



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 предмету: «Операционные системы»

**Тема: Создание виртуальной файловой
системы.**

Преподаватель: Рязанова Н. Ю.

Студент: Гасанзаде М.А.,

Группа: ИУ7-66Б

Москва, 2020 г.

ЛИСТИНГ

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

#define MYFS_MAGIC_NUMBER 0x13131313;
#define SLABNAME "my_cache"
struct myfs_inode
{
    int i_mode;
    unsigned long i_ino;
};

int inode_number = 0;
static struct kmem_cache *cache;

static void myfs_put_super(struct super_block *sb)
{
    printk(KERN_DEBUG "MYFS super block destroyed\n");
}

int free_allocated_inodes(struct inode *inode)
{
    kmem_cache_free(cache, inode->i_private);
    return 1;
}

static struct super_operations const myfs_super_ops = {
    .put_super = myfs_put_super,
    .statfs = simple_statfs,
    .drop_inode = free_allocated_inodes,
};

static struct inode *myfs_make_inode(struct super_block *sb, int mode)
{
    struct inode *ret = new_inode(sb);
    if (ret)
    {
        struct myfs_inode *my_inode = kmem_cache_alloc(cache, GFP_KERNEL);
        inode_init_owner(ret, NULL, mode);
        *my_inode = (struct myfs_inode){
            .i_mode = ret->i_mode,
            .i_ino = ret->i_ino
        };
        ret->i_size = PAGE_SIZE;
        ret->i_atime = ret->i_mtime = ret->i_ctime = current_time(ret);
        ret->i_private = my_inode;
    }
    return ret;
}

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

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

```

sb->s_magic = MYFS_MAGIC_NUMBER;
sb->s_op = &myfs_super_ops;

root = myfs_make_inode(sb, S_IFDIR|0755);
if (!root)
{
    printk(KERN_ERR "MYFS inode allocation failed\n");
    return -ENOMEM;
}
root->i_op = &simple_dir_inode_operations;
root->i_fop = &simple_dir_operations;

sb->s_root = d_make_root(root);

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

return 0;
}

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

static struct file_system_type myfs_type = {
    .owner = THIS_MODULE,
    .name = "myfs",
    .mount = myfs_mount,
    .kill_sb = kill_block_super,
};

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

static int __init myfs_init(void)
{
    int ret = register_filesystem(&myfs_type);
    cache = kmem_cache_create(SLABNAME, sizeof(struct myfs_inode), 0, 0,
co);
    if (ret != 0)
    {
        printk(KERN_ERR "MYFS can't register filesystem\n");
        return ret;
    }
    printk(KERN_INFO "MYFS filesystem registered");
    return 0;
}

```

```
static void __exit myfs_exit(void)
{
    int ret = unregister_filesystem(&myfs_type);
    if (ret != 0)
        printk(KERN_ERR "MYFS can't unregister filesystem!\n");
    kmem_cache_destroy(cache);
    printk(KERN_INFO "MYFS unregistered %d", inode_number);
}

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

module_init(myfs_init);
module_exit(myfs_exit);
```

РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

Для создания виртуальной файловой системы необходимо скомпилировать загружаемый модуль ядра и загрузить его с помощью команды `insmod`.

```
wizard@wizard-VM:~/Desktop/Lab_8$ sudo make
make -C /lib/modules/5.3.0-51-generic/build M=/home/wizard/Desktop/Lab_8 modules
make[1]: Entering directory '/usr/src/linux-headers-5.3.0-51-generic'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.3.0-51-generic'
```

Рисунок 1 - сборка модуля ядра

```
wizard@wizard-VM:~/Desktop/Lab_8$ sudo insmod Lab_8.ko
wizard@wizard-VM:~/Desktop/Lab_8$ lsmod | grep Lab_8
Lab_8                  16384  0
wizard@wizard-VM:~/Desktop/Lab_8$
```

Рисунок 2 - загрузка модуля ядра.

Создадим образ диска (`touch image`) и корень файловой системы (`mkdir dir`).

Смонтируем (`sudo mount -o loop -t myfs ./image ./dir`) файловую систему, размонтируем (`sudo umount ./dir`) ее и посмотрим в `syslog`.

```
wizard@wizard-VM:~/Desktop/Lab_8$ sudo mount -o loop -t myfs ./image ./labdir/
wizard@wizard-VM:~/Desktop/Lab_8$ tail -f /var/log/syslog | grep MYFS
May 12 23:52:13 wizard-VM kernel: [ 1097.665909] MYFS filesystem registered
^C
wizard@wizard-VM:~/Desktop/Lab_8$ sudo umount ./labdir
wizard@wizard-VM:~/Desktop/Lab_8$ tail -f /var/log/syslog | grep MYFS
May 12 23:52:13 wizard-VM kernel: [ 1097.665909] MYFS filesystem registered
May 12 23:52:48 wizard-VM kernel: [ 1301.534175] MYFS mounted
May 12 23:52:48 wizard-VM kernel: [ 1336.729573] MYFS super block destroyed
```

Рисунок 3 – Создание образа диска (`touch image`)
Корня файловой системы (`mkdir labdir`)
Монтирования/размонтирования системы.
Вывод отладочной информации в `syslog`

```
wizard@wizard-VM:~/Desktop/Lab_8$ sudo cat /proc/slabinfo | grep my_cache
my_cache      0      0    24 170    1 : tunables    0    0    0 : slabdata    0    0    0
my_cache      0      0    24 170    1 : tunables    0    0    0 : slabdata    0    0    0
wizard@wizard-VM:~/Desktop/Lab_8$
```

Рисунок 3 - Состояние Slab кэша до монтирования системы.

```
wizard@wizard-VM:~/Desktop/Lab_8$ sudo cat /proc/slabinfo | grep my_cache
my_cache     170    170    24 170    1 : tunables    0    0    0 : slabdata    1    1    0
my_cache      0      0    24 170    1 : tunables    0    0    0 : slabdata    0    0    0
```

Рисунок 4 - Состояние Slab кэша после монтирования системы.

```
wizard@wizard-VM:~/Desktop/Lab_8$ mount | grep myfs  
/home/wizard/Desktop/Lab_8/image on /home/wizard/Desktop/Lab_8/labdir type myfs (rw,relatime)  
wizard@wizard-VM:~/Desktop/Lab_8$
```

Рисунок 5 - Вывод информации о монтированной файловой системе

ВЫВОД

В результате проделанной лабораторной работы, были получены навыки создания, а также монтирования виртуальной файловой системы. В дополнении была изучена возможность создания slab кэша для inode.