



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»

Лабораторная работа №3
По курсу «Операционные системы»
Тема: Загружаемые модули ядра

Студент: Кондрашова О.П.
Группа: ИУ7-65Б
Преподаватель: Рязанова Н.Ю.

Москва, 2020г.

Задание 1

Листинг 1. Код mod.c

```
1. #include <linux/init.h>
2. #include <linux/module.h>
3. #include <linux/init_task.h>
4.
5. MODULE_LICENSE("GPL");
6. MODULE_AUTHOR("Kondrashova");
7. MODULE_DESCRIPTION("lab3");
8.
9. static int __init my_module_init(void)
10. {
11.     printk(KERN_INFO "Module1: Hello world\n");
12.
13.     struct task_struct *task = &init_task;
14.     do
15.     {
16.         // Информация о процессах берется из task_struct
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.     // Выводит insmod
23.     printk(KERN_INFO "Module1: current: %s - %d, parent: %s - %d\n",
24.         current->comm, current->pid, current->parent->comm, current->parent->pid);
25.     return 0;
26. }
27.
28. static void __exit my_module_exit(void)
29. {
30.     printk(KERN_INFO "Module1: Good bye\n");
31. }
32.
33. module_init(my_module_init);
34. module_exit(my_module_exit);
```

Листинг 1. Makefile

```
1. ifneq ($(KERNELRELEASE),)
2.     obj-m := mod.o
3. else
4.     CURRENT = $(shell uname -r)
5.     KDIR = /lib/modules/$(CURRENT)/build
6.     PWD = $(shell pwd)
7.
8.     default:
9.         $(MAKE) -C $(KDIR) M=$(PWD) modules
10.
11. clean:
12.     rm -rf .tmp_versions
13.     rm *.ko
14.     rm *.o
15.     rm *.mod.c
16.     rm *.symvers
17.     rm *.order
18.
19. endif
```

Загрузка модуля ядра:

```
olga@olga-VirtualBox:~/Documents/lab3/part1$ sudo insmod mod.ko
[sudo] password for olga:
olga@olga-VirtualBox:~/Documents/lab3/part1$ lsmod | grep mod
mod                16384  0
```

Вывод буфера сообщений ядра в стандартный поток вывода:

```
olga@olga-VirtualBox:~/Documents/lab3/part1$ sudo dmesg | grep Module1
[ 4079.646424] Module1: Hello world
[ 4079.646426] Module1: process: swapper/0 - 0, parent: swapper/0 - 0
[ 4079.646427] Module1: process: systemd - 1, parent: swapper/0 - 0
[ 4079.646427] Module1: process: kthreadd - 2, parent: swapper/0 - 0
[ 4079.646428] Module1: process: rcu_gp - 3, parent: kthreadd - 2
[ 4079.646429] Module1: process: rcu_par_gp - 4, parent: kthreadd - 2
[ 4079.646429] Module1: process: kworker/0:0 - 5, parent: kthreadd - 2
[ 4079.646430] Module1: process: kworker/0:0H - 6, parent: kthreadd - 2
[ 4079.646431] Module1: process: mm_percpu_wq - 9, parent: kthreadd - 2
[ 4079.646432] Module1: process: ksoftirqd/0 - 10, parent: kthreadd - 2
[ 4079.646432] Module1: process: rcu_sched - 11, parent: kthreadd - 2
[ 4079.646433] Module1: process: migration/0 - 12, parent: kthreadd - 2
[ 4079.646434] Module1: process: idle_inject/0 - 13, parent: kthreadd - 2
[ 4079.646434] Module1: process: cpuhp/0 - 14, parent: kthreadd - 2
[ 4079.646435] Module1: process: kdevtmpfs - 15, parent: kthreadd - 2
[ 4079.646436] Module1: process: netns - 16, parent: kthreadd - 2
[ 4079.646437] Module1: process: rcu_tasks_kthre - 17, parent: kthreadd - 2
[ 4079.646437] Module1: process: kauditd - 18, parent: kthreadd - 2
[ 4079.646438] Module1: process: khungtaskd - 19, parent: kthreadd - 2
[ 4079.646446] Module1: process: oom_reaper - 20, parent: kthreadd - 2
[ 4079.646447] Module1: process: writeback - 21, parent: kthreadd - 2
[ 4079.646448] Module1: process: kcompactd0 - 22, parent: kthreadd - 2
[ 4079.646448] Module1: process: ksmd - 23, parent: kthreadd - 2
[ 4079.646449] Module1: process: khugepaged - 24, parent: kthreadd - 2
[ 4079.646450] Module1: process: kintegrityd - 116, parent: kthreadd - 2
[ 4079.646451] Module1: process: kblockd - 117, parent: kthreadd - 2
[ 4079.646451] Module1: process: blkcg_punt_bio - 118, parent: kthreadd - 2
[ 4079.646452] Module1: process: tpm dev wq - 119, parent: kthreadd - 2
```

Выгрузка яда

В буфере сообщений видно, что current выводит insmod

```
olga@olga-VirtualBox:~/Documents/lab3/part1$ sudo rmmod mod
olga@olga-VirtualBox:~/Documents/lab3/part1$ sudo dmesg | tail -4
[ 4079.646597] Module1: process: sudo - 3767, parent: bash - 2192
[ 4079.646597] Module1: process: insmod - 3768, parent: sudo - 3767
[ 4079.646598] Module1: current: insmod - 3768, parent: sudo - 3767
[ 4265.240842] Module1: Good bye
```

Задание 2

В данных модулях демонстрируется работа экспортируемых данных и функций. Загружаемый модуль md1 экспортирует переменные md1_str_data и md1_int_data, которые используются в модулях md2 и md3.

Листинг 3. Код md.h

```
1. extern int md1_int_data;
2. extern char* md1_str_data;
3. extern char* md1_get_str(int n);
4. extern int md1_factorial(int n);
```

Листинг 4. Код md1.c

```
1. #include <linux/init.h>
2. #include <linux/module.h>
3. #include "md.h"
4.
5. MODULE_LICENSE("GPL");
6. MODULE_AUTHOR("Kondrashova");
7. MODULE_DESCRIPTION("lab3");
8.
9. char* md1_str_data = "Привет мир";
10. int md1_int_data = 64;
11.
12. extern char* md1_get_str(int n)
13. {
14.     printk(KERN_INFO "md1: md1_get_str(%d) called\n", n);
15.     switch (n)
16.     {
17.         case 1:
18.             return "Первая строка";
19.             break;
20.         case 2:
21.             return "Вторая строка";
22.             break;
23.         default:
24.             return "1 - первая строка, 2 - вторая строка";
25.             break;
26.     }
27. }
28.
29. extern int md1_factorial(int n)
30. {
31.     printk(KERN_INFO "md1: 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. EXPORT_SYMBOL(md1_str_data);
45. EXPORT_SYMBOL(md1_int_data);
46. EXPORT_SYMBOL(md1_get_str);
47. EXPORT_SYMBOL(md1_factorial);
```

```

48.
49. static int __init my_module_init(void)
50. {
51.     printk(KERN_INFO "md1: loaded\n");
52.     return 0;
53. }
54.
55. static void __exit my_module_exit(void)
56. {
57.     printk(KERN_INFO "md1: unloaded\n");
58. }
59.
60. module_init(my_module_init);
61. module_exit(my_module_exit);

```

Листинг 5. Код md2.c

```

1. #include <linux/init.h>
2. #include <linux/module.h>
3. #include "md.h"
4.
5. MODULE_LICENSE("GPL");
6. MODULE_AUTHOR("Kondrashova");
7. MODULE_DESCRIPTION("lab3");
8.
9. static int __init my_module_init(void)
10. {
11.     printk(KERN_INFO "md2: Старт\n");
12.     printk(KERN_INFO "md2: Число экспортированное из md1: %d\n", md1_int_data);
13.     printk(KERN_INFO "md2: Строка экспортированная из md1: %s\n", md1_str_data);
14.     printk(KERN_INFO "md2: Результат работы функции md1_get_str(0): %s\n", md1_get_str(
15.         0));
16.     printk(KERN_INFO "md2: Результат работы функции md1_get_str(1): %s\n", md1_get_str(
17.         1));
18.     printk(KERN_INFO "md2: Результат работы функции md1_get_str(2): %s\n", md1_get_str(
19.         2));
20.     printk(KERN_INFO "md2: Результат работы функции md1_factorial(5): %d\n", md1_factor
21.         ial(5));
22.     return 0;
23. }
24.
25. static void __exit my_module_exit(void)
26. {
27.     printk(KERN_INFO "md2: unloaded\n");
28. }
29.
30. module_init(my_module_init);
31. module_exit(my_module_exit);

```

Листинг 6. Код md3.c

```

1. #include <linux/init.h>
2. #include <linux/module.h>
3. #include "md.h"
4.
5. MODULE_LICENSE("GPL");
6. MODULE_AUTHOR("Kondrashova");
7. MODULE_DESCRIPTION("lab3");
8.
9. static int __init my_module_init(void)
10. {
11.     printk(KERN_INFO "md3: Старт\n");

```

```

12.     printk(KERN_INFO "md3: Число экспортированное из md1: %d\n", md1_int_data);
13.     printk(KERN_INFO "md3: Строка экспортированная из md1: %s\n", md1_str_data);
14.     printk(KERN_INFO "md3: Результат работы функции md1_get_str(0): %s\n", md1_get_str(
    0));
15.     printk(KERN_INFO "md3: Результат работы функции md1_get_str(1): %s\n", md1_get_str(
    1));
16.     printk(KERN_INFO "md3: Результат работы функции md1_get_str(2): %s\n", md1_get_str(
    2));
17.     printk(KERN_INFO "md3: Результат работы функции md1_factorial(5): %d\n", md1_factor
    ial(5));
18.     return -1;
19. }
20.
21. module_init(my_module_init);

```

Листинг 7. Makefile

```

1. ifneq ($(KERNELRELEASE),)
2.     obj-m := md1.o md2.o md3.o
3. else
4.     CURRENT = $(shell uname -r)
5.     KDIR = /lib/modules/$(CURRENT)/build
6.     PWD = $(shell pwd)
7.
8.     default:
9.         $(MAKE) -C $(KDIR) M=$(PWD) modules
10.
11. clean:
12.     rm -rf .tmp_versions
13.     rm *.ko
14.     rm *.o
15.     rm *.mod.c
16.     rm *.symvers
17.     rm *.order
18.
19. endif

```

Т.к. модули md2 и md3 импортируют данные из модуля md1, необходимо сначала загрузить модуль md1. В противном случае возникает ошибка.

```

olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo insmod md2.ko
[sudo] password for olga:
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo dmesg |tail -4
[ 7209.518075] md2: Unknown symbol md1_int_data (err -2)
[ 7209.518086] md2: Unknown symbol md1_factorial (err -2)
[ 7209.518096] md2: Unknown symbol md1_get_str (err -2)
[ 7209.518104] md2: Unknown symbol md1_str_data (err -2)
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Unknown symbol in module
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo dmesg |tail -4
[ 7741.364743] md3: Unknown symbol md1_int_data (err -2)
[ 7741.364796] md3: Unknown symbol md1_factorial (err -2)
[ 7741.364896] md3: Unknown symbol md1_get_str (err -2)
[ 7741.364907] md3: Unknown symbol md1_str_data (err -2)

```

Загрузим модули в правильном порядке. Модули md1 и md2 загружены успешно.

Функция инициализации модуля md3 возвращает ненулевое значение, что означает ошибку инициализации модуля. Модуль md3 не будет загружен, но произойдет это уже после выполнения кода инициализирующей функции модуля в пространстве ядра.

```
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo insmod md1.ko
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo insmod md2.ko
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
```

Вывод буфера сообщений ядра в стандартный поток вывода:

```
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo dmesg | tail -12
[ 8350.070170] md1: loaded
[ 8363.438680] md2: Старт
[ 8363.438682] md2: Число экспортированное из md1: 64
[ 8363.438683] md2: Строка экспортированная из md1: Привет мир
[ 8363.438684] md1: md1_get_str(0) called
[ 8363.438685] md2: Результат работы функции md1_get_str(0): 1 - первая строка, 2 - вторая строка
[ 8363.438685] md1: md1_get_str(1) called
[ 8363.438686] md2: Результат работы функции md1_get_str(1): Первая строка
[ 8363.438686] md1: md1_get_str(2) called
[ 8363.438686] md2: Результат работы функции md1_get_str(2): Вторая строка
[ 8363.438687] md1: md1_factorial(5) called
[ 8363.438687] md2: Результат работы функции md1_factorial(5): 120
```

Модуль md1 используется модулем md2

```
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo lsmod | grep md
md2                16384  0
md1                16384  1 md2
```

Пока модуль md1 используется другим модулем, его нельзя выгрузить

```
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2
```

Выгружаем модули в обратном порядке. Если сначала выгрузить md2, который импортирует данные из md1, то можно будет выгрузить и md1

```
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo rmmod md2
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo rmmod md1
olga@olga-VirtualBox:~/Documents/lab3/part2$ sudo dmesg | tail -2
[ 9090.354261] md2: unloaded
[ 9093.174841] md1: unloaded
```