



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

Дисциплина Операционные системы

Тема Загружаемые модуля ядра

Студент Набиев Ф.М.

Группа ИУ7-63Б

Оценка (баллы)

Преподаватель Рязанова Н.Ю.

Москва, 2020 г.

1 Задание №1

1.1 Условие

Реализовать загружаемый модуль ядра, который при загрузке записывает в системный журнал сообщение о запущенных процессах. Модуль должен собираться при помощи Make-файла.

1.2 Реализация

Листинг 1.1 – Makefile

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

Листинг 1.2 – md.c

```
1 #include <linux/init.h>
2 #include <linux/sched.h>
3 #include <linux/module.h>
4 #include <linux/kernel.h>
5 #include <linux/init_task.h>
6
7 MODULE_LICENSE("GPL");
8 MODULE_AUTHOR("Faris Nabiev");
9
10 static void __exit my_module_exit(void);
11 static int __init my_module_init(void);
```

```

12
13 module_init(my_module_init);
14 module_exit(my_module_exit);
15
16 // Инициализация модуля
17 static int __init my_module_init(void)
18 {
19     struct task_struct *task = &init_task;
20
21     printk(KERN_INFO "Module task_01: loaded!\n");
22
23     do
24     {
25         printk(KERN_INFO "Module task_01: "
26                 "process: %s - %d, " "parent: %s - %d\n",
27                 task->comm, task->pid, task->parent->comm, task->parent->pid);
28     }
29     while ((task = next_task(task)) != &init_task);
30
31     printk(KERN_INFO "Module task_01: "
32             "current: %s - %d, parent: %s - %d\n",
33             current->comm, current->pid,
34             current->parent->comm, current->parent->pid);
35
36     return 0;
37 }
38
39 // Выгрузка модуля
40 static void __exit my_module_exit(void)
41 {
42     printk(KERN_INFO "Module task_01: unloaded\n");
43 }

```

1.3 Результаты работы

Рис. 1.1 – Сборка и загрузка модуля

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
# make
make -C /lib/modules/5.6.0-0.bpo.2-amd64/build M=/home/faris/Docu
make[1]: Entering directory '/usr/src/linux-headers-5.6.0-0.bpo.2-
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystem
MODPOST 1 modules
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystem
LD [M] /home/faris/Documents/Repositories/bmstu/OperatingSystem
make[1]: Leaving directory '/usr/src/linux-headers-5.6.0-0.bpo.2-

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
# sudo insmod md.ko
```

Рис. 1.2 – Проверка списка загруженных модулей

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
# sudo lsmod
Module                Size  Used by
md                    16384  0
fuse                  139264  5
ctr                   16384  2
ccm                   20480  6
```

Рис. 1.3 – Вывод буфера сообщений ядра

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01 master ?
# sudo dmesg | tail -n 278
[55344.787744] Module task_01: loaded!
[55344.787746] Module task_01: process: swapper/0 - 0, parent: swapper/0 - 0
[55344.787747] Module task_01: process: systemd - 1, parent: swapper/0 - 0
[55344.787748] Module task_01: process: kthreadd - 2, parent: swapper/0 - 0
[55344.787750] Module task_01: process: rcu_gp - 3, parent: kthreadd - 2
[55344.787751] Module task_01: process: rcu_par_gp - 4, parent: kthreadd - 2
[55344.787752] Module task_01: process: kworker/0:0H - 6, parent: kthreadd - 2
[55344.787753] Module task_01: process: mm_percpu_wq - 9, parent: kthreadd - 2
[55344.787754] Module task_01: process: ksoftirqd/0 - 10, parent: kthreadd - 2
[55344.787755] Module task_01: process: rcu_sched - 11, parent: kthreadd - 2
[55344.787757] Module task_01: process: migration/0 - 12, parent: kthreadd - 2
[55344.787758] Module task_01: process: cpuhp/0 - 13, parent: kthreadd - 2
[55344.787759] Module task_01: process: cpuhp/1 - 14, parent: kthreadd - 2
```

Рис. 1.4 – Выгрузка модуля и проверка буфера сообщений ядра

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
# sudo rmmod md

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
# sudo dmesg | tail -n 1
[55662.563736] Module task_01: unloaded
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_01
```

2 Задание №2

2.1 Условие

Реализовать три загружаемых модуля ядра: md1, md2, md3.

2.1.1 md1

Модуль md1 демонстрирует возможность создания экспортируемых данных и функции. Данный модуль ядра должен содержать:

- экспортируемые строковые (char *) и численные (int) данные;
- экспортируемые функции возвращающие строковые и числовые значения;

2.1.2 md2

Модуль md2 демонстрирует использование данных и функции, экспортируемых первым модулем (md1). Данный модуль должен при загрузке:

- вызывать все экспортированные модулем md1 процедуры и вывести в системный журнал возвращаемые ими значения с указанием имени вызванной процедуры;
- вывести в системный журнал все экспортированные модулем md1 данные.

2.1.3 md3

Модуль md3 демонстрирует сценарии некорректного завершения установки модуля, и возможность использования загружаемого модуля в качестве функции, выполняемой в пространстве ядра. Процедура инициализации этого загружаемого модуля должна возвращать ненулевое значение и выводить в системный журнал данные и возвращаемые значения экспортированных модулем md1 процедур (аналогично

md2). Данный модуль включен в работу для проработки вопросов, связанных с отладкой модулей ядра.

2.2 Реализация

Листинг 2.1 – 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 endif
```

Листинг 2.2 – md1.h

```
1 #ifndef OSLAB03_MD1_H_
2 #define OSLAB03_MD1_H_
3
4 extern int    md1_int_data;
5 extern char *md1_str_data;
6
7 extern char *md1_get_str(int n);
8 extern int    md1_factorial(int n);
9
10 #endif // OSLAB03_MD1_H_
```

Листинг 2.3 – md1.c

```
1 #include "md1.h"
2
3 #include <linux/init.h>
```

```

4 #include <linux/module.h>
5
6 MODULE_LICENSE("GPL");
7 MODULE_AUTHOR("Faris Nabiev");
8
9 static int __init my_module_init(void);
10 static void __exit my_module_exit(void);
11
12 module_init(my_module_init);
13 module_exit(my_module_exit);
14
15 int md1_int_data = 255;
16 char *md1_str_data = "First module str data";
17
18 static int __init my_module_init(void)
19 {
20     printk(KERN_INFO "Module 1 task_02: init\n");
21
22     return 0;
23 }
24
25 static void __exit my_module_exit(void)
26 {
27     printk(KERN_INFO "Module 1 task_02: unloaded\n");
28 }
29
30 extern char *md1_get_str(int n)
31 {
32     printk(KERN_INFO "Module 1 task_02: md1_get_str(%d) has called\n", n);
33
34     switch (n)
35     {
36         case 1:
37             return "First message";
38
39         case 2:
40             return "Second message";
41
42         default:
43             return "Default message";
44     }
45 }
46
47 extern int md1_factorial(int n)
48 {
49     int i;
50     int res = 1;
51

```

```

52     printk(KERN_INFO "Module 1 task_02: mdl_factorial(%d) has called\n",
53             n);
54
55     if (n > 1)
56         for (i = 2; i <= n; i++)
57             res *= i;
58
59     return res;
60 }
61
62 // Экспорт данных
63 EXPORT_SYMBOL(mdl_str_data);
64 EXPORT_SYMBOL(mdl_int_data);
65 // Экспорт функций
66 EXPORT_SYMBOL(mdl_get_str);
67 EXPORT_SYMBOL(mdl_factorial);

```

Листинг 2.4 – md2.c

```

1  #include "mdl.h"
2
3  #include <linux/init.h>
4  #include <linux/module.h>
5
6  MODULE_LICENSE("GPL");
7  MODULE_AUTHOR("Faris Nabiev");
8
9  static int __init my_module_init(void);
10 static void __exit my_module_exit(void);
11
12 module_init(my_module_init);
13 module_exit(my_module_exit);
14
15 static int __init my_module_init(void)
16 {
17     printk(KERN_INFO "Module 2 task_02: init\n");
18
19     printk(KERN_INFO "Module 2 task_02: mdl_int_data      = %d\n",
20            mdl_int_data);
21     printk(KERN_INFO "Module 2 task_02: mdl_str_data      = %s\n",
22            mdl_str_data);
23     printk(KERN_INFO "Module 2 task_02: mdl_get_str(10)   = %s\n",
24            mdl_get_str(10));
25     printk(KERN_INFO "Module 2 task_02: mdl_get_str(1)    = %s\n",
26            mdl_get_str(1));
27     printk(KERN_INFO "Module 2 task_02: mdl_get_str(2)    = %s\n",
28            mdl_get_str(2));
29     printk(KERN_INFO "Module 2 task_02: mdl_factorial(10) = %d\n",

```



```

30         mdl_factorial(10));
31
32     return 0;
33 }
34
35 static void __exit my_module_exit(void)
36 {
37     printk(KERN_INFO "Module 2 task_02: unloaded\n");
38 }

```

Листинг 2.5 – md3.c

```

1  #include "mdl.h"
2
3  #include <linux/init.h>
4  #include <linux/module.h>
5
6  MODULE_LICENSE("GPL");
7  MODULE_AUTHOR("Faris Nabiev");
8
9  static int __init my_module_init(void);
10
11 module_init(my_module_init);
12
13 static int __init my_module_init(void)
14 {
15     printk(KERN_INFO "Module 3 task_02: init\n");
16
17     printk(KERN_INFO "Module 3 task_02: mdl_int_data      = %d\n",
18            mdl_int_data);
19     printk(KERN_INFO "Module 3 task_02: mdl_str_data      = %s\n",
20            mdl_str_data);
21     printk(KERN_INFO "Module 3 task_02: mdl_get_str(10)   = %s\n",
22            mdl_get_str(10));
23     printk(KERN_INFO "Module 3 task_02: mdl_get_str(1)    = %s\n",
24            mdl_get_str(1));
25     printk(KERN_INFO "Module 3 task_02: mdl_get_str(2)    = %s\n",
26            mdl_get_str(2));
27     printk(KERN_INFO "Module 3 task_02: mdl_factorial(10) = %d\n",
28            mdl_factorial(10));
29
30     return -1;
31 }

```

2.3 Результаты работы

Рис. 2.1 – Сборка модулей

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# make
make -C /lib/modules/5.6.0-0.bpo.2-amd64/build M=/home/faris/Documents/...
make[1]: Entering directory '/usr/src/linux-headers-5.6.0-0.bpo.2-amd64'
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
MODPOST 3 modules
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
LD [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
LD [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
CC [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
LD [M] /home/faris/Documents/Repositories/bmstu/OperatingSystems/lab_...
make[1]: Leaving directory '/usr/src/linux-headers-5.6.0-0.bpo.2-amd64'
```

Рис. 2.2 – Попытка загрузки модулей в неправильном порядке

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# sudo insmod md2.ko
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# sudo dmesg | tail -n 4
[57478.700036] md2: Unknown symbol md1_int_data (err -2)
[57478.700105] md2: Unknown symbol md1_factorial (err -2)
[57478.700168] md2: Unknown symbol md1_get_str (err -2)
[57478.700228] md2: Unknown symbol md1_str_data (err -2)
```

Причина возникшей ошибки заключается в том, что модуль md2 содержит ссылки на неизвестные ему имена. Значит, сначала нужно загрузить модуль md1.

Рис. 2.3 – Загрузка модулей в правильном порядке

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# sudo insmod md1.ko

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# sudo insmod md2.ko

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master
# sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
```

Модуль md3 не был загружен, потому что его функция инициализации возвращает -1.

Рис. 2.4 – Проверка списка загруженных модулей

```
# sudo lsmod
Module                Size  Used by
md2                   16384  0
md1                   16384  1 md2
fuse                  139264  5
ctr                   16384  2
```

Рис. 2.5 – Вывод буфера сообщений ядра

```
# sudo dmesg | tail -n 23
[57969.417375] Module 1 task_02: init
[57972.035324] Module 2 task_02: init
[57972.035327] Module 2 task_02: md1_int_data      = 255
[57972.035329] Module 2 task_02: md1_str_data      = First module str data
[57972.035331] Module 1 task_02: md1_get_str(10) has called
[57972.035332] Module 2 task_02: md1_get_str(10)  = Default message
[57972.035334] Module 1 task_02: md1_get_str(1) has called
[57972.035335] Module 2 task_02: md1_get_str(1)   = First message
[57972.035336] Module 1 task_02: md1_get_str(2) has called
[57972.035338] Module 2 task_02: md1_get_str(2)   = Second message
[57972.035339] Module 1 task_02: md1_factorial(10) has called
[57972.035341] Module 2 task_02: md1_factorial(10) = 3628800
[57974.321082] Module 3 task_02: init
[57974.321086] Module 3 task_02: md1_int_data      = 255
[57974.321088] Module 3 task_02: md1_str_data      = First module str data
[57974.321091] Module 1 task_02: md1_get_str(10) has called
[57974.321093] Module 3 task_02: md1_get_str(10)  = Default message
[57974.321094] Module 1 task_02: md1_get_str(1) has called
[57974.321096] Module 3 task_02: md1_get_str(1)   = First message
[57974.321097] Module 1 task_02: md1_get_str(2) has called
[57974.321098] Module 3 task_02: md1_get_str(2)   = Second message
[57974.321100] Module 1 task_02: md1_factorial(10) has called
[57974.321102] Module 3 task_02: md1_factorial(10) = 3628800
```

Рис. 2.6 – Попытка выгрузки модулей в неправильном порядке

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master ?
# sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2
```

Попытка выгрузки модулей в неправильном порядке также вызывает ошибку.

Рис. 2.7 – Выгрузка модулей в правильном порядке и вывод буфера сообщений ядра

```
~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master ?
# sudo rmmod md2

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master ?
# sudo rmmod md1

~/Documents/Repositories/bmstu/OperatingSystems/lab_13/task_02 master ?
# sudo dmesg | tail -n 2
[57853.614284] Module 2 task_02: unloaded
[57855.066155] Module 1 task_02: unloaded
```