



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»

Отчет по лабораторной работе №8

Дисциплина: Операционные системы

Тема: Создание виртуальной файловой системы

Студент. ИУ7-64Б
(Группа)

(Подпись, дата)

Т. Р. Садулаева
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

Н.Ю. Рязанова
(И.О.Фамилия)

2021 г.

Код программы

```
#include <linux/module.h>
#include <linux/fs.h>
#include <linux/slab.h>

#define VFS_MAGIC_NUMBER 0x13131313;
#define SLABNAME "my_cache"

static int sco = 0;
static struct kmem_cache *cache = NULL;
static void* line = NULL;

static void VFS_put_sup(struct super_block *sup_block)
{
    printk(KERN_DEBUG "VFS super block destroyed\n");
}

static struct super_operations const VFS_super_ops = {
    .put_super = VFS_put_sup,
    .statfs = simple_statfs,
    .drop_inode = generic_delete_inode,
};

struct VFS_inode
{
    int i_mode; unsigned
    long i_ino;
} VFS_inode;

static struct inode *VFS_create_inode(struct super_block *sup_block, int mode)
{
    struct inode *root = new_inode(sup_block);

    if (root)
    {
        inode_init_owner(root, NULL, mode);
        root->i_size = PAGE_SIZE;
        root->i_atime = current_time(root);
        root->i_mtime = current_time(root);
        root->i_ctime = current_time(root);
        root->i_private = &VFS_inode;
    }
    return root;
}

static int VFS_fill_sup_block(struct super_block *sup_block, void *data, int silent)
{
    struct inode *root = NULL;

    sup_block->s_blocksize = PAGE_SIZE; sup_block->
    >s_blocksize_bits = PAGE_SHIFT; sup_block->
```

```

>s_magic = VFS_MAGIC_NUMBER; sup_block->s_op =
&VFS_super_ops;
root = VFS_create_inode(sup_block, S_IFDIR|0755);

if (!root)
{
    printk(KERN_ERR "VFS inode allocation failed\n");
    return -ENOMEM;
}

root->i_op = &simple_dir_inode_operations;
root->i_fop = &simple_dir_operations;
sup_block->s_root = d_make_root(root);

if (!sup_block->s_root)
{
    printk(KERN_ERR "VFS root creation failed\n");
    iput(root);
    return -ENOMEM;
}
return 0;
}

static struct dentry* VFS_mount(struct file_system_type * type, int flags, char const
*dev, void
*data)
{
    struct dentry *const entry = mount_nodev(type, flags, data, VFS_fill_sup_block);
    if (IS_ERR(entry))
        printk(KERN_ERR "VFS mounting failed!\n");
    else
        printk(KERN_DEBUG "VFS mounted\n");
    return entry;
}

static struct file_system_type VFS_type = {
    .owner = THIS_MODULE,
    .name = "VFS",
    .mount = VFS_mount,
    .kill_sb = kill_block_super,
};

void co (void *p)
{
    *(int *)p = (int)p; sco++;
}

static int ____init VFS_init(void)
{
    int ret;

    line = kmem_cache_alloc(cache, GFP_KERNEL);
    if (!line)

```

```

{
    printk(KERN_ERR "VFS kmem_cache_alloc error\n");
    kmem_cache_free(cache, line);
}

cache = kmem_cache_create(SLABNAME, sizeof(struct VFS_inode), 0, 0, co);
if (!cache)
{
    printk(KERN_ERR "VFS_MODULE cannot allocate cache\n");
    kmem_cache_destroy(cache);
    return -ENOMEM;
}

ret = register_filesystem(&VFS_type);
if (ret != 0)
{
    printk(KERN_ERR "VFS_MODULE cannot register filesystem\n");
    return ret;
}

printk(KERN_INFO "VFS_MODULE filesystem loaded\n");
return 0;
}

static void ____exit VFS_exit(void)
{
    int ret; kmem_cache_free(cache,
    line);
    kmem_cache_destroy(cache);

    ret = unregister_filesystem(&VFS_type);
    if (ret != 0)
        printk(KERN_ERR "VFS_MODULE cannot unregister filesystem!\n");
    printk(KERN_INFO "VFS_MODULE unloaded\n");
}

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Sadulaeva Teona");
module_init(VFS_init);
module_exit(VFS_exit);

```

Демонстрация работы программы

Загрузка модуля ядра

```
TM1613:~/8$ sudo insmod vfs.ko
```

```
TM1613:~/8$ lsmod
Module              Size  Used by
vfs                  16384  0
```

Создание образа диска и каталога

```
TM1613:~/8$ touch image
```

```
TM1613:~/8$ mkdir dir
```

Монтирование виртуальной файловой системы

```
TM1613:~/8$ sudo mount -o loop -t VFS ./image ./dir
```

```
TM1613:~/8$ dmesg | tail -2
[ 70.077554] VFS_MODULE filesystem loaded
[ 226.160985] VFS mounted
```

Файловая система в файловой подсистеме:

```
TM1613:~/8$ ls -la
```

```
4096 июн 14 10:56 .
4096 июн 14 10:51 ..
4096 июн 12 21:58 dir
  0 июн 12 21:58 image
1581 июн 14 10:48 Makefile
  0 июн 14 10:48 Module.symvers
3258 июн 14 10:48 vfs.c
10320 июн 14 10:29 vfs.ko
  24 июн 14 10:29 vfs.mod
```