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

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 9

Дисциплина	Операционые системы.
Тема	Обработчики прерываний.
Студент	Степанов А. О.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Рязанова Н.Ю.

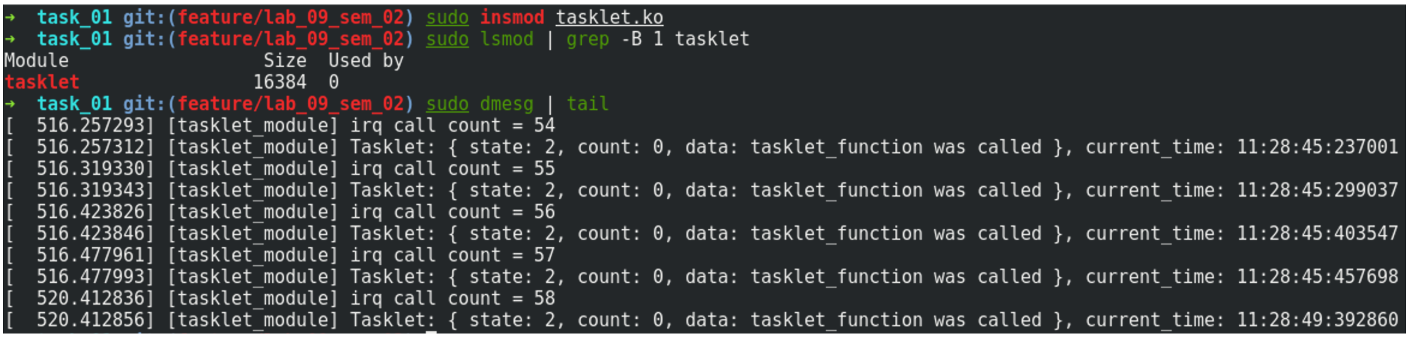
Москва, 2020 г.

ТАСКЛЕТЫ

Листинг 1: Текст программы

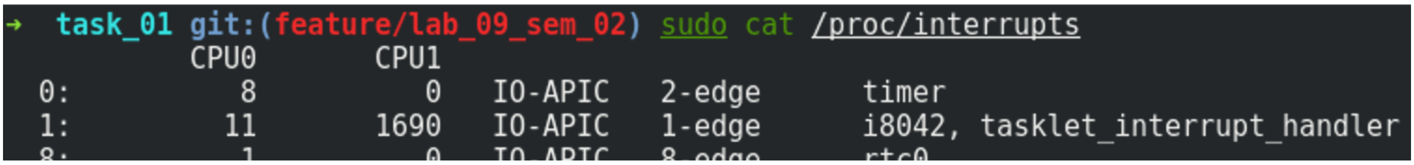
```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/interrupt.h>
#include <linux/time.h>

MODULE_AUTHOR("Alexander Stepanov");
MODULE_LICENSE("GPL");

static int irq = 1;
static int irq_call_count = 0;
static int dev_id;
char tasklet_data[] = "tasklet_function was called";

void tasklet_function(unsigned long data);

DECLARE_TASKLET(tasklet, tasklet_function, (unsigned long)&tasklet_data);

void tasklet_function(unsigned long data)
{
    struct timeval t;
    struct tm brocken;
    do_gettimeofday(&t);
    time_to_tm(t.tv_sec, 0, &brocken);

    printk(KERN_INFO
        "[tasklet_module] Tasklet: { state: %ld, count: %d, data: %s },"
        "current_time: %d:%d:%d:%ld\n",
        tasklet.state, atomic_read(&tasklet.count), (char *)tasklet.data,
        brocken.tm_hour + 3, brocken.tm_min, brocken.tm_sec, t.tv_usec);
}

static irqreturn_t interrupt_handler(int irq, void *dev_id)
{
    irq_call_count++;
    printk(KERN_INFO
    "[tasklet_module] irq call count = %d\n", irq_call_count);
    tasklet_schedule(&tasklet);
    return IRQ_NONE;
```

```c
}

static int __init tasklet_module_init(void)
{
    int ret = request_irq(
        irq, interrupt_handler, IRQF_SHARED,
        "tasklet_interrupt_handler", &dev_id
    );

    if (ret)
    {
        printk(KERN_ERR "[tasklet_module] error while handle irq\n");
        return -1;
    }

    printk(KERN_INFO "[tasklet_module] success load\n");
    return 0;
}

static void __exit tasklet_module_exit(void)
{
    tasklet_kill(&tasklet);
    free_irq(irq, &dev_id);
    printk(KERN_INFO "[tasklet_module] unload module\n");
}

module_init(tasklet_module_init);
module_exit(tasklet_module_exit);
```



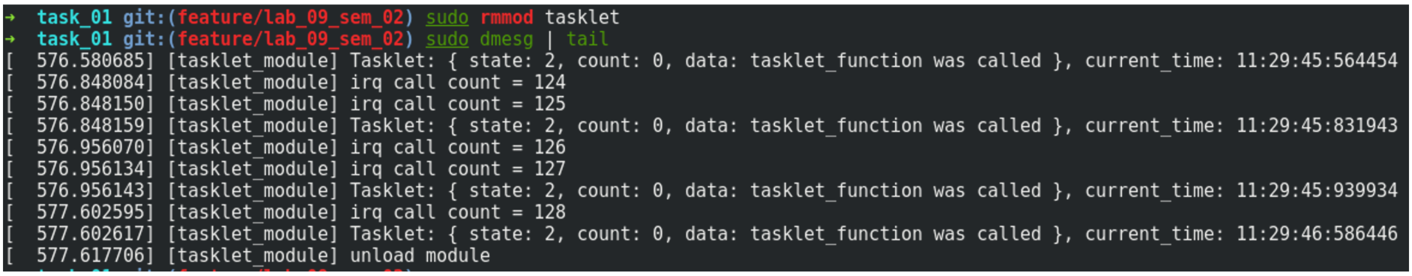
```
→  task_01 git:(feature/lab_09_sem_02) sudo insmod tasklet.ko
→  task_01 git:(feature/lab_09_sem_02) sudo lsmod | grep -B 1 tasklet
Module                  Size  Used by
tasklet                16384  0
→  task_01 git:(feature/lab_09_sem_02) sudo dmesg | tail
[  516.257293] [tasklet_module] irq call count = 54
[  516.257312] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:28:45:237001
[  516.319330] [tasklet_module] irq call count = 55
[  516.319343] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:28:45:299037
[  516.423826] [tasklet_module] irq call count = 56
[  516.423846] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:28:45:403547
[  516.477961] [tasklet_module] irq call count = 57
[  516.477993] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:28:45:457698
[  520.412836] [tasklet_module] irq call count = 58
[  520.412856] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:28:49:392860
```

Рис. 1: Загрузка модуля



```
→  task_01 git:(feature/lab_09_sem_02) sudo cat /proc/interrupts
            CPU0       CPU1
  0:           8          0   IO-APIC    2-edge      timer
  1:          11       1690   IO-APIC    1-edge      i8042, tasklet_interrupt_handler
  8:           1          0   IO-APIC    8-edge      rtc0
```

Рис. 2: Прерывания



```
→  task_01 git:(feature/lab_09_sem_02) sudo rmmod tasklet
→  task_01 git:(feature/lab_09_sem_02) sudo dmesg | tail
[  576.580685] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:29:45:564454
[  576.848084] [tasklet_module] irq call count = 124
[  576.848150] [tasklet_module] irq call count = 125
[  576.848159] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:29:45:831943
[  576.956070] [tasklet_module] irq call count = 126
[  576.956134] [tasklet_module] irq call count = 127
[  576.956143] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:29:45:939934
[  577.602595] [tasklet_module] irq call count = 128
[  577.602617] [tasklet_module] Tasklet: { state: 2, count: 0, data: tasklet_function was called }, current_time: 11:29:46:586446
[  577.617706] [tasklet_module] unload module
```

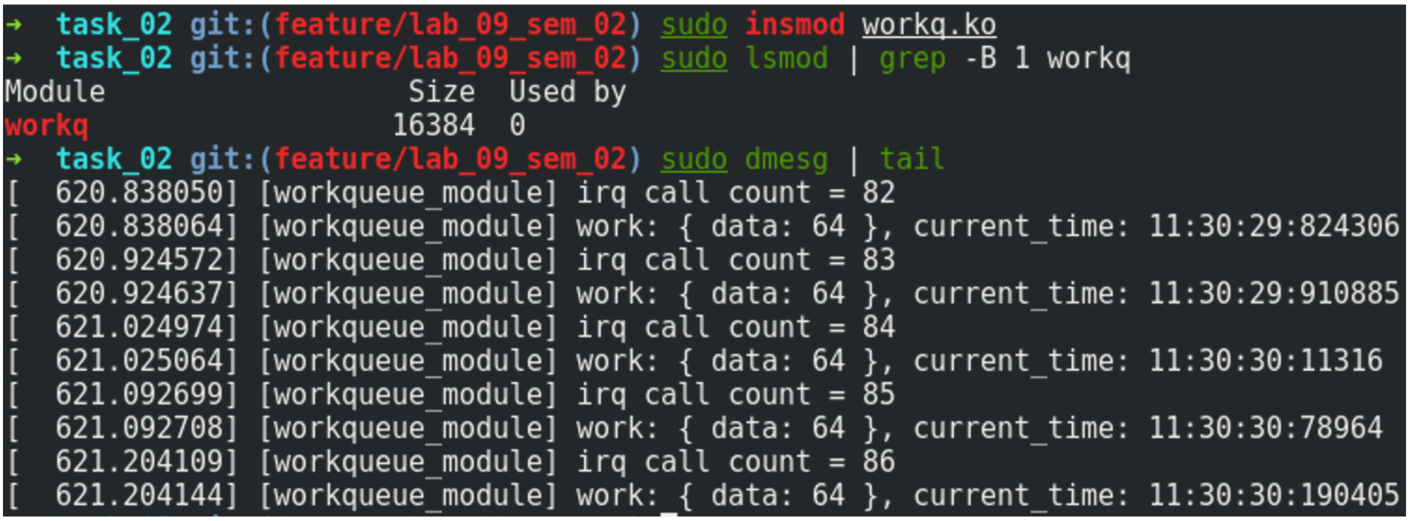
Рис. 3: Выгрузка модуля

ОЧЕРЕДИ РАБОТ

Листинг 2: Текст программы

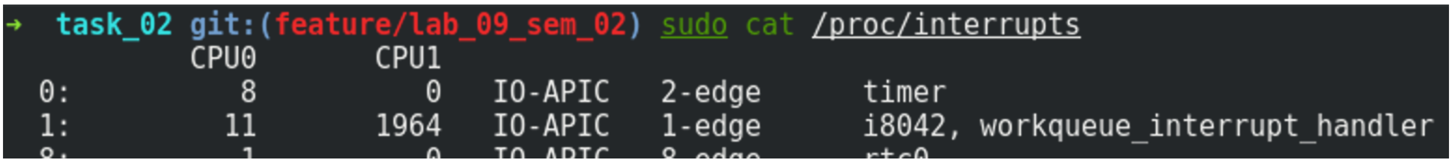
```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/interrupt.h>
#include <linux/workqueue.h>
#include <linux/time.h>

MODULE_AUTHOR("Alexander␣Stepanov");
MODULE_LICENSE("GPL");

static int irq = 1;
static int irq_call_count = 0;
static int dev_id;
static struct workqueue_struct *workq = NULL;

void work_function(struct work_struct *work)
{
    struct timeval t;
    struct tm brocken;
    do_gettimeofday(&t);
    time_to_tm(t.tv_sec, 0, &brocken);

    printk(KERN_INFO
        "[workqueue_module]␣work:␣{␣data:␣%ld␣},"
        "current_time:␣%d:%d:%d:%ld\n",
        atomic_long_read(&work->data),
        brocken.tm_hour + 3, brocken.tm_min, brocken.tm_sec, t.tv_usec);
}

DECLARE_WORK(work, work_function);

static irqreturn_t interrupt_handler(int irq, void *dev_id)
{
    irq_call_count++;
    queue_work(workq, &work);
    printk(KERN_INFO
        "[workqueue_module]␣irq␣call␣count␣=␣%d\n", irq_call_count);
    return IRQ_NONE;
}

static int __init workqueue_module_init(void)
{
    int ret = request_irq(
        irq, interrupt_handler, IRQF_SHARED,
        "workqueue_interrupt_handler", &dev_id
    );

    if (ret)
    {
        printk(KERN_ERR "[workqueue_module]␣error␣while␣handle␣irq\n");
        return -1;
    }

    workq = create_workqueue("workqueue");

    if (workq == NULL)
    {
        printk(KERN_ERR "[workqueue_module]␣error␣while␣create␣workqueue\n");
        return -1;
    }

    printk(KERN_INFO "[workqueue_module]␣success␣load\n");
    return 0;
}
```

```

static void __exit workqueue_module_exit(void)
{
    flush_workqueue(workq);
    destroy_workqueue(workq);
    free_irq(irq, &dev_id);
    printk(KERN_INFO "[workqueue_module] unload module\n");
}

module_init(workqueue_module_init);
module_exit(workqueue_module_exit);
```



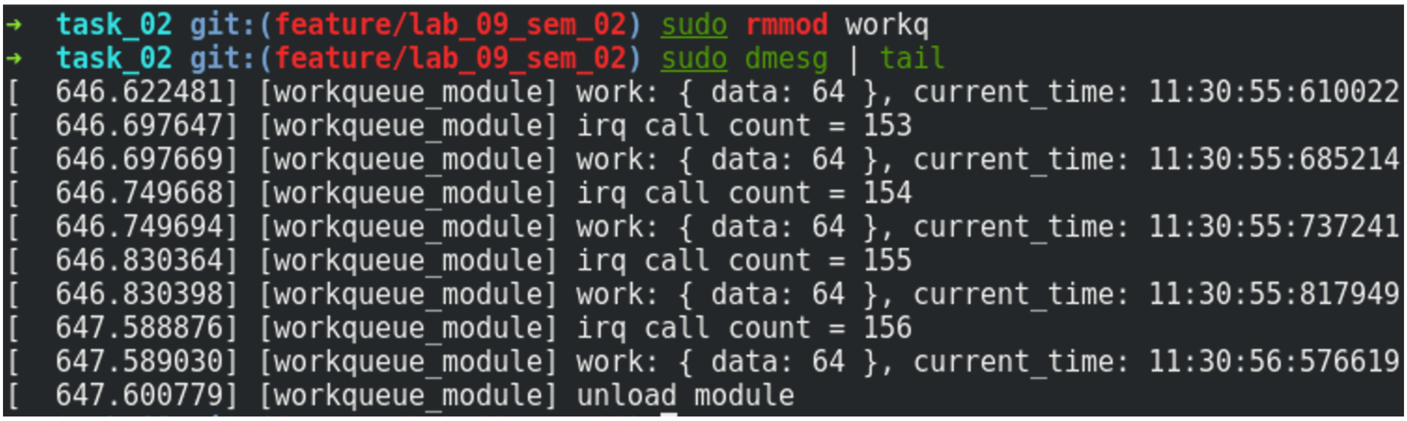
```
➜  task_02 git:(feature/lab_09_sem_02) sudo insmod workq.ko
➜  task_02 git:(feature/lab_09_sem_02) sudo lsmod | grep -B 1 workq
Module                  Size  Used by
workq                  16384  0
➜  task_02 git:(feature/lab_09_sem_02) sudo dmesg | tail
[  620.838050] [workqueue_module] irq call count = 82
[  620.838064] [workqueue_module] work: { data: 64 }, current_time: 11:30:29:824306
[  620.924572] [workqueue_module] irq call count = 83
[  620.924637] [workqueue_module] work: { data: 64 }, current_time: 11:30:29:910885
[  621.024974] [workqueue_module] irq call count = 84
[  621.025064] [workqueue_module] work: { data: 64 }, current_time: 11:30:30:11316
[  621.092699] [workqueue_module] irq call count = 85
[  621.092708] [workqueue_module] work: { data: 64 }, current_time: 11:30:30:78964
[  621.204109] [workqueue_module] irq call count = 86
[  621.204144] [workqueue_module] work: { data: 64 }, current_time: 11:30:30:190405
```

Рис. 4: Загрузка модуля



```
➜  task_02 git:(feature/lab_09_sem_02) sudo cat /proc/interrupts
           CPU0       CPU1
  0:          8          0   IO-APIC   2-edge      timer
  1:         11       1964   IO-APIC   1-edge      i8042, workqueue_interrupt_handler
  8:          1          0   IO-APIC   8-edge      rtc0
```

Рис. 5: Прерывания



```
→  task_02 git:(feature/lab_09_sem_02) sudo rmmod workq
→  task_02 git:(feature/lab_09_sem_02) sudo dmesg | tail
[  646.622481] [workqueue_module] work: { data: 64 }, current_time: 11:30:55:610022
[  646.697647] [workqueue_module] irq call count = 153
[  646.697669] [workqueue_module] work: { data: 64 }, current_time: 11:30:55:685214
[  646.749668] [workqueue_module] irq call count = 154
[  646.749694] [workqueue_module] work: { data: 64 }, current_time: 11:30:55:737241
[  646.830364] [workqueue_module] irq call count = 155
[  646.830398] [workqueue_module] work: { data: 64 }, current_time: 11:30:55:817949
[  647.588876] [workqueue_module] irq call count = 156
[  647.589030] [workqueue_module] work: { data: 64 }, current_time: 11:30:56:576619
[  647.600779] [workqueue_module] unload module
```

Рис. 6: Выгрузка модуля