



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»

---

Дисциплина: «Операционные системы»

Лабораторная работа №9

Тема работы:  
«Обработчики прерываний»

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватель: Рязанова Н. Ю.

Москва, 2020 г.

## Задание (тасклет)

- 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 тасклете используя, или `printk()`, или `seq_file` interface - `<linux/seq_file.h>` (Jonathan Corber: <http://lwn.net/Articles/driver-porting/>).

## Листинг кода программы

```
1  #include <linux/module.h>
2  #include <linux/kernel.h>
3  #include <linux/init.h>
4  #include <linux/interrupt.h>
5  #include <linux/timex.h>
6
7  MODULE_LICENSE("GPL");
8  MODULE_AUTHOR("Ilya Levushkin");
9
10 #define SHARED_IRQ 1
11 static int irq = SHARED_IRQ, my_dev_id, irq_counter = 0;
12
13 module_param(irq, int, S_IRUGO);
14
15 char my_tasklet_data[] = "my_tasklet_function was called";
16
17 void my_tasklet_function(unsigned long data);
18
19 DECLARE_TASKLET(my_tasklet, my_tasklet_function,
20                 (unsigned long)&my_tasklet_data);
21
22
23
24 void my_tasklet_function(unsigned long data)
25 {
26     printk(KERN_INFO "[TASKLET] state: %ld, count: %d, data: %ld",
27            my_tasklet.state, my_tasklet.count, my_tasklet.data);
28     return;
29 }
30
31 static irqreturn_t my_interrupt( int irq, void *dev_id)
32 {
33     if (irq == SHARED_IRQ)
34     {
35         irq_counter++;
36         printk(KERN_INFO "[INTERRUPT] In the ISR: counter = %d\n", irq_counter);
37         tasklet_schedule(&my_tasklet);
38         return IRQ_HANDLED;
39     }
40     else
41         return IRQ_NONE;
42 }
43
```

```

38
39 static int __init my_tasklet_init(void)
40 {
41     if (request_irq(irq, my_interrupt, IRQF_SHARED, "my_interrupt", &my_dev_id))
42         return -1;
43     printk(KERN_INFO
44         "[INTERRUPT] Successfully loading ISR handler on IRQ %d\n", irq);
45     printk(KERN_INFO "[MODULE] Module is now loaded.\n");
46     return 0;
47 }
48
49 static void __exit my_tasklet_exit(void)
50 {
51     tasklet_kill(&my_tasklet);
52     synchronize_irq(irq);
53     free_irq(irq, &my_dev_id);
54     printk(KERN_INFO
55         "[INTERRUPT] Successfully unloading, irq_counter = %d\n", irq_counter);
56     printk(KERN_INFO "[MODULE] Module is now unloaded.\n");
57     return;
58 }
59
60 module_init(my_tasklet_init);
61 module_exit(my_tasklet_exit);
62

```

## Демонстрация работы программы

Загрузка модуля ядра «tasklet» с параметром `irq = 1` (номер прерывания) и последние 50 сообщений, выведенных модулями ядра (на рисунке показаны последние 17).

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ sudo insmod tasklet.ko irq=1
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ dmesg | tail -50
```

```
[ 2714.605825] [INTERRUPT] Successfully loading ISR handler on IRQ 1
[ 2714.605827] [MODULE] Module is now loaded.
[ 2714.650912] [INTERRUPT] In the ISR: counter = 1
[ 2714.650928] [TASKLET] state: 2, count: 0, data: -1067450320
[ 2715.083109] [INTERRUPT] In the ISR: counter = 2
[ 2715.083206] [INTERRUPT] In the ISR: counter = 3
[ 2715.083215] [TASKLET] state: 2, count: 0, data: -1067450320
[ 2715.179019] [INTERRUPT] In the ISR: counter = 4
[ 2715.179115] [INTERRUPT] In the ISR: counter = 5
[ 2715.179123] [TASKLET] state: 2, count: 0, data: -1067450320
[ 2715.251026] [INTERRUPT] In the ISR: counter = 6
[ 2715.251119] [INTERRUPT] In the ISR: counter = 7
[ 2715.251129] [TASKLET] state: 2, count: 0, data: -1067450320
[ 2715.346923] [INTERRUPT] In the ISR: counter = 8
[ 2715.347084] [INTERRUPT] In the ISR: counter = 9
[ 2715.347094] [TASKLET] state: 2, count: 0, data: -1067450320
[ 2716.082966] [INTERRUPT] In the ISR: counter = 10
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ lsmod | grep tasklet
tasklet                16384  0
```

Вывод списка загруженных модулей ядра, чье название содержит строку «tasklet»

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ lsmod | grep tasklet
tasklet                16384  0
```

Установленный обработчик 1-ого прерывания в системе:

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ cat /proc/interrupts
```

	CPU0	CPU1			
0:	2	0	IO-APIC	2-edge	timer
1:	304	0	IO-APIC	1-edge	i8042, my_interrupt
8:	0	1	IO-APIC	8-edge	rtc0
9:	0	9349	IO-APIC	9-fasteoi	acpi
12:	0	143	IO-APIC	12-edge	i8042
14:	0	0	IO-APIC	14-edge	ata_piix
15:	0	0	IO-APIC	15-edge	ata_piix
18:	2175	1645	IO-APIC	18-fasteoi	uhci_hcd:usb2
19:	67	0	IO-APIC	19-fasteoi	ehci_hcd:usb1
22:	0	0	IO-APIC	22-fasteoi	virtio1
25:	31	0	PCI-MSI	487424-edge	xhci_hcd
26:	2320	0	PCI-MSI	49152-edge	prl_tg
27:	0	4	PCI-MSI	81920-edge	virtio0-config
28:	1612	5	PCI-MSI	81921-edge	virtio0-input.0
29:	1520	149	PCI-MSI	81922-edge	virtio0-output.0
30:	17009	7518	PCI-MSI	512000-edge	ahci[0000:00:1f.2]
31:	9234	561	PCI-MSI	524288-edge	prl_drm
32:	1568	0	PCI-MSI	516096-edge	snd_hda_intel:card0
NMI:	0	0	Non-maskable interrupts		
LOC:	66056	75359	Local timer interrupts		
SPU:	0	0	Spurious interrupts		
PMI:	0	0	Performance monitoring interrupts		
IWI:	94	0	IRQ work interrupts		
RTR:	0	0	APIC ICR read retries		
RES:	177039	174423	Rescheduling interrupts		
CAL:	2294	2257	Function call interrupts		
TLB:	5076	5709	TLB shutdowns		
TRM:	0	0	Thermal event interrupts		
THR:	0	0	Threshold APIC interrupts		
DFR:	0	0	Deferred Error APIC interrupts		
MCE:	0	0	Machine check exceptions		
MCP:	5	5	Machine check polls		
HYP:	0	0	Hypervisor callback interrupts		
HRE:	0	0	Hyper-V reenlightenment interrupts		
HVS:	0	0	Hyper-V stimer0 interrupts		
ERR:	0				
MIS:	0				
PIN:	0	0	Posted-interrupt notification event		
NPI:	0	0	Nested posted-interrupt event		
PIW:	0	0	Posted-interrupt wakeup event		

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$
```

Выгрузка модуля ядра «tasklet», последние два сообщения, выведенные модулями ядра.

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ sudo rmmod tasklet
```

```
[ 2728.741453] [INTERRUPT] Successfully unloading, irq_counter = 54  
[ 2728.741455] [MODULE] Module is now unloaded.
```

Вывод списка загруженных модулей ядра, чье название содержит строку «tasklet»

```
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$ lsmod | grep tasklet  
ilalevuskin@ubuntu:~/Desktop/9_os/tasklet$
```

Видно, что модуль ядра «tasklet» успешно выгружен.

## Задание (очередь работ)

- Написать загружаемый модуль ядра, в котором зарегистрировать обработчик аппаратного прерывания с флагом `IRQF_SHARED`.
- Инициализировать очередь работ.
- В обработчике прерывания запланировать очередь работ на выполнение.
- Вывести информацию об очереди работ используя, или `printk()`, или `seq_file` interface - `<linux/seq_file.h>` (Jonathan Corber: <http://lwn.net//Articles//driver-porting/>).

## Листинг кода программы

```
1  #include <linux/module.h>
2  #include <linux/kernel.h>
3  #include <linux/init.h>
4  #include <linux/interrupt.h>
5  #include <linux/workqueue.h>
6
7  MODULE_LICENSE("GPL");
8  MODULE_AUTHOR("Ilya Levushkin");
9
10 #define SHARED_IRQ 1
11 static int irq = SHARED_IRQ, my_dev_id, irq_counter = 0;
12 module_param( irq, int, S_IRUGO );
13
14 struct workqueue_struct *my_wq;
15 void hardwork_function(struct work_struct *work);
16
17 DECLARE_WORK(hardwork, hardwork_function);
18
19
20 void hardwork_function(struct work_struct *work)
21 {
22     printk(KERN_INFO "[WQ] data: %d", work->data);
23     return;
24 }
25
26 static irqreturn_t my_interrupt(int irq, void *dev_id)
27 {
28     if (irq == SHARED_IRQ) {
29         irq_counter++;
30         printk(KERN_INFO "[INTERRUPT] In the ISR: counter = %d\n", irq_counter);
31         queue_work(my_wq, &hardwork);
32         return IRQ_HANDLED;
33     }
34     else return IRQ_NONE;
35 }
36
37 static int __init my_workqueue_init(void)
38 {
39     if(request_irq(irq, my_interrupt, IRQF_SHARED, "my_interrupt2B", &my_dev_id))
40         return -1;
41
42     printk(KERN_INFO
43         "[INTERRUPT] Successfully loading ISR handler on IRQ %d\n", irq);
```



```

42
43     if (my_wq = create_workqueue("my_queue"))
44     {
45         printk(KERN_INFO "[MODULE] Workqueue created.\n");
46     }
47
48     printk(KERN_INFO "[MODULE] Module is now loaded.\n");
49     return 0;
50 }
51
52 static void __exit my_workqueue_exit(void)
53 {
54     flush_workqueue(my_wq);
55     destroy_workqueue(my_wq);
56     synchronize_irq(irq);
57     free_irq(irq, &my_dev_id);
58     printk(KERN_INFO
59         "[INTERRUPT] Successfully unloading, irq_counter = %d\n", irq_counter);
60     printk(KERN_INFO "[MODULE] Module is now unloaded.\n");
61     return;
62 }
63
64 module_init(my_workqueue_init);
65 module_exit(my_workqueue_exit);
66

```

## Демонстрация работы программы

Загрузка модуля ядра «queue» с параметром `irq = 1` (номер прерывания) и последние 50 сообщений, выведенных модулями ядра (на рисунке показаны последние 30).

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ sudo insmod queue.ko irq=1
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ dmesg | tail -50
```

```
[ 2776.288127] [INTERRUPT] Successfully loading ISR handler on IRQ 1
[ 2776.290819] [MODULE] Workqueue created.
[ 2776.290820] [MODULE] Module is now loaded.
[ 2776.367691] [INTERRUPT] In the ISR: counter = 1
[ 2776.367741] [WQ] data: 0
[ 2777.224078] [INTERRUPT] In the ISR: counter = 2
[ 2777.224177] [INTERRUPT] In the ISR: counter = 3
[ 2777.224198] [WQ] data: 0
[ 2777.335620] [INTERRUPT] In the ISR: counter = 4
[ 2777.335669] [INTERRUPT] In the ISR: counter = 5
[ 2777.335685] [WQ] data: 0
[ 2777.528213] [INTERRUPT] In the ISR: counter = 6
[ 2777.528311] [INTERRUPT] In the ISR: counter = 7
[ 2777.528332] [WQ] data: 0
[ 2777.640036] [INTERRUPT] In the ISR: counter = 8
[ 2777.640118] [INTERRUPT] In the ISR: counter = 9
[ 2777.640404] [WQ] data: 0
[ 2777.960007] [INTERRUPT] In the ISR: counter = 10
[ 2777.960103] [INTERRUPT] In the ISR: counter = 11
[ 2777.960124] [WQ] data: 0
[ 2778.072180] [INTERRUPT] In the ISR: counter = 12
[ 2778.072280] [INTERRUPT] In the ISR: counter = 13
[ 2778.072310] [WQ] data: 0
[ 2778.808086] [INTERRUPT] In the ISR: counter = 14
[ 2778.808187] [INTERRUPT] In the ISR: counter = 15
[ 2778.808469] [WQ] data: 0
[ 2778.927765] [INTERRUPT] In the ISR: counter = 16
[ 2778.927841] [INTERRUPT] In the ISR: counter = 17
[ 2778.927858] [WQ] data: 0
[ 2779.576069] [INTERRUPT] In the ISR: counter = 18
```

Вывод списка загруженных модулей ядра, чье название содержит строку «queue»

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ lsmod | grep queue
queue                16384  0
```

Установленный обработчик 1-ого прерывания в системе:

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ cat /proc/interrupts
          CPU0           CPU1
 0:         2             0   IO-APIC  2-edge       timer
 1:       596             0   IO-APIC  1-edge       i8042, my_interrupt2
 8:         0             1   IO-APIC  8-edge       rtc0
 9:         0          9935   IO-APIC  9-fasteoi    acpi
12:         0          143   IO-APIC 12-edge       i8042
14:         0             0   IO-APIC 14-edge       ata_piix
15:         0             0   IO-APIC 15-edge       ata_piix
18:      2924          4249   IO-APIC 18-fasteoi    uhci_hcd:usb2
19:         67             0   IO-APIC 19-fasteoi    ehci_hcd:usb1
22:         0             0   IO-APIC 22-fasteoi    virtio1
25:         31             0   PCI-MSI 487424-edge    xhci_hcd
26:      2468             0   PCI-MSI 49152-edge    prl_tg
27:         0             4   PCI-MSI 81920-edge    virtio0-config
28:      1613             9   PCI-MSI 81921-edge    virtio0-input.0
29:      1527          149   PCI-MSI 81922-edge    virtio0-output.0
30:     17219          7817   PCI-MSI 512000-edge    ahci[0000:00:1f.2]
31:     14145          561   PCI-MSI 524288-edge    prl_drm
32:      1568             0   PCI-MSI 516096-edge    snd_hda_intel:card0
NMI:         0             0   Non-maskable interrupts
LOC:     75259          86818   Local timer interrupts
SPU:         0             0   Spurious interrupts
PMI:         0             0   Performance monitoring interrupts
IWI:        149             0   IRQ work interrupts
RTR:         0             0   APIC ICR read retries
RES:     221928        218010   Rescheduling interrupts
CAL:       2446          2290   Function call interrupts
TLB:       6133          6588   TLB shutdowns
TRM:         0             0   Thermal event interrupts
THR:         0             0   Threshold APIC interrupts
DFR:         0             0   Deferred Error APIC interrupts
MCE:         0             0   Machine check exceptions
MCP:         5             5   Machine check polls
HYP:         0             0   Hypervisor callback interrupts
HRE:         0             0   Hyper-V reenlightenment interrupts
HVS:         0             0   Hyper-V stimer0 interrupts
ERR:         0
MIS:         0
PIN:         0             0   Posted-interrupt notification event
NPI:         0             0   Nested posted-interrupt event
PIW:         0             0   Posted-interrupt wakeup event
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$
```

Выгрузка модуля ядра «queue», последние два сообщения, выведенные модулями ядра.

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ sudo rmmod queue
```

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ dmesg | tail -2
[ 2809.522301] [INTERRUPT] Successfully unloading, irq_counter = 152
[ 2809.522302] [MODULE] Module is now unloaded.
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$
```

Вывод списка загруженных модулей ядра, чье название содержит строку «queue»

```
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$ lsmod | grep queue
ilalevuskin@ubuntu:~/Desktop/9_os/workqueue$
```

Видно, что модуль ядра «queue» успешно выгружен.