



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 предмету: «Операционные системы»

Тема: Загружаемые модули ядра

Преподаватель: Рязанова Н.Ю.

Студент: Мирзоян С.А.,

Группа: ИУ7-65Б

Москва, 2020 г.

Задание 2

Реализовать три загружаемых модуля ядра:

- Вызываемый модуль md1
- Вызывающий модуль md2
- «Отладочный» модуль md3

Загружаемые модули должны собираться при помощи Make-файла (сборка командой make). Вызов каждой функции модуля должен сопровождаться записью в системный журнал информации, какая функция, какого модуля была вызвана.

Модули

Модуль md1 демонстрирует возможность создания экспортируемых данных и функций.

Модуль md2 демонстрирует использование данных и функций, экспортируемых первым модулем.

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

Листинг

Модуль 1

```
1. #include <linux/init.h>
2.
3. #include <linux/module.h>
4.
5. #include "md.h"
6.
7.
8.
9. MODULE_LICENSE( "GPL" );
10.
```

```

11.     MODULE_AUTHOR( "Sergey Mirzoyan" );
12.
13.
14.
15.     char* mdl_str_data = "Привет мир!";
16.
17.     int mdl_int_data = 42;
18.
19.
20.
21.     extern char* mdl_get_str(int n)
22.
23.     {
24.
25.         printk( "+ mdl: mdl_get_str() called!\n" );
26.
27.         switch (n)
28.
29.         {
30.
31.             case 1:
32.
33.                 return "Hello world!\n";
34.
35.                 break;
36.
37.             case 2:
38.
39.                 return "Привет Мир!\n";
40.
41.                 break;
42.
43.             default:
44.
45.                 return "Передайте 1 для получения английского сообщения
или 2 для получения русского.\n";
46.
47.                 break;
48.
49.         }

```

```
50.
51.     }
52.
53.
54.
55.     extern int mdl_factorial(int n)
56.
57.     {
58.
59.         int i, ans;
60.
61.         ans = 1;
62.
63.
64.
65.         printk( "+ mdl: mdl_factorial() called!\n" );
66.
67.         for (i = 2; i <= n; i++) ans *= i;
68.
69.
70.
71.         return ans;
72.
73.     }
74.
75.
76.
77.     // экспортируем данные
78.
79.     EXPORT_SYMBOL(mdl_str_data);
80.
81.     EXPORT_SYMBOL(mdl_int_data);
82.
83.     // экспортируем функции
84.
85.     EXPORT_SYMBOL(mdl_get_str);
86.
87.     EXPORT_SYMBOL(mdl_factorial);
88.
89.
```

```

90.
91.
92.
93.     static int __init md_init( void )
94.
95.     {
96.
97.         printk( "+ md1: module md1 start!\n" );
98.
99.         return 0;
100.
101.     }
102.
103.
104.
105.     static void __exit md_exit( void )
106.
107.     {
108.
109.         printk( "+ md1: module md1 unloaded!\n" );
110.
111.     }
112.
113.
114.
115.     module_init( md_init );
116.
117.     module_exit( md_exit );

```

Модуль 2

```

1. #include <linux/init.h>
2.
3. #include <linux/module.h>
4.
5. #include "md.h"
6.
7.
8.

```

```

9. MODULE_LICENSE( "GPL" );
10.
11.     MODULE_AUTHOR( "Sergey Mirzoyan" );
12.
13.
14.
15.     static int __init md_init( void )
16.
17.     {
18.
19.         printk( "+ md2: module md2 start!\n" );
20.
21.         printk( "+ md2: Число экспортированное из md1 : %d\n",
                md1_int_data );
22.
23.         printk( "+ md2: Строка экспортированная из md1 : %s\n",
                md1_str_data );
24.
25.         printk( "+ md2: Результат работы функции
                md1_get_str(0) : %s\n", md1_get_str(0) );
26.
27.         printk( "+ md2: Результат работы функции
                md1_get_str(1) : %s\n", md1_get_str(1) );
28.
29.         printk( "+ md2: Результат работы функции
                md1_get_str(2) : %s\n", md1_get_str(2) );
30.
31.         printk( "+ md2: Результат работы функции md1 facto-
                rial(4) : %d\n", md1_factorial(4) );
32.
33.         return 0;
34.
35.     }
36.
37.
38.
39.     static void __exit md_exit( void )
40.
41.     {
42.

```

```

43.         printk( "+ md2: module md2 unloaded!\n" );
44.
45.     }
46.
47.
48.
49.     module_init( md_init );
50.
51.     module_exit( md_exit );

```

Модуль 3

```

1.#include <linux/module.h>
2.#include <linux/init.h>
3.#include "md.h"
4.
5.MODULE_LICENSE( "GPL" );
6.MODULE_AUTHOR( "Sergey Mirzoyan" );
7.
8.static int __init md_init( void )
9.{
10.    printk( "+ md3: module md3 start!\n" );
11.    printk( "+ md3: Число экспортированное из md1 : %d\n",
        md1_int_data );
12.    printk( "+ md3: Строка экспортированная из md1 : %s\n",
        md1_str_data );
13.    printk( "+ md3: Результат работы функции
        md1_get_str(0) : %s\n", md1_get_str(0) );
14.    printk( "+ md3: Результат работы функции
        md1_get_str(1) : %s\n", md1_get_str(1) );
15.    printk( "+ md3: Результат работы функции
        md1_get_str(2) : %s\n", md1_get_str(2) );
16.    printk( "+ md3: Результат работы функции md1_factorial(4) : %d\n", md1_factorial(4) );
17.    return -1;
18.}
19.
20.    static void __exit md_exit( void )
21.    {
22.        printk( "+ md1: module md1 unloaded!\n" );

```

```

23.     }
24.
25.     module_init( md_init );
26.     module_exit( md_exit );

```

Результат работы программы

```

sergey@sergey-VirtualBox:~/TASK2$ sudo insmod md2.ko
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
sergey@sergey-VirtualBox:~/TASK2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Unknown symbol in module
sergey@sergey-VirtualBox:~/TASK2$ sudo insmod md1.ko
sergey@sergey-VirtualBox:~/TASK2$ sudo insmod md2.ko

```

Рисунок 1. Загрузка модулей.

На рисунке 1 можно увидеть, что первые две попытки загрузить модули 2 и 3 не обвенчались успехом. Это связано с тем, что при загрузке модулей важен порядок загрузки. Так как модули 2 и 3 используют экспортируемые первым модулем данные и функции, то первым загружен быть должен именно первый модуль.

```

[ 1057.928626] + md1: module md1 start!
[ 1062.785680] + md2: module md2 start!
[ 1062.785682] + md2: Число экспортированное из md1 : 42
[ 1062.785684] + md2: Строка экспортированная из md1 : Привет мир!
[ 1062.785686] + md1: md1_get_str() called!
[ 1062.785688] + md2: Результат работы функции md1_get_str(0) : Передайте 1 для получения английского сообщения или 2 для получения русского.
[ 1062.785690] + md1: md1_get_str() called!
[ 1062.785692] + md2: Результат работы функции md1_get_str(1) : Hello world!
[ 1062.785693] + md1: md1_get_str() called!
[ 1062.785695] + md2: Результат работы функции md1_get_str(2) : Привет Мир!
[ 1062.785698] + md1: md1_factorial() called!
[ 1062.785700] + md2: Результат работы функции md1_factorial(4) : 24
[ 1065.497643] + md3: module md3 start!
[ 1065.497646] + md3: Число экспортированное из md1 : 42
[ 1065.497648] + md3: Строка экспортированная из md1 : Привет мир!
[ 1065.497650] + md1: md1_get_str() called!
[ 1065.497652] + md3: Результат работы функции md1_get_str(0) : Передайте 1 для получения английского сообщения или 2 для получения русского.
[ 1065.497653] + md1: md1_get_str() called!
[ 1065.497655] + md3: Результат работы функции md1_get_str(1) : Hello world!
[ 1065.497657] + md1: md1_get_str() called!
[ 1065.497659] + md3: Результат работы функции md1_get_str(2) : Привет Мир!
[ 1065.497661] + md1: md1_factorial() called!
[ 1065.497662] + md3: Результат работы функции md1_factorial(4) : 24

```

Рисунок 2. Вывод сообщений о работе модулей 1 и 2
(команда dmesg | tail n40)

```

sergey@sergey-VirtualBox:~/TASK2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted

```

Рисунок 3. Ошибка при загрузке 3 модуля.

Данная ошибка возникает при значении return -1; (см. Листинг -> md3 -> строка 17). Его нужно заменить на return 0;


```

sergey@sergey-VirtualBox:~/TASK2$ dmesg | tail -n14
[ 1980.405252] + md3: module md3 start!
[ 1980.405254] + md3: Число экспортированное из md1 : 42
[ 1980.405256] + md3: Строка экспортированная из md1 : Привет мир!
[ 1980.405258] + md1: md1_get_str() called!
[ 1980.405260] + md3: Результат работы функции md1_get_str(0) : Передайте 1 для получения английского сообщения или 2 для получения русского.
[ 1980.405262] + md1: md1_get_str() called!
[ 1980.405264] + md3: Результат работы функции md1_get_str(1) : Hello world!
[ 1980.405266] + md1: md1_get_str() called!
[ 1980.405267] + md3: Результат работы функции md1_get_str(2) : Привет Мир!
[ 1980.405269] + md1: md1_factorial() called!
[ 1980.405271] + md3: Результат работы функции md1_factorial(4) : 24

```

Рисунок 4. Успешная загрузка третьего модуля и пример его работы

```

sergey@sergey-VirtualBox:~/TASK2$ sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2 md3
sergey@sergey-VirtualBox:~/TASK2$ sudo rmmod md2
sergey@sergey-VirtualBox:~/TASK2$ sudo rmmod md3
sergey@sergey-VirtualBox:~/TASK2$ sudo rmmod md1
[ 2234.541836] + md2: module md2 unloaded!
[ 2236.455579] + md1: module md1 unloaded!
[ 2238.034854] + md1: module md1 unloaded!
sergey@sergey-VirtualBox:~/TASK2$

```

Рисунок 5.1, 5.2 Выгрузка модулей.

При выгрузке модулей также важен порядок, в котором они выгружаются. Так, если при загрузке модулей сперва загружался первый модуль, то теперь при выгрузке он будет являться завершающим, так как пока он используется другими модулями, он не может быть выгружен.