

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ  
ТЕХНОЛОГИЙ

## **Лабораторная работа №2**

по дисциплине

«Системы ввода-вывода»

Вариант 1

***Выполнили:***

Векшин А. И. Р3316/CBB 1.2

Дашкевич Е.В. Р3308/CBB 1.2

***Преподаватель:***

Сергей Михайлович Табунщик

# Введение

**Тема:** «Основы написания драйверов устройств с использованием операционной системы»

**Цель:** познакомиться с основами разработки драйверов устройств с использованием операционной системы на примере создания драйверов символьных устройств под операционную систему Linux.

## Текст задания

При записи текста в файл символьного устройства должен осуществляться подсчет введенных символов. Последовательность полученных результатов (количество символов) с момента загрузки модуля ядра должна выводиться при чтении файла.

# Основное задание

## Чтение

```
static ssize_t my_read(struct file *f, char __user *buf, size_t len, loff_t
*off) {
    char str[256];
    ssize_t str_len;
    int i, idx;

    str_len = snprintf(str, sizeof(str), "Total symbols entered: %d\nHistory
(last 10 inputs):\n", counter);

    for (i = 0; i < HISTORY_SIZE; i++) {
        idx = (history_index + i) % HISTORY_SIZE;
        if (history[idx] > 0) {
            str_len += snprintf(str + str_len, sizeof(str) - str_len, "%d\n",
history[idx]);
        }
    }

    if (len < str_len) return -EINVAL;
    if (*off > 0) return 0;
    if (copy_to_user(buf, str, str_len)) return -EFAULT;

    *off += str_len;
    return str_len;
}
```

## Запись

```
static ssize_t my_write(struct file *f, const char __user *buf, size_t len,
loff_t *off) {
    size_t bytes_to_write = (len > BUF_SIZE) ? BUF_SIZE : len;
    int i = 0;

    if (copy_from_user(device_buffer, buf, bytes_to_write))
        return -EFAULT;

    device_buffer[bytes_to_write] = '\0';
    data_len = bytes_to_write;

    while (device_buffer[i] != '\0') {
        i++;
        counter++;
    }

    history[history_index] = i;
    history_index = (history_index + 1) % HISTORY_SIZE;

    printk(KERN_INFO "Driver: wrote %zu symbols\n", i);
    return bytes_to_write;
}
```

## Результаты работы

### Основной сценарий

```
ubuntu@ubuntu:~/code$ sudo cat /dev/var1
Total symbols entered: 0
History (last 10 inputs):
ubuntu@ubuntu:~/code$ echo "aaa" | sudo tee -a /dev/var1
aaa
ubuntu@ubuntu:~/code$ echo "bbbb" | sudo tee -a /dev/var1
bbbb
ubuntu@ubuntu:~/code$ sudo cat /dev/var1
Total symbols entered: 9
History (last 10 inputs):
4
5
ubuntu@ubuntu:~/code$ echo "ccccc" | sudo tee -a /dev/var1
ccccc
ubuntu@ubuntu:~/code$ echo "dddddd" | sudo tee -a /dev/var1
dddddd
ubuntu@ubuntu:~/code$ sudo cat /dev/var1
Total symbols entered: 22
History (last 10 inputs):
4
5
6
7
```

## dmesg

```
[ 817.135815] Hello!  
[ 1076.946739] Driver: open()  
[ 1076.965427] Driver: close()  
[ 1122.651453] Driver: open()  
[ 1122.668214] Driver: wrote 4 symbols  
[ 1122.679000] Driver: close()  
[ 1132.314482] Driver: open()  
[ 1132.325605] Driver: wrote 5 symbols  
[ 1132.326632] Driver: close()  
[ 1136.507696] Driver: open()  
[ 1136.542910] Driver: close()  
[ 1152.620992] Driver: open()  
[ 1152.643262] Driver: wrote 6 symbols  
[ 1152.644194] Driver: close()  
[ 1170.847939] Driver: open()  
[ 1170.863673] Driver: wrote 7 symbols  
[ 1170.864379] Driver: close()  
[ 1180.753560] Driver: open()  
[ 1180.767615] Driver: close()  
[ 1259.030617] Bye!!!
```