



**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 № 8
По курсу «Операционные системы»**

Тема Создание виртуальной файловой системы

Студент Неклепаева А. Н.

Группа ИУ7-63б

Преподаватель Рязанова Н. Ю.

Москва.
2020 г.

Листинг кода

```
1  #include <linux/module.h>
2  #include <linux/kernel.h>
3  #include <linux/init.h>
4  #include <linux/fs.h>
5  #include <linux/time.h>
6  #include <linux/slab.h>
7
8  MODULE_LICENSE( "GPL" );
9  MODULE_AUTHOR( "Anastasia_Neklepaeva" );
10
11 #define MAGIC_NUMBER 0x13131313
12
13 #define SLABNAME "my_cache"
14 struct kmem_cache *cache = NULL;
15 static int number = 7;
16 module_param(number, int, 0);
17
18 struct myfs_inode
19 {
20     int i_mode;
21     unsigned long i_ino;
22 };
23
24 static int size = sizeof(struct myfs_inode);
25
26 // деструктор суперблока
27 static void myfs_put_super(struct super_block * sb)
28 {
29     printk(KERN_DEBUG "MYFS_super_block_destroyed!\n" );
30 }
31
32 // put_super хранит деструктор суперблока
33 static struct super_operations const myfs_super_ops =
34 {
35     .put_super = myfs_put_super,
36     .statfs = simple_statfs,
37     .drop_inode = generic_delete_inode,
38 };
39
40
41 // размещение новой структуры inode
42 // и заполнение ее значениями: размером и временами
43 static struct inode *myfs_make_inode(struct super_block * sb, int mode)
44 {
45     struct inode *ret = new_inode(sb);
46     struct myfs_inode *my_inode;
47
48     if (ret)
49     {
```

```

50     inode_init_owner(ret, NULL, mode);
51     ret->i_size = PAGE_SIZE;
52     ret->i_atime = ret->i_mtime = ret->i_ctime = current_time(ret);
53
54     my_inode = kmem_cache_alloc(cache, GFP_KERNEL);
55
56     my_inode->i_mode = ret->i_mode;
57     my_inode->i_ino = ret->i_ino;
58
59     ret->i_private = my_inode;
60 }
61 return ret;
62 }
63
64 // инициализация суперблока, построение корневого каталога
65 static int myfs_fill_sb(struct super_block* sb, void* data, int silent)
66 {
67     struct inode* root = NULL;
68     sb->s_blocksize = PAGE_SIZE;
69     sb->s_blocksize_bits = PAGE_SHIFT;
70     sb->s_magic = MAGIC_NUMBER;
71     sb->s_op = &myfs_super_ops;
72
73     // создание inode для корневого каталога
74     root = myfs_make_inode(sb, S_IFDIR | 0755);
75     if (!root)
76     {
77         printk(KERN_ERR "MYFS_inode_allocation_failed!\n");
78         return -ENOMEM;
79     }
80
81     root->i_op = &simple_dir_inode_operations;
82     root->i_fop = &simple_dir_operations;
83     sb->s_root = d_make_root(root);
84
85     if (!sb->s_root)
86     {
87         printk(KERN_ERR "MYFS_root_creation_failed!\n");
88         iput(root);
89         return -ENOMEM;
90     }
91
92     return 0;
93 }
94
95 // функция возвращает структуру, описывающую корневой каталог файловой системы
96 static struct dentry* myfs_mount(struct file_system_type *type, int flags,
97                                 char const *dev, void *data)
98 {
99     struct dentry* const entry = mount_nodev(type, flags, data, myfs_fill_sb);
100     if (IS_ERR(entry))
101         printk(KERN_ERR "MYFS_mounting_failed!\n");

```

```

102     else
103         printk(KERN_DEBUG "MYFS_ mounted!\n") ;
104     return entry;
105
106 }
107
108 // структура, "описывающая" создаваемую файловую систему
109 // owner отвечает за счетчик ссылок на модуль
110 // name хранит название файловой системы
111 static struct file_system_type myfs_type =
112 {
113     .owner = THIS_MODULE,
114     .name = "myfs",
115     .mount = myfs_mount,
116     .kill_sb = kill_anon_super,
117 };
118
119 static int __init myfs_init(void)
120 {
121     // регистрация файловой системы
122     int ret = register_filesystem(& myfs_type);
123     if(ret != 0)
124     {
125         printk(KERN_ERR "MYFS_MODULE_ cannot_ register_ filesystem!\n");
126         return ret;
127     }
128
129     cache = kmem_cache_create(SLABNAME, size, 0, SLAB_HWCACHE_ALIGN, NULL);
130
131     if (!cache)
132     {
133         printk( KERN_ERR "kmem_cache_create_ error\n");
134         kmem_cache_destroy(cache);
135         return -ENOMEM;
136     }
137
138     printk(KERN_DEBUG "MYFS_MODULE_ loaded!\n");
139     return 0;
140 }
141
142
143
144 static void __exit myfs_exit(void)
145 {
146     int ret;
147
148     // deregistration файловой системы
149     ret = unregister_filesystem(&myfs_type);
150     if (ret != 0)
151         printk(KERN_ERR "MYFS_MODULE_ cannot_ unregister_ filesystem_!\n");
152     printk(KERN_DEBUG "MYFS_MODULE_ unloaded!\n");
153

```

```

154 |     kmem_cache_destroy(cache);
155 | }
156 |
157 | module_init( myfs_init );
158 | module_exit( myfs_exit );

```

Демонстрация работы

Загрузка модуля ядра

```

anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ sudo insmod lab_08.
ko
[sudo] пароль для anastasia:
anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ lsmod
Module                Size  Used by
lab_08                 16384  0

```

Создание образа диска и каталога, который является точкой монтирования файловой системы

```

anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ touch image
anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ mkdir dir

```

Монтирование виртуальной файловой системы, проверка системного лога

```

anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ sudo mount -o loop
-t myfs ./image ./dir
anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ dmesg | grep MYFS
[ 151.026341] MYFS_MODULE loaded !
[ 462.131687] MYFS mounted!

```

Вывод информации о виртуальной файловой системе

```

anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ mount | grep myfs
/dev/loop47 on /home/anastasia/bmstu/sem_6/os/lab_08/dir type myfs (rw,relatime)

```

Размонтирование виртуальной файловой системы, выгрузка модуля, проверка системного лога

```

anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ sudo umount ./dir
anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ sudo rmmod lab_08
anastasia@anastasia-Swift-SF314-54G:~/bmstu/sem_6/os/lab_08$ dmesg | grep MYFS
[ 151.026341] MYFS_MODULE loaded !
[ 462.131687] MYFS mounted!
[ 800.316233] MYFS super block destroyed!
[ 821.016469] MYFS_MODULE unloaded !

```