lsmod | grep my_workqueue
my_workqueue      16384  0
[winterpuma@winterpuma ~]$ dmesg | grep LAB9
dmesg: read kernel buffer failed: Operation not permitted
[winterpuma@winterpuma ~]$ sudo dmesg | grep LAB9
[ 1948.920680] LAB9 Workqueue created!
[ 1948.920681] LAB9 Module loaded!
[ 1949.062143] LAB9 workqueue in my_interrupt
[ 1949.062153] LAB9 workqueue: counter 1
[ 1951.014499] LAB9 workqueue in my_interrupt
[ 1951.014508] LAB9 workqueue: counter 2
[ 1951.102380] LAB9 workqueue in my_interrupt
[ 1951.102391] LAB9 workqueue: counter 3
```

Просмотр содержимого файла /proc/interrupts:

```
[winterpuma@winterpuma ~]$ 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:	0	0	0	0	0	0	0	2861	IR-IO-APIC	1-edge	i8042, my_interrupt
8:	0	0	0	0	0	1	0	0	IR-IO-APIC	8-edge	rtc0
9:	0	6	0	0	0	0	0	0	IR-IO-APIC	9-fasteoi	acpi
14:	0	0	1	0	0	0	0	0	IR-IO-APIC	14-fasteoi	INT3450:00

Выгрузка модуля при помощи команды rmmod:

```
[winterpuma@winterpuma ~]$ sudo rmmod my_workqueue
[winterpuma@winterpuma ~]$ sudo dmesg | grep LAB9 | tail -4
[ 2001.621119] LAB9 workqueue: counter 281
[ 2001.793276] LAB9 workqueue in my_interrupt
[ 2001.793286] LAB9 workqueue: counter 282
[ 2001.804096] LAB9 Module unloaded
[winterpuma@winterpuma ~]$
```