



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 № 3

Дисциплина	Операционные системы.
Тема	Загружаемые модули ядра.
Студент	Сиденко А.Г.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Рязанова Н.Ю.

Москва, 2020 г.

## Пример простого загружаемого модуля ядра

```
1 #include <linux/init.h>
2 #include <linux/module.h>
3 #include <linux/kernel.h>
4
5 MODULE_LICENSE("GPL");
6 MODULE_AUTHOR("Sidenko");
7 MODULE_DESCRIPTION("Lab3");
8
9 // Инициализация модуля
10 static int __init my_module_init(void)
11 {
12     printk(KERN_INFO "MODULE1: _loaded!\n");
13
14     struct task_struct *task = &init_task;
15     do
16     {
17         printk(KERN_INFO "MODULE1: _process: _%s _%d, _parent: _%s _%d\n",
18             task->comm, task->pid, task->parent->comm, task->parent->pid);
19     }
20     while ((task = next_task(task)) != &init_task);
21
22     printk(KERN_INFO "MODULE1: _current: _%s _%d, _parent: _%s _%d\n",
23         current->comm, current->pid, current->parent->comm, current->parent->pid);
24     return 0;
25 }
26
27 // Удаление модуля
28 static void __exit my_module_exit(void)
29 {
30     printk(KERN_INFO "MODULE1: _unloaded\n");
31 }
32
33 module_init(my_module_init);
34 module_exit(my_module_exit);
```

### 1. Загрузка модуля ядра:

```
+ [~/Desktop/task1] $ sudo insmod md.ko
[sudo] password for parallels:
```

```
+ [~/Desktop/task1] $ lsmod | grep md
md                16384 0
```

### 2. Вывод буфера сообщений ядра в стандартный поток вывода:

```

+ ~/Desktop/task1 sudo dmesg | grep MODULE1
[ 102.862336] MODULE1: loaded!
[ 102.862338] MODULE1: process: swapper/0 - 0, parent: swapper/0 - 0
[ 102.862339] MODULE1: process: systemd - 1, parent: swapper/0 - 0
[ 102.862340] MODULE1: process: kthreadd - 2, parent: swapper/0 - 0
[ 102.862341] MODULE1: process: ksoftirqd/0 - 3, parent: kthreadd - 2
[ 102.862342] MODULE1: process: kworker/0:0 - 4, parent: kthreadd - 2
[ 102.862343] MODULE1: process: kworker/0:0H - 5, parent: kthreadd - 2
[ 102.862344] MODULE1: process: kworker/u64:0 - 6, parent: kthreadd - 2
[ 102.862349] MODULE1: process: rcu_sched - 7, parent: kthreadd - 2
[ 102.862350] MODULE1: process: rcu_bh - 8, parent: kthreadd - 2
[ 102.862351] MODULE1: process: migration/0 - 9, parent: kthreadd - 2
[ 102.862352] MODULE1: process: lru-add-drain - 10, parent: kthreadd - 2
[ 102.862353] MODULE1: process: watchdog/0 - 11, parent: kthreadd - 2
[ 102.862354] MODULE1: process: cpuhp/0 - 12, parent: kthreadd - 2
[ 102.862355] MODULE1: process: cpuhp/1 - 13, parent: kthreadd - 2
[ 102.862356] MODULE1: process: watchdog/1 - 14, parent: kthreadd - 2
[ 102.862357] MODULE1: process: migration/1 - 15, parent: kthreadd - 2
[ 102.862358] MODULE1: process: ksoftirqd/1 - 16, parent: kthreadd - 2
[ 102.862359] MODULE1: process: kworker/1:0 - 17, parent: kthreadd - 2
[ 102.862360] MODULE1: process: kworker/1:0H - 18, parent: kthreadd - 2
[ 102.862361] MODULE1: process: kworker/1:0H - 18, parent: kthreadd - 2
[ 102.862536] MODULE1: process: tracker-store - 2149, parent: systemd - 1894
[ 102.862537] MODULE1: process: tracker-miner-a - 2151, parent: gnome-session-b - 1925
[ 102.862538] MODULE1: process: tracker-miner-u - 2158, parent: gnome-session-b - 1925
[ 102.862540] MODULE1: process: prlnd - 2193, parent: prlcc - 2139
[ 102.862541] MODULE1: process: prlcp - 2194, parent: prlcc - 2139
[ 102.862542] MODULE1: process: prlsga - 2195, parent: prlcc - 2139
[ 102.862543] MODULE1: process: prlshprof - 2196, parent: prlcc - 2139
[ 102.862545] MODULE1: process: evolution-calen - 2204, parent: systemd - 1894
[ 102.862546] MODULE1: process: evolution-calen - 2305, parent: evolution-calen - 2204
[ 102.862547] MODULE1: process: evolution-addre - 2330, parent: systemd - 1894
[ 102.862548] MODULE1: process: evolution-calen - 2333, parent: evolution-calen - 2204
[ 102.862549] MODULE1: process: evolution-addre - 2365, parent: evolution-addre - 2330
[ 102.862550] MODULE1: process: gvfsd-metadata - 2400, parent: systemd - 1894
[ 102.862552] MODULE1: process: nautilus - 2469, parent: systemd - 1894
[ 102.862553] MODULE1: process: gvfsd-trash - 2473, parent: systemd - 1894
[ 102.862554] MODULE1: process: gvfsd-burn - 2491, parent: systemd - 1894
[ 102.862555] MODULE1: process: gnome-terminal- - 2505, parent: systemd - 1894
[ 102.862556] MODULE1: process: zsh - 2514, parent: gnome-terminal- - 2505
[ 102.862557] MODULE1: process: sudo - 3125, parent: zsh - 2514
[ 102.862558] MODULE1: process: insmod - 3138, parent: sudo - 3125
[ 102.862559] MODULE1: current: insmod - 3138, parent: sudo - 3125

```

### 3. Выгрузка модуля ядра:

```

+ ~/Desktop/task1 sudo dmesg | tail -2
[ 102.862559] MODULE1: current: insmod - 3138, parent: sudo - 3125
[ 413.061674] MODULE1: unloaded
+ ~/Desktop/task1

```

## Модуль md1

Модуль md1 демонстрирует возможность создания экспортируемых данных и функций.

```
1 #include <linux/init.h>
2 #include <linux/module.h>
3 #include "md.h"
4
5 MODULE_LICENSE("GPL");
6 MODULE_AUTHOR("Sidenko");
7 MODULE_DESCRIPTION("Lab3");
8
9 char* md1_str_data = "Привет_из_модуля_1!";
10 int md1_int_data = 111;
11
12 extern char* md1_get_str(int n)
13 {
14     printk(KERN_INFO "MODULE1: _md1_get_str(%d) _called\n", n);
15     switch (n)
16     {
17     case 1:
18         return "Привет!";
19         break;
20     case 2:
21         return "Пока!";
22         break;
23     default:
24         return "Передайте_1_для_приветствия_или_2_для_прощания";
25         break;
26     }
27 }
28
29 extern int md1_factorial(int n)
30 {
31     printk(KERN_INFO "MODULE1: _md1_factorial(%d) _called\n", n);
32
33     int i, answer = 1;
34
35     if (n <= 0)
36         return 0;
37
38     for (i = 2; i <= n; i++)
39         answer *= i;
40
41     return answer;
42 }
43
44 // экспортируем данные
45 EXPORT_SYMBOL(md1_str_data);
46 EXPORT_SYMBOL(md1_int_data);
47 // экспортируем функции
48 EXPORT_SYMBOL(md1_get_str);
```

```

49 EXPORT_SYMBOL(md1_factorial);
50
51 static int __init my_module_init(void)
52 {
53     printk(KERN_INFO "MODULE1: _loaded\n");
54     return 0;
55 }
56
57 static void __exit my_module_exit(void)
58 {
59     printk(KERN_INFO "MODULE1: _unloaded\n");
60 }
61
62 module_init(my_module_init);
63 module_exit(my_module_exit);

```

## Модуль md2

Модуль md2 демонстрирует использование данных и функций экспортируемых первым модулем (md1).

```

1  #include <linux/init.h>
2  #include <linux/module.h>
3  #include "md.h"
4
5  MODULE_LICENSE("GPL");
6  MODULE_AUTHOR("Sidenko");
7  MODULE_DESCRIPTION("Lab3");
8
9  static int __init my_module_init(void)
10 {
11     printk(KERN_INFO "MODULE2: _loaded\n");
12     printk(KERN_INFO
13         "MODULE2: _Число_экспортированное_из_md1_: %d\n", md1_int_data);
14     printk(KERN_INFO
15         "MODULE2: _Строка_экспортированная_из_md1_: %s\n", md1_str_data);
16     printk(KERN_INFO
17         "MODULE2: _Результат_работы_функции_md1_get_str(2)_: %s\n",
18             md1_get_str(10));
19     printk(KERN_INFO
20         "MODULE2: _Результат_работы_функции_md1_get_str(0)_: %s\n",
21             md1_get_str(1));
22     printk(KERN_INFO
23         "MODULE2: _Результат_работы_функции_md1_get_str(1)_: %s\n",
24             md1_get_str(2));
25     printk(KERN_INFO
26         "MODULE2: _Результат_работы_функции_md1_factorial(4)_: %d\n",
27             md1_factorial(10));
28     return 0;
29 }
30
31 static void __exit my_module_exit(void)
32 {

```

```

33 | printk(KERN_INFO "MODULE2: _unloaded\n");
34 | }
35 |
36 | module_init(my_module_init);
37 | module_exit(my_module_exit);

```

## Модуль md3

Модуль md3 демонстрирует сценарий некорректного завершения установки модуля, и возможность использования загружаемого модуля в качестве функции выполняемой в пространстве ядра.

```

1 | #include <linux/init.h>
2 | #include <linux/module.h>
3 | #include "md.h"
4 |
5 | MODULE_LICENSE("GPL");
6 | MODULE_AUTHOR("Sidenko");
7 | MODULE_DESCRIPTION("Lab3");
8 |
9 | static int __init my_module_init(void)
10 | {
11 |     printk(KERN_INFO "MODULE3: _loaded\n");
12 |     printk(KERN_INFO
13 |         "MODULE3: _Число_экспортированное_из_md1_: %d\n", md1_int_data);
14 |     printk(KERN_INFO
15 |         "MODULE3: _Строка_экспортированная_из_md1_: %s\n", md1_str_data);
16 |     printk(KERN_INFO
17 |         "MODULE3: _Результат_работы_функции_md1_get_str(2): %s\n",
18 |             md1_get_str(10));
19 |     printk(KERN_INFO
20 |         "MODULE3: _Результат_работы_функции_md1_get_str(0): %s\n",
21 |             md1_get_str(1));
22 |     printk(KERN_INFO
23 |         "MODULE3: _Результат_работы_функции_md1_get_str(1): %s\n",
24 |             md1_get_str(2));
25 |     printk(KERN_INFO
26 |         "MODULE3: _Результат_работы_функции_md1_factorial(4): %d\n",
27 |             md1_factorial(10));
28 |     return -1;
29 | }
30 |
31 | module_init(my_module_init);

```

1. Необходимо загружать модули в правильном порядке, сначала md1, затем md2. Иначе возникнет ошибка, потому что модуль md2 содержит ссылки на неизвестные ядру имена (хотя они определены в другом модуле md1):

```

+ [~/Desktop/task2] sudo insmod md2.ko
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
+ [~/Desktop/task2] sudo dmesg | tail -4
[18996.118782] md2: Unknown symbol md1_int_data (err 0)
[18996.118799] md2: Unknown symbol md1_factorial (err 0)
[18996.118814] md2: Unknown symbol md1_get_str (err 0)
[18996.118828] md2: Unknown symbol md1_str_data (err 0)
+ [~/Desktop/task2]
+ [~/Desktop/task2] sudo insmod md1.ko
sudo] password for parallels:
+ [~/Desktop/task2] sudo insmod md2.ko

```

2. Вывод буфера сообщений ядра в стандартный поток вывода:

```

+ [~/Desktop/task2] sudo dmesg | tail -12
[18396.612085] MODULE1: loaded
[18407.283326] MODULE2: loaded
[18407.283328] MODULE2: Число экспортированное из md1 : 111
[18407.283329] MODULE2: Строка экспортированная из md1 : Привет из модуля 1!
[18407.283330] MODULE1: md1_get_str(10) called
[18407.283330] MODULE2: Результат работы функции md1_get_str(10) : Передайте 1 для приветствия или 2 для прощания
[18407.283331] MODULE1: md1_get_str(1) called
[18407.283331] MODULE2: Результат работы функции md1_get_str(1) : Привет!
[18407.283332] MODULE1: md1_get_str(2) called
[18407.283332] MODULE2: Результат работы функции md1_get_str(2) : Пока!
[18407.283333] MODULE1: md1_factorial(10) called
[18407.283334] MODULE2: Результат работы функции md1_factorial(10) : 3628800

```

3. А выгружаем модули в обратном порядке, так как на модуль md1 ссылается некоторые другие модули или объекты ядра и до тех пор, пока число ссылок модуль в системе не станет нулевым, модуль не может быть выгружен. Модуль md2 ссылается на md1, поэтому сначала необходимо выгрузить его:

```

+ [~/Desktop/task2] sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2
+ [~/Desktop/task2] lsmod | grep md
md2                16384  0
md1                 16384  1 md2
+ [~/Desktop/task2] sudo rmmod md2
+ [~/Desktop/task2] sudo rmmod md1
+ [~/Desktop/task2] sudo dmesg | tail -2
[18602.876520] MODULE2: unloaded
[18604.406531] MODULE1: unloaded

```

4. Функция инициализации модуля 3, выполнив все предписанные ей действия, преднамеренно возвращает ненулевое значение, что означает ошибку инициализации модуля. Этот модуль не будет подгружен к ядру, но произойдёт это уже после выполнения кода инициализирующей функции модуля в пространстве ядра:

```

+ ~/Desktop/task2 sudo insmod md1.ko
+ ~/Desktop/task2 sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
x + ~/Desktop/task2 sudo dmesg | tail -12
[18755.253669] MODULE1: loaded
[18759.194954] MODULE3: loaded
[18759.194955] MODULE3: Число экспортированное из md1 : 111
[18759.194956] MODULE3: Строка экспортированная из md1 : Привет из модуля 1!
[18759.194957] MODULE1: md1_get_str(10) called
[18759.194958] MODULE3: Результат работы функции md1_get_str(2) : Передайте 1 для приветствия или 2 для прощания
[18759.194960] MODULE1: md1_get_str(1) called
[18759.194964] MODULE3: Результат работы функции md1_get_str(0) : Привет!
[18759.194964] MODULE1: md1_get_str(2) called
[18759.194965] MODULE3: Результат работы функции md1_get_str(1) : Пока!
[18759.194965] MODULE1: md1_factorial(10) called
[18759.194966] MODULE3: Результат работы функции md1_factorial(4) : 3628800
+ ~/Desktop/task2 lsmod | grep md
md1 16384 0

```