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	Операционые системы.
Тема	Загружаемые модули ядра.
Студент	Степанов А. О.
Группа	ИУ7-63Б
Оценка (баллы)	
Преподаватель	Рязанова Н.Ю.

Москва, 2020 г.

ЧАСТЬ 1

Листинг 1: Текст программы

```
1  #include <linux/module.h>
2  #include <linux/init.h>
3  #include <linux/kernel.h>
4  #include <linux/sched.h>
5  #include <linux/init_task.h>
6
7  MODULE_LICENSE("GPL");
8  MODULE_AUTHOR("Alexander Stepanov");
9  MODULE_DESCRIPTION("BMSTU operating sysytems lab_03_01");
10
11 static int __init alex_module_init(void)
12 {
13     printk(KERN_INFO "Module loaded\n");
14
15     struct task_struct *task = &init_task;
16
17     do
18     {
19         printk(KERN_INFO "%s -- %d, parent: %s -- %d\n",
20             task->comm, task->pid, task->parent->comm, task->parent->pid);
21     }
22     while ((task = next_task(task)) != &init_task);
23
24
25     printk(KERN_INFO "current: %s -- %d, parent: %s -- %d\n",
26         current->comm,
27         current->pid,
28         current->parent->comm,
29         current->parent->pid
30     );
31
32     return 0;
33 }
34
35 static void __exit alex_module_exit(void)
36 {
37     printk(KERN_INFO "Module exited\n");
38 }
39
```

```
40 module_init(alex_module_init);
41 module_exit(alex_module_exit);
```

```
→  task_01 git:(release/lab_03_sem_02) ✗ sudo insmod md.ko
→  task_01 git:(release/lab_03_sem_02) ✗ sudo lsmod | grep -B 1 md
Module                  Size  Used by
md                     16384  0
```

Рис. 1: Загрузка модуля

```
[  164.835963] Module loaded
[  164.835966] swapper/0 -- 0, parent: swapper/0 -- 0
[  164.835967] systemd -- 1, parent: swapper/0 -- 0
[  164.835968] kthreadd -- 2, parent: swapper/0 -- 0
[  164.835969] ksoftirqd/0 -- 3, parent: kthreadd -- 2
[  164.835971] kworker/0:0 -- 4, parent: kthreadd -- 2
[  164.835972] kworker/0:0H -- 5, parent: kthreadd -- 2
[  164.835973] kworker/u64:0 -- 6, parent: kthreadd -- 2
[  164.835974] rcu_sched -- 7, parent: kthreadd -- 2
[  164.835975] rcu_bh -- 8, parent: kthreadd -- 2
[  164.835976] migration/0 -- 9, parent: kthreadd -- 2
[  164.835977] lru-add-drain -- 10, parent: kthreadd -- 2
[  164.835978] watchdog/0 -- 11, parent: kthreadd -- 2
[  164.835980] cpuhp/0 -- 12, parent: kthreadd -- 2
[  164.835981] cpuhp/1 -- 13, parent: kthreadd -- 2
[  164.835981] watchdog/1 -- 14, parent: kthreadd -- 2
[  164.835983] migration/1 -- 15, parent: kthreadd -- 2
[  164.835984] ksoftirqd/1 -- 16, parent: kthreadd -- 2
[  164.835985] kworker/1:0 -- 17, parent: kthreadd -- 2
[  164.835986] kworker/1:0H -- 18, parent: kthreadd -- 2
[  164.835987] kdevtmpfs -- 19, parent: kthreadd -- 2
```

Рис. 2: Вывод dmesg (часть 1)

```
[  164.836164] gnome-software -- 1534, parent: gnome-session-b -- 1326
[  164.836165] evolution-alarm -- 1535, parent: gnome-session-b -- 1326
[  164.836166] tracker-extract -- 1537, parent: gnome-session-b -- 1326
[  164.836168] prlcc -- 1539, parent: gnome-session-b -- 1326
[  164.836169] tracker-miner-f -- 1542, parent: gnome-session-b -- 1326
[  164.836170] tracker-store -- 1543, parent: systemd -- 1304
[  164.836171] prldnd -- 1544, parent: prlcc -- 1539
[  164.836173] prlcp -- 1545, parent: prlcc -- 1539
[  164.836174] prlsga -- 1546, parent: prlcc -- 1539
[  164.836175] prlshprof -- 1547, parent: prlcc -- 1539
[  164.836176] tracker-miner-a -- 1550, parent: gnome-session-b -- 1326
[  164.836177] tracker-miner-u -- 1559, parent: gnome-session-b -- 1326
[  164.836178] gsd-printer -- 1586, parent: systemd -- 1
[  164.836179] evolution-calen -- 1607, parent: evolution-calen -- 1531
[  164.836180] dconf-service -- 1646, parent: systemd -- 1304
[  164.836182] evolution-addre -- 1650, parent: systemd -- 1304
[  164.836183] evolution-calen -- 1652, parent: evolution-calen -- 1531
[  164.836184] evolution-addre -- 1682, parent: evolution-addre -- 1650
[  164.836185] gnome-terminal- -- 1820, parent: systemd -- 1304
[  164.836186] zsh -- 1826, parent: gnome-terminal- -- 1820
[  164.836187] sudo -- 2875, parent: zsh -- 1826
[  164.836188] insmod -- 2876, parent: sudo -- 2875
[  164.836189] current: insmod -- 2876, parent: sudo -- 2875
```

Рис. 3: Вывод dmesg (часть 2)

```
→  task_01 git:(release/lab_03_sem_02) ✗ sudo rmmod md
→  task_01 git:(release/lab_03_sem_02) ✗ sudo dmesg | tail -1
[  247.902029] Module exited
```

Рис. 4: Выгрузка модуля

ЧАСТЬ 2

Листинг 2: md1.c

```
1  #include <linux/init.h>
2  #include <linux/module.h>
3  #include "md.h"
4
5  MODULE_LICENSE("GPL");
6  MODULE_AUTHOR("Alexander Stepanov");
7
8  char* md1_str_data = "Привет мир!";
9  int md1_int_data = 42;
10
```

```
11 extern char* md1_get_str(int n)
12 {
13     printk( "+ md1: md1_get_str() called!\n" );
14     switch (n)
15     {
16     case 1:
17         return "Hello world!\n";
18         break;
19     case 2:
20         return "Привет Мир!\n";
21         break;
22     default:
23         return "Передайте 1 для получения английского сообщения"
24             "или 2 для получения русского.\n";
25         break;
26     }
27 }
28
29 extern int md1_factorial(int n)
30 {
31     int i, ans;
32     ans = 1;
33
34     printk( "+ md1: md1_factorial() called!\n" );
35     for (i = 2; i <= n; i++) ans *= i;
36
37     return ans;
38 }
39
40 /* экспортируем данные */
41 EXPORT_SYMBOL(md1_str_data);
42 EXPORT_SYMBOL(md1_int_data);
43 /* экспортируем функции */
44 EXPORT_SYMBOL(md1_get_str);
45 EXPORT_SYMBOL(md1_factorial);
46
47
48 static int __init md_init( void )
49 {
50    printk( "+ md1: module md1 start!\n" );
51    return 0;
52 }
```

```
53
54 static void __exit md_exit( void )
55 {
56     printk( "+ md1: module md1 unloaded!\n" );
57 }
58
59 module_init( md_init );
60 module_exit( md_exit );
```

Листинг 3: md.h

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

Листинг 4: md2.c

```
1 #include <linux/init.h>
2 #include <linux/module.h>
3 #include "md.h"
4
5 MODULE_LICENSE("GPL");
6 MODULE_AUTHOR("Alexander Stepanov");
7
8 static int __init md_init( void )
9 {
10     printk( "+ md2: module md2 start!\n" );
11     printk( "+ md2: Число экспортированное из md1 : %d\n", md1_int_data );
12     printk( "+ md2: Строка экспортированная из md1 : %s\n", md1_str_data );
13     printk( "+ md2: Результат работы функции md1_get_str(0) : %s\n",
14      md1_get_str(0) );
15     printk( "+ md2: Результат работы функции md1_get_str(1) : %s\n",
16      md1_get_str(1) );
17     printk( "+ md2: Результат работы функции md1_get_str(2) : %s\n",
18      md1_get_str(2) );
19     printk( "+ md2: Результат работы функции md1_factorial(4) : %d\n",
20      md1_factorial(4) );
21     return 0;
22 }
23
24 static void __exit md_exit( void )
25 {
26     printk( "+ md2: module md2 unloaded!\n" );
```

```
27 }
28
29 module_init( md_init );
30 module_exit( md_exit );
```

Листинг 5: md3.c

```
1  #include <linux/init.h>
2  #include <linux/module.h>
3  #include "md.h"
4
5  MODULE_LICENSE("GPL");
6  MODULE_AUTHOR("Alexander Stepanov");
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",
14      md1_get_str(0));
15     printk("+ md3: Результат работы функции md1_get_str(1) : %s\n",
16      md1_get_str(1));
17     printk("+ md3: Результат работы функции md1_get_str(2) : %s\n",
18      md1_get_str(2));
19     printk("+ md3: Результат работы функции md1_factorial(4) : %d\n",
20      md1_factorial(4));
21     return -1;
22 }
23
24 module_init( md_init );
```

В данных модулях продемострирована работа экспортируемых данных и функций. Модуль md1 экспортирует переменные md1_str_data и md1_int_data, которые импортируют модули md2 и md3. Поскольку md2 и md3 импортируют данные из модуля md, то они не могут быть загружены до загрузки модуля md1, что можно увидеть на рисунке 5.

```
→  task_02 git:(release/lab_03_sem_02) ✗ sudo lsmod | head
Module                  Size  Used by
fuse                   98304  3
usblp                  20480  0
prl_fs_freeze          16384  0
prl_fs                 28672  2
prl_eth                16384  0
x86_pkg_temp_thermal    16384  0
coretemp               16384  0
crct10dif_pclmul       16384  0
crc32_pclmul           16384  0
→  task_02 git:(release/lab_03_sem_02) ✗ sudo insmod md2.ko
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
→  task_02 git:(release/lab_03_sem_02) ✗ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Unknown symbol in module
```

Рис. 5: Попытка загрузить md2 и md3 до md1

Если же загрузить сначала модуль, который экспортирует данные, то все модули загрузятся (кроме md3, при инициализации которого возвращается -1) (Рисунок 6).

```
→  task_02 git:(release/lab_03_sem_02) ✗ sudo insmod md1.ko
→  task_02 git:(release/lab_03_sem_02) ✗ sudo insmod md2.ko
→  task_02 git:(release/lab_03_sem_02) ✗ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
→  task_02 git:(release/lab_03_sem_02) ✗ sudo lsmod | head -5
Module                  Size  Used by
md2                    16384  0
md1                    16384  1 md2
fuse                   98304  3
usblp                  20480  0
```

Рис. 6: Попытка загрузка модулей

```
[  710.168967] + md1: module md1 start!
[  711.660661] + md2: module md2 start!
[  711.660663] + md2: Число экспортированное из md1 : 42
[  711.660663] + md2: Строка экспортированная из md1 : Привет мир!
[  711.660664] + md1: md1_get_str() called!
[  711.660664] + md2: Результат работы функции md1_get_str(0) : Передайте 1 для получения английского сообщения или 2 для получения русского.

[  711.660665] + md1: md1_get_str() called!
[  711.660665] + md2: Результат работы функции md1_get_str(1) : Hello world!

[  711.660665] + md1: md1_get_str() called!
[  711.660666] + md2: Результат работы функции md1_get_str(2) : Привет Мир!

[  711.660666] + md1: md1_factorial() called!
[  711.660666] + md2: Результат работы функции md1_factorial(4) : 24
[  713.126393] + md3: module md3 start!
[  713.126395] + md3: Число экспортированное из md1 : 42
[  713.126395] + md3: Строка экспортированная из md1 : Привет мир!
[  713.126396] + md1: md1_get_str() called!
[  713.126396] + md3: Результат работы функции md1_get_str(0) : Передайте 1 для получения английского сообщения или 2 для получения русского.

[  713.126397] + md1: md1_get_str() called!
[  713.126397] + md3: Результат работы функции md1_get_str(1) : Hello world!

[  713.126397] + md1: md1_get_str() called!
[  713.126398] + md3: Результат работы функции md1_get_str(2) : Привет Мир!

[  713.126398] + md1: md1_factorial() called!
[  713.126398] + md3: Результат работы функции md1_factorial(4) : 24
```

Рис. 7: Вывод dmesg

В выводе команды lsmod можно увидеть, что модуль md1 используется модулем md2. Пока модуль md1 используется другим модулем, его нельзя выгрузить, что видно на рисунке 8.

```
→  task_02 git:(release/lab_03_sem_02) ✗ sudo lsmod | grep -B 1 md
Module                  Size  Used by
md2                    16384  0
md1                    16384  1 md2
→  task_02 git:(release/lab_03_sem_02) ✗ sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2
```

Рис. 8: Попытка выгрузить md1 до md2

Если сначала выгрузить модуль md2, который импортирует данные, то можно выгрузить и md1 (Рисунок 9).

```
→  task_02 git:(release/lab_03_sem_02) ✗ sudo rmmod md2
→  task_02 git:(release/lab_03_sem_02) ✗ sudo lsmod | grep -B 1 md
Module                  Size  Used by
md1                    16384  0
→  task_02 git:(release/lab_03_sem_02) ✗ sudo rmmod md1
→  task_02 git:(release/lab_03_sem_02) ✗ sudo lsmod | grep -B 1 md
→  task_02 git:(release/lab_03_sem_02) ✗
```

Рис. 9: Выгрузка модулей