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»

О Т Ч Е Т

по лабораторной работе № ____

Дисциплина: *Операционные системы*

Студент	***ИУ7И-66Б***		**Нгуен Ф. С.**
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			**Рязанова Н. Ю.**
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

код программы

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/vmalloc.h>
#include <linux/proc_fs.h>
#include <asm/uaccess.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Nguyensanghso@gmail.com");


#define OK 0

#define FORTUNE_DIRNAME "Fortdir"
#define FORTUNE_FILENAME "fortfile"
#define FORTUNE_SYMLINK "fortune_symlink"
#define FORTUNE_PATH FORTUNE_DIRNAME "/" FORTUNE_FILENAME

#define MAX_COOKIE_BUF_SIZE PAGE_SIZE

#define KERN_LOG_MSG() { printk(KERN_INFO "FORTUNE_MODULE: %s called.\n",
__func__); }
#define KERN_ERR_MSG(err) { printk(KERN_ERR "FORTUNE_MODULE: %s.\n", err); }
#define KERN_INFO_MSG(msg) { printk(KERN_INFO "FORTUNE_MODULE: %s.\n", msg); }

static struct proc_dir_entry *fortune_dir, *fortune_file, *fortune_symlink =
NULL;
static char *cookie_buffer = NULL;

static int read_index = 0;
static int write_index = 0;

char tmp_buffer[MAX_COOKIE_BUF_SIZE];

static int fortune_open(struct inode *sp_inode, struct file *sp_file)
{
    KERN_LOG_MSG();
    return OK;
}

static int fortune_release(struct inode *sp_node, struct file *sp_file)
{
    KERN_LOG_MSG();
    return OK;
}

static ssize_t fortune_write(struct file *file, const char __user *buf, size_t
len, loff_t *ppos)
{
    KERN_LOG_MSG();

    if (len > MAX_COOKIE_BUF_SIZE - write_index + 1)
    {
        KERN_ERR_MSG("Buffer overflow");
        return -ENOSPC;
    }

    if (copy_from_user(&cookie_buffer[write_index], buf, len) != 0)
    {
        KERN_ERR_MSG("copy_from_user function get a error");
        return -EFAULT;
    }
```

```
    write_index += len;
    cookie_buffer[write_index - 1] = '\0';

    return len;
}

static ssize_t fortune_read(struct file *file, char __user *buf, size_t len,
loff_t *f_pos)
{
    KERN_LOG_MSG();

    if (*f_pos > 0 || write_index == 0)
    {
        return 0;
    }

    if (read_index >= write_index)
    {
        read_index = 0;
    }

    int read_len = snprintf(tmp_buffer, MAX_COOKIE_BUF_SIZE, "%s\n",
&cookie_buffer[read_index]);
    if (copy_to_user(buf, tmp_buffer, read_len) != 0)
    {
        KERN_ERR_MSG("copy_to_user function get a error")
        return -EFAULT;
    }

    read_index += read_len;
    *f_pos += read_len;

    return read_len;
}

static const struct proc_ops fops =
{
    proc_read: fortune_read,
    proc_write: fortune_write,
    proc_open: fortune_open,
    proc_release: fortune_release,
};

static void cleanup_fortune(void)
{
    KERN_LOG_MSG();

    if (fortune_symlink != NULL)
    {
        remove_proc_entry(FORTUNE_SYMLINK, NULL);
    }

    if (fortune_file != NULL)
    {
        remove_proc_entry(FORTUNE_FILENAME, fortune_dir);
    }

    if (fortune_dir != NULL)
    {
        remove_proc_entry(FORTUNE_DIRNAME, NULL);
    }

    vfree(cookie_buffer);
}

static int __init fortune_init(void)
{
```

```
    KERN_LOG_MSG();

    if ((cookie_buffer = vzalloc(MAX_COOKIE_BUF_SIZE)) == NULL)
    {
        KERN_ERR_MSG("Allocate memory error.");
        return -ENOMEM;
    }

    if ((fortune_dir = proc_mkdir(FORTUNE_DIRNAME, NULL)) == NULL)
    {
        KERN_ERR_MSG("Error during create directory in proc");
        cleanup_fortune();
        return -ENOMEM;
    }

    if ((fortune_file = proc_create(FORTUNE_FILENAME, 0666, fortune_dir, &fops))
== NULL)
    {
        KERN_ERR_MSG("Error during create file in proc");
        cleanup_fortune();
        return -ENOMEM;
    }

    if ((fortune_symlink = proc_symlink(FORTUNE_SYMLINK, NULL, FORTUNE_PATH)) ==
NULL)
    {
        KERN_ERR_MSG("Error during create symlink in proc");
        cleanup_fortune();
        return -ENOMEM;
    }

    KERN_INFO_MSG("Module has benn successfully loaded.\n");
    return OK;
}

static void __exit fortune_exit(void)
{
    KERN_LOG_MSG();
    cleanup_fortune();
    KERN_INFO_MSG("Module has been successfully removed");
}

module_init(fortune_init);
module_exit(fortune_exit);
```

```
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ sudo insmod fortune.ko
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ lsmod | head -5
Module                  Size  Used by
fortune                20480  0
nls_utf8               16384  1
isofs                  49152  1
rfcomm                 81920  4
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ echo "This is First message" > /proc/Fortdir/fortfile
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ echo "This is Second  message" > /proc/Fortdir/fortfile
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ echo "This is 3th  message" > /proc/Fortdir/fortfile
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ cat /proc/Fortdir/fortfile
This is First message
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ cat /proc/Fortdir/fortfile
This is Second  message
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ cat /proc/Fortdir/fortfile
This is 3th  message
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$
```


nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$ dmesg | tail -10
[667.794057] FORTUNE_MODULE: fortune_read called.
[667.794067] FORTUNE_MODULE: fortune_release called.
[670.350732] FORTUNE_MODULE: fortune_open called.
[670.350742] FORTUNE_MODULE: fortune_read called.
[670.350755] FORTUNE_MODULE: fortune_read called.
[670.350769] FORTUNE_MODULE: fortune_release called.
[672.190172] FORTUNE_MODULE: fortune_open called.
[672.190183] FORTUNE_MODULE: fortune_read called.
[672.190223] FORTUNE_MODULE: fortune_read called.
[672.190239] FORTUNE_MODULE: fortune_release called.
nguyensang@K-virtual-machine:~/Desktop/OS2021/lab4/part2$