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

Студент: Лаврова А. А.,
Группа: ИУ7-65Б
Преподаватель: Рязанова Н. Ю.

Москва, 2020 г.

Задание №1

Листинг программы – файл mod.c

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/init_task.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Lavrova");
MODULE_DESCRIPTION("Lab_3");

static int __init my_module_init(void)
{
   printk(KERN_INFO "Module1: Hello world\n");

   struct task_struct *task = &init_task;
   do
   {
     printk(KERN_INFO "Module1: process: %s - %d, parent: %s -
%d\n",
     task->comm, task->pid, task->parent->comm, task->parent-
>pid);
   }
   while ((task = next_task(task)) != &init_task);

   printk(KERN_INFO "Module1: current: %s - %d, parent: %s -
%d\n",
   current->comm, current->pid, current->parent->comm, current-
>parent->pid);
   return 0;
}

static void __exit my_module_exit(void)
{
   printk(KERN_INFO "Module1: Good bye\n");
}

module_init(my_module_init);
module_exit(my_module_exit);
```

Листинг– Makefile:

```
ifneq ($(KERNELRELEASE),)
        obj-m := mod.o

else
        CURRENT = $(shell uname -r)
        KDIR = /lib/modules/$(CURRENT)/build
        PWD = $(shell pwd)

default:
        $(MAKE) -C $(KDIR) M=$(PWD) modules
```

```
clean:
	rm -rf .tmp_versions
	rm *.ko
	rm *.o
	rm *.mod.c
	rm *.symvers
	rm *.order

endif
```

Результаты работы программы:

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod mod.ko
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ lsmod | grep mod
mod                    16384  0
```

Рис.1: Загрузка модуля ядра

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | grep Module1
[10746.853515] Module1: Hello world
[10746.853517] Module1: process: swapper/0 - 0, parent: swapper/0 - 0
[10746.853518] Module1: process: systemd - 1, parent: swapper/0 - 0
[10746.853518] Module1: process: kthreadd - 2, parent: swapper/0 - 0
[10746.853519] Module1: process: kworker/0:0H - 4, parent: kthreadd - 2
[10746.853520] Module1: process: mm_percpu_wq - 6, parent: kthreadd - 2
[10746.853521] Module1: process: ksoftirqd/0 - 7, parent: kthreadd - 2
[10746.853521] Module1: process: rcu_sched - 8, parent: kthreadd - 2
[10746.853522] Module1: process: rcu_bh - 9, parent: kthreadd - 2
[10746.853523] Module1: process: migration/0 - 10, parent: kthreadd - 2
[10746.853523] Module1: process: watchdog/0 - 11, parent: kthreadd - 2
[10746.853524] Module1: process: cpuhp/0 - 12, parent: kthreadd - 2
[10746.853525] Module1: process: cpuhp/1 - 13, parent: kthreadd - 2
[10746.853526] Module1: process: watchdog/1 - 14, parent: kthreadd - 2
[10746.853527] Module1: process: migration/1 - 15, parent: kthreadd - 2
[10746.853527] Module1: process: ksoftirqd/1 - 16, parent: kthreadd - 2
[10746.853528] Module1: process: kworker/1:0H - 18, parent: kthreadd - 2
[10746.853529] Module1: process: kdevtmpfs - 19, parent: kthreadd - 2
[10746.853529] Module1: process: netns - 20, parent: kthreadd - 2
[10746.853530] Module1: process: rcu_tasks_kthre - 21, parent: kthreadd - 2
[10746.853531] Module1: process: kauditd - 22, parent: kthreadd - 2
[10746.853531] Module1: process: khungtaskd - 25, parent: kthreadd - 2
[10746.853532] Module1: process: oom_reaper - 26, parent: kthreadd - 2
[10746.853533] Module1: process: writeback - 27, parent: kthreadd - 2
[10746.853534] Module1: process: kcompactd0 - 28, parent: kthreadd - 2
[10746.853534] Module1: process: ksmd - 29, parent: kthreadd - 2
[10746.853535] Module1: process: khugepaged - 30, parent: kthreadd - 2
```

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

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo rmmod mod
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | tail -4
[10746.853723] Module1: process: sudo - 22781, parent: bash - 22131
[10746.853724] Module1: process: insmod - 22782, parent: sudo - 22781
[10746.853724] Module1: current: insmod — 22782, parent: sudo — 22781
[10855.570307] Module1: Good bye
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$
```

Рис. 3: Выгрузка яда (В буфере сообщений видно, что current выводит insmod)

Задание №2

Листинг программы – файл md.h:

```
extern int md1_int_data;
extern char* md1_str_data;
extern char* md1_get_str(int n);
extern int md1_factorial(int n);
```

Листинг программы – файл md1.c:

```
#include <linux/init.h>
#include <linux/module.h>
#include "md.h"

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Lavrova");
MODULE_DESCRIPTION("Lab_3");

char* md1_str_data = "Привет мир";
int md1_int_data = 64;

extern char* md1_get_str(int n)
{
    printk(KERN_INFO "md1: md1_get_str(%d) called\n", n);
    switch (n)
    {
    case 1:
      return "Первая строка";
      break;
    case 2:
       return "Вторая строка";
      break;
    default:
      return "1 - первая строка, 2 - вторая строка";
      break;
  }
}

extern int md1_factorial(int n)
{
  printk(KERN_INFO "md1: md1_factorial(%d) called\n", n);

  int i, answer = 1;

  if (n <= 0)
    return 0;

  for (i = 2; i <= n; i++)
    answer *= i;

  return answer;
```

```
}

EXPORT_SYMBOL(md1_str_data);
EXPORT_SYMBOL(md1_int_data);
EXPORT_SYMBOL(md1_get_str);
EXPORT_SYMBOL(md1_factorial);

static int __init my_module_init(void)
{
  printk(KERN_INFO "md1: loaded\n");
  return 0;
}

static void __exit my_module_exit(void)
{
  printk(KERN_INFO "md1: unloaded\n");
}

module_init(my_module_init);
module_exit(my_module_exit);
```

Листинг программы – файл md2.c:

```
#include <linux/init.h>
#include <linux/module.h>
#include "md.h"

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Lavrova");
MODULE_DESCRIPTION("Lab_3");

static int __init my_module_init(void)
{
  printk(KERN_INFO "md2: Старт\n");
  printk(KERN_INFO "md2: Число экспортированное из md1: %d\n",
md1_int_data);
  printk(KERN_INFO "md2: Строка экспортированная из md1: %s\n",
md1_str_data);
  printk(KERN_INFO "md2: Результат работы функции
md1_get_str(0): %s\n", md1_get_str(0));
  printk(KERN_INFO "md2: Результат работы функции
md1_get_str(1): %s\n", md1_get_str(1));
  printk(KERN_INFO "md2: Результат работы функции
md1_get_str(2): %s\n", md1_get_str(2));
  printk(KERN_INFO "md2: Результат работы функции
md1_factorial(5): %d\n", md1_factorial(5));
  return 0;
}

static void __exit my_module_exit(void)
{
  printk(KERN_INFO "md2: unloaded\n");
```

```
}

module_init(my_module_init);
module_exit(my_module_exit);
```

Листинг программы – файл md3.c:

```
#include <linux/init.h>
#include <linux/module.h>
#include "md.h"

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Lavrova");
MODULE_DESCRIPTION("Lab_3");

static int __init my_module_init(void)
{
  printk(KERN_INFO "md3: Старт\n");
  printk(KERN_INFO "md3: Число экспортированное из md1: %d\n",
md1_int_data);
  printk(KERN_INFO "md3: Строка экспортированная из md1: %s\n",
md1_str_data);
  printk(KERN_INFO "md3: Результат работы функции
md1_get_str(0): %s\n", md1_get_str(0));
  printk(KERN_INFO "md3: Результат работы функции
md1_get_str(1): %s\n", md1_get_str(1));
  printk(KERN_INFO "md3: Результат работы функции
md1_get_str(2): %s\n", md1_get_str(2));
  printk(KERN_INFO "md3: Результат работы функции
md1_factorial(5): %d\n", md1_factorial(5));
  return -1;
}

module_init(my_module_init);
```

Листинг – Makefile:

```
ifneq ($(KERNELRELEASE),)
        obj-m := md1.o md2.o md3.o

else
        CURRENT = $(shell uname -r)
        KDIR = /lib/modules/$(CURRENT)/build
        PWD = $(shell pwd)

default:
        $(MAKE) -C $(KDIR) M=$(PWD) modules

clean:
        rm -rf .tmp_versions
        rm *.ko
        rm *.o
        rm *.mod.c
        rm *.symvers
        rm *.order
```

```
endif
```

Результаты работы программы:

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod md2.ko
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | tail -4
[10994.695036] md2: Unknown symbol md1_int_data (err 0)
[10994.695053] md2: Unknown symbol md1_factorial (err 0)
[10994.695066] md2: Unknown symbol md1_get_str (err 0)
[10994.695077] md2: Unknown symbol md1_str_data (err 0)
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Unknown symbol in module
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | tail -4
[11052.739782] md3: Unknown symbol md1_int_data (err 0)
[11052.739796] md3: Unknown symbol md1_factorial (err 0)
[11052.739810] md3: Unknown symbol md1_get_str (err 0)
[11052.739821] md3: Unknown symbol md1_str_data (err 0)
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$
```

Рис.4: Т.к. модули md2 и md3 импортируют данные из модуля md1, необходимо сначала загрузить модуль md1. Иначе возникает ошибка.

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod md1.ko
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod md2.ko
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$
```

Рис.5: 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.

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | tail -12
[11089.069262] md1: loaded
[11098.113333] md2: Старт
[11098.113334] md2: Число экспортированное из md1: 64
[11098.113334] md2: Строка экспортированная из md1: Привет мир
[11098.113335] md1: md1_get_str(0) called
[11098.113336] md2: Результат работы функции md1_get_str(0): 1 - первая строка, 2
 - вторая строка
[11098.113336] md1: md1_get_str(1) called
[11098.113336] md2: Результат работы функции md1_get_str(1): Первая строка
[11098.113337] md1: md1_get_str(2) called
[11098.113337] md2: Результат работы функции md1_get_str(2): Вторая строка
[11098.113337] md1: md1_factorial(5) called
[11098.113338] md2: Результат работы функции md1_factorial(5): 120
```

Рис.6: Вывод буфера сообщений ядра в стандартный поток вывода:

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo lsmod | grep md
md2                    16384  0
md1                    16384  1 md2
```

Рис.7: Модуль md1 используется модулем md2

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo rmmod md1
rmmod: ERROR: Module md1 is in use by: md2
```

Рис.8: Пока модуль md1 используется другим модулем, его нельзя выгрузить

```
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo rmmod md2
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo rmmod md1
parallels@parallels-Parallels-Virtual-Platform:~/lab_3$ sudo dmesg | tail -2
[11578.106581] md2: unloaded
[11582.219130] md1: unloaded
```

Рис. 9: Выгружаем модули в обратном порядке. Если сначала выгрузить md2, который импортирует данные из md1, то можно будет выгрузить и md1