



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: Гасанзаде М.А.,

Группа: ИУ7-66Б

Москва, 2020 г.

## ОГЛАВЛЕНИЕ

ЗАДАНИЕ 1 .....	3
Листинг 1 - Код модуля md .....	3
Результат работы программы .....	3
ЗАДАНИЕ 2 .....	7
Модуль md1 .....	7
Листинг 2 – Код модуля md1 .....	7
Результат работы программы .....	9
Модуль md2 .....	9
Листинг 3 – Код модуля md2 .....	9
Результат работы программы .....	10
Модуль md3 .....	11
Листинг 4 – Код модуля md3 .....	11
Результат работы программы .....	12

## ЗАДАНИЕ 1

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

### Листинг 1 - Код модуля md

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

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Hasanzade M.A.");

static int __init md_init(void) {
    printk(KERN_INFO "Module is loaded.\n");
    struct task_struct *task = &init_task;
    do {
        printk(KERN_INFO " |comm: %15s|pid: %8d|ppid: %15d|\n", task->comm, task->pid, task->parent->comm, task->parent->pid);
        } while ((task = next_task(task)) != &init_task);

    printk(KERN_INFO " [current]|comm: %15s|pid: %8d|ppid: %15d|\n", current->comm, current->pid, current->parent->comm, current->parent->pid);
    return 0;
}

static void __exit md_exit(void) {
    printk(KERN_INFO " Module is unloaded.\n");
}

module_init(md_init);
module_exit(md_exit);
```

### Результат работы программы

```
wizard@wizard-VM:~/Desktop/Lab_3$ sudo insmod md.ko
[sudo] password for wizard:
```

.....

```

[ 1333.308498] Module is loaded.
[ 1333.308502] |comm:      swapper/0|pid:      0|ppid:      -1528741392|
[ 1333.308506] |comm:      systemd|pid:      1|ppid:      -1528741392|
[ 1333.308509] |comm:      kthreadd|pid:      2|ppid:      -1528741392|
[ 1333.308512] |comm:      rcu_gp|pid:      3|ppid:      1876400|
[ 1333.308516] |comm:      rcu_par_gp|pid:      4|ppid:      1876400|
[ 1333.308519] |comm:      kworker/0:0H|pid:      6|ppid:      1876400|
[ 1333.308561] |comm:      mm_percpu_wq|pid:      9|ppid:      1876400|
[ 1333.308565] |comm:      ksoftirqd/0|pid:      10|ppid:      1876400|
[ 1333.308568] |comm:      rcu_sched|pid:      11|ppid:      1876400|
[ 1333.308571] |comm:      migration/0|pid:      12|ppid:      1876400|
[ 1333.308574] |comm:      idle_inject/0|pid:      13|ppid:      1876400|
[ 1333.308578] |comm:      cpuhp/0|pid:      14|ppid:      1876400|
[ 1333.308581] |comm:      cpuhp/1|pid:      15|ppid:      1876400|
[ 1333.308584] |comm:      idle_inject/1|pid:      16|ppid:      1876400|
[ 1333.308587] |comm:      migration/1|pid:      17|ppid:      1876400|
[ 1333.308590] |comm:      ksoftirqd/1|pid:      18|ppid:      1876400|
[ 1333.308594] |comm:      kworker/1:0H|pid:      20|ppid:      1876400|
[ 1333.308597] |comm:      kdevtmpfs|pid:      21|ppid:      1876400|
[ 1333.308600] |comm:      netns|pid:      22|ppid:      1876400|
[ 1333.308603] |comm:      rcu_tasks_kthre|pid:      23|ppid:      1876400|
[ 1333.308607] |comm:      kauditd|pid:      24|ppid:      1876400|
[ 1333.308610] |comm:      khungtaskd|pid:      25|ppid:      1876400|
[ 1333.308613] |comm:      oom_reaper|pid:      26|ppid:      1876400|
[ 1333.308617] |comm:      writeback|pid:      27|ppid:      1876400|
[ 1333.308620] |comm:      kcompactd0|pid:      28|ppid:      1876400|
[ 1333.308623] |comm:      ksmd|pid:      29|ppid:      1876400|
[ 1333.308626] |comm:      khugepaged|pid:      30|ppid:      1876400|
[ 1333.308630] |comm:      kworker/u4:1|pid:      32|ppid:      1876400|

```

```

[ 1333.308633] | comm:      kintegrityd|pid:      123|ppid:      1876400|
[ 1333.308636] | comm:      kblockd|pid:      124|ppid:      1876400|
[ 1333.308639] | comm:    blkcg_punt_bio|pid:      125|ppid:      1876400|
[ 1333.308643] | comm:      tpm_dev_wq|pid:      126|ppid:      1876400|
[ 1333.308646] | comm:      ata_sff|pid:      127|ppid:      1876400|
[ 1333.308649] | comm:      md|pid:      128|ppid:      1876400|
[ 1333.308652] | comm:    edac-poller|pid:      129|ppid:      1876400|
[ 1333.308655] | comm:    devfreq_wq|pid:      130|ppid:      1876400|
[ 1333.308659] | comm:    watchdogd|pid:      131|ppid:      1876400|
[ 1333.308662] | comm:    kswapd0|pid:      135|ppid:      1876400|
[ 1333.308665] | comm:    kworker/u5:0|pid:      136|ppid:      1876400|
[ 1333.308668] | comm:  ecryptfs-kthrea|pid:      137|ppid:      1876400|
[ 1333.308692] | comm:    kthrotld|pid:      140|ppid:      1876400|
[ 1333.308715] | comm:  acpi_thermal_pm|pid:      141|ppid:      1876400|
[ 1333.308718] | comm:    scsi_eh_0|pid:      142|ppid:      1876400|
[ 1333.308721] | comm:    scsi_tmf_0|pid:      143|ppid:      1876400|
[ 1333.308725] | comm:    scsi_eh_1|pid:      144|ppid:      1876400|
[ 1333.308728] | comm:    scsi_tmf_1|pid:      145|ppid:      1876400|
[ 1333.308731] | comm:    ipv6_addrconf|pid:      148|ppid:      1876400|
[ 1333.308734] | comm:    kstrp|pid:      159|ppid:      1876400|
[ 1333.308738] | comm:    kworker/u4:4|pid:      174|ppid:      1876400|
[ 1333.308741] | comm:  charger_manager|pid:      179|ppid:      1876400|
[ 1333.308744] | comm:    kworker/1:1H|pid:      180|ppid:      1876400|
[ 1333.308748] | comm:    kworker/0:2|pid:      181|ppid:      1876400|
[ 1333.308751] | comm:    scsi_eh_2|pid:      218|ppid:      1876400|
[ 1333.308754] | comm:    scsi_tmf_2|pid:      219|ppid:      1876400|
[ 1333.308757] | comm:    kworker/0:1H|pid:      244|ppid:      1876400|
[ 1333.308761] | comm:    jbd2/sda2-8|pid:      245|ppid:      1876400|
[ 1333.308764] | comm:  ext4-rsv-conver|pid:      246|ppid:      1876400|
[ 1333.308767] | comm:  systemd-journal|pid:      284|ppid:      1870448|
[ 1333.308770] | comm:    systemd-udevd|pid:      306|ppid:      1870448|
[ 1333.308773] | comm:      loop0|pid:      313|ppid:      1876400|
[ 1333.308777] | comm:      loop1|pid:      314|ppid:      1876400|
[ 1333.308780] | comm:      loop2|pid:      316|ppid:      1876400|
[ 1333.308783] | comm:      loop3|pid:      317|ppid:      1876400|
[ 1333.308786] | comm:      loop4|pid:      318|ppid:      1876400|

[ 1333.309202] | comm:      gedit|pid:      2753|ppid:      -759297424|
[ 1333.309205] | comm:      sudo|pid:      3189|ppid:      1061900336|
[ 1333.309208] | comm:      insmod|pid:      3190|ppid:      -1998303184|
[ 1333.309212] | comm:    systemd-udevd|pid:      3191|ppid:      -567303760|
[ 1333.309215] | [current] | comm:      insmod|pid:      3190|ppid:      -1998303184|

```

Рис. 1 – сборка и информация модуля ядра

```
wizard@wizard-VM:~/Desktop/Lab_3$ sudo rmmod md  
wizard@wizard-VM:~/Desktop/Lab_3$ dmesg | tail -2  
[ 600.384709] ---insmod-2277, parent sudo-2276  
[ 1194.728954] Module is unloaded.
```

Рис. 2 – выгрузка модуля ядра

## 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.

### Модуль md1

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

- Экспортируемые строковые (char \*) и численные (int) данные.
- Экспортируемые функции, возвращающие строковые и числовые значения.

Например:

- Функция, возвращающая в зависимости от переданного целочисленного параметра различные строки (на усмотрение студента);
- Функция, производящая подсчет факториала переданного целочисленного параметра;
- Функция, возвращающая 0.

### Листинг 2 – Код модуля md1

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

MODULE_LICENSE( "GPL" );
MODULE_AUTHOR( "Hasanzade M.A" );
```

```

char* mdl_str_data = "Hello world!";
int mdl_int_data = 42;

extern char* mdl_get_str(int n) {
    printk( "+ mdl: mdl_get_str() called!\n" );
    switch (n){
        case 1:
            return "Hello world!\n";
            break;
        case 2:
            return "Привет Мир!\n";
            break;
        default:
            return "Передайте 1 для получения английского
сообщения или 2 для получения русского.\n";
            break;
    }
}

extern int mdl_factorial(int n) {
    int i, ans;
    ans = 1;
    printk( "+ mdl: mdl_factorial() called!\n" );
    for (i = 2; i <= n; i++) ans *= i;
    return ans;
}

extern int mdl_return_zero(void){
    printk( "+ mdl: mdl_return_zero() called!\n" );
    return 0;
}

EXPORT_SYMBOL(mdl_str_data);
EXPORT_SYMBOL(mdl_int_data);

EXPORT_SYMBOL(mdl_get_str);
EXPORT_SYMBOL(mdl_factorial);
EXPORT_SYMBOL(mdl_return_zero);

static int __init mdl_init( void ) {
    printk( "+ mdl: Module mdl is loaded!\n" );
    return 0;
}

static void __exit mdl_exit( void ) {

```



```

    printk( "+ md1: Module md1 is unloaded!\n" );
}

module_init( md1_init );
module_exit( md1_exit );

```

### Результат работы программы

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md1.ko
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -1
[ 2899.533599] + md1: Module md1 is loaded!

```

Рис. 3 – загрузка и информация от модуля ядра

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo rmmod md1.ko
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -1
[ 2930.768211] + md1: Module md1 is unloaded!

```

Рис. 4 – выгрузка модуля ядра

### Модуль md2

Модуль md2 демонстрирует использование данных и функций 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.

### Листинг 3 – Код модуля md2

```

#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Hasanzade M.A");

extern char* md1_str_data;
extern int md1_int_data;
extern char* md1_get_str(int num);
extern int md1_factorial(int num);
extern int md1_return_zero(void);

```

```

static int __init md2_init(void) {
    printk(KERN_INFO "+md2: Module 2 is loaded.\n");

    printk(KERN_INFO "+ md2: Export char* from MD1
- %s.\n", md1_str_data);
    printk(KERN_INFO "+ md2: Export int from MD1 -
%d.\n", md1_int_data);

    printk(KERN_INFO "+ md2: Function
'md1_get_str(1)' from MD1 return %s.\n",
md1_get_str(1));
    printk(KERN_INFO "+ md2: Function
'md1_factorial(5)' from MD1 return %d.\n",
md1_factorial(5));
    printk(KERN_INFO "+ md2: Function
'md1_return_zero' from MD1 return %d.\n",
md1_return_zero());
    return 0;
}

static void __exit md2_exit(void) {
    printk(KERN_INFO "+ md2: Module 2 is unloaded.\n");
}

module_init(md2_init);
module_exit(md2_exit);

```

### Результат работы программы

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md2.ko
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -10
[ 2711.757246] +md2: Module 2 is loaded.
[ 2711.757251] + md2: Export char* from MD1 - Hello world!.
[ 2711.757253] + md2: Export int from MD1 - 42.
[ 2711.757256] + md1: md1_get_str() called!
[ 2711.757258] + md2: Function 'md1_get_str(1)' from MD1 return Hello world!
.
[ 2711.757260] + md1: md1_factorial() called!
[ 2711.757263] + md2: Function 'md1_factorial(5)' from MD1 return 120.
[ 2711.757266] + md1: md1_return_zero() called!
[ 2711.757268] + md2: Function 'md1_return_zero' from MD1 return 0.

```

Рис. 5 – загрузка и информация от модуля ядра

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md2.ko
[sudo] password for wizard:
insmod: ERROR: could not insert module md2.ko: Unknown symbol in module
wizard@wizard-VM:~/Desktop/Lab_3/part_2$

```

Рис. 6 – ситуация, когда данные, к которым обращается модуль не доступны.

```
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo rmmod md2
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -1
[ 2760.044781] + md2: Module 2 is unloaded.
```

Рис. 7 – выгрузка модуля ядра

## Модуль md3

Модуль md3 демонстрирует сценарий 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.

### Листинг 4 – Код модуля md3

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

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Hasanzade M.A");

extern char* md1_str_data;
extern int md1_int_data;
extern char* md1_get_str(int num);
extern int md1_factorial(int num);
extern int md1_return_zero(void);

static int __init md3_init(void) {
    printk(KERN_INFO "+md3: Module 3 is loaded.\n");

    printk(KERN_INFO "+ md3: Export char* from MD1 - %s.\n", md1_str_data);
    printk(KERN_INFO "+ md3: Export int from MD1 - %d.\n", md1_int_data);

    printk(KERN_INFO "+ md3: Function 'md1_get_str(1)' from MD1 return %s.\n",
md1_get_str(1));
```

```

        printk(KERN_INFO "+ md3: Function
'md1_factorial(5)' from MD1 return %d.\n",
md1_factorial(5));
        printk(KERN_INFO "+ md3: Function
'md1_return_zero' from MD1 return %d.\n",
md1_return_zero());
        return 0;
}

static void __exit md3_exit(void) {
    printk(KERN_INFO "+ md3: Module 3 is unloaded.\n");
}

module_init(md3_init);
module_exit(md3_exit);

```

### Результат работы программы

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Unknown symbol in module

```

Рис. 8 – ситуация, когда данные, к которым обращается модуль не доступны

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md3.ko
insmod: ERROR: could not insert module md3.ko: Operation not permitted

```

Рис. 9 – Ошибка при значении `return -1;` (Листинг 4). Нужно поменять на значение 0.

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo insmod md3.ko
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -10
[ 3609.648059] +md3: Module 3 is loaded.
[ 3609.648064] + md3: Export char* from MD1 - Hello world!.
[ 3609.648067] + md3: Export int from MD1 - 42.
[ 3609.648069] + md1: md1_get_str() called!
[ 3609.648072] + md3: Function 'md1_get_str(1)' from MD1 return Hello world!
.
[ 3609.648074] + md1: md1_factorial() called!
[ 3609.648077] + md3: Function 'md1_factorial(5)' from MD1 return 120.
[ 3609.648079] + md1: md1_return_zero() called!
[ 3609.648082] + md3: Function 'md1_return_zero' from MD1 return 0.
wizard@wizard-VM:~/Desktop/Lab_3/part_2$

```

Рис. 10 – Успешная загрузка модуля и информация, записанная им.

```

wizard@wizard-VM:~/Desktop/Lab_3/part_2$ sudo rmmod md3
wizard@wizard-VM:~/Desktop/Lab_3/part_2$ dmesg | tail -1
[ 3728.501643] + md3: Module 3 is unloaded.

```

Рис. 11 – выгрузка модуля ядра