|  |  |
| --- | --- |
|  | **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](#_Toc42116243)

[Листинг 3](#_Toc42116244)

[Результат работы программы 5](#_Toc42116245)

[Часть 2 Очередь работ 6](#_Toc42116246)

[Листинг 6](#_Toc42116247)

[Результат работы программы 7](#_Toc42116248)

# 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");

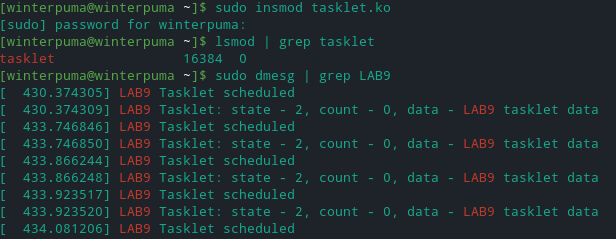
}

module\_init(my\_tasklet\_init);

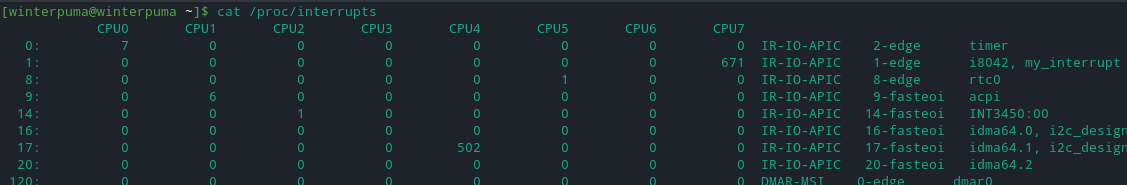
module\_exit(my\_tasklet\_exit);

## Результат работы программы

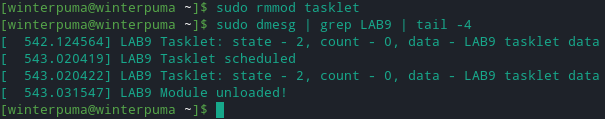
Загрузка модуля при помощи команды insmod:



Просмотр содержимого файла /proc/interrupts:



Выгрузка модуля при помощи команды rmmod:



# 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");

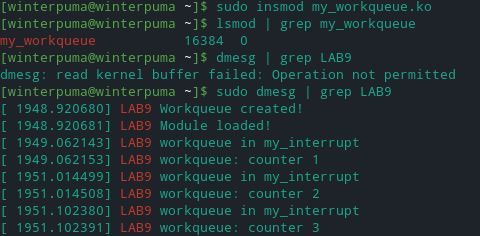
}

module\_init(my\_workqueue\_init)

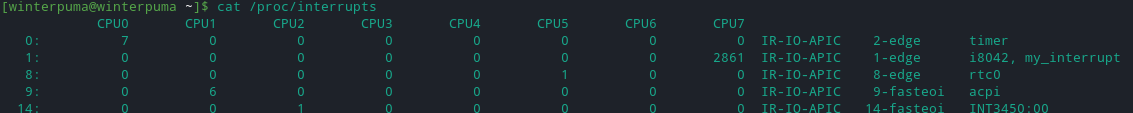
module\_exit(my\_workqueue\_exit)

## Результат работы программы

Загрузка модуля при помощи команды insmod:



Просмотр содержимого файла /proc/interrupts:



Выгрузка модуля при помощи команды rmmod:

