



Факультет программной инженерии и компьютерной техники

Лабораторная работа №2
по дисциплине «Системы вводы-вывода»
Вариант – №3

Выполнила:
Студент группы Р3332
Чмурова М.В.

Преподаватель:
Быковский Сергей Вячеславович

Санкт-Петербург
2025

Оглавление

<i>Цель работы</i>	2
<i>Задачи</i>	2
<i>Реализованные функции</i>	3
<i>Выполнение и демонстрация работы</i>	4
<i>Вывод</i>	4

Цель работы

Познакомится с основами разработки драйверов устройств с использованием операционной системы на примере создания драйверов символьных устройств под операционную систему Linux.

Задачи

1. Написать драйвер символьного устройства, удовлетворяющий требованиям:
 - должен создавать символьное устройство `/dev/varN`, где N – это номер варианта
 - должен обрабатывать операции записи и чтения в соответствии с вариантом задания: при записи текста в файл символьного устройства должно запоминаться количество введенных букв (не буквы на считаются). Последовательность полученных результатов с момента загрузки модуля ядра должна выводиться при чтении файла.
2. Оформить отчет по работе в электронном формате

Реализованные функции

Функции `my_read` и `my_write`, вызываемые при чтении и записи

соответственно:

```
static ssize_t my_read(struct file *f, char __user *buf, size_t len, loff_t *off)
{
    char result[256];
    int result_len = 0;
    int i;
    result_len = snprintf(result, sizeof(result), "Total written characters: %d\n",
total_count);
    for (i = 0; i < write_index; i++) {
        result_len += snprintf(result + result_len, sizeof(result) - result_len,
"Write %d: %d characters\n", i + 1, local_counts[i]);
    }
    if (*off >= result_len)
        return 0;
    if (len > result_len - *off)
        len = result_len - *off;
    if (copy_to_user(buf, result + *off, len))
        return -EFAULT;
    *off += len;
    printk(KERN_INFO "Driver: read()\n");
    return len;
}

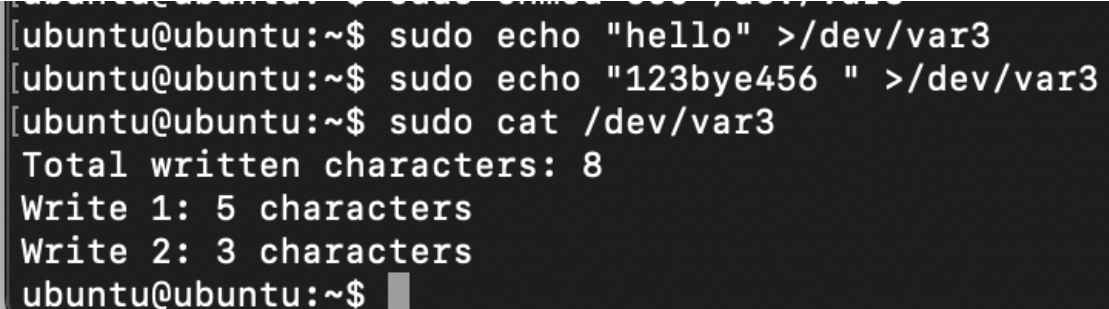
static ssize_t my_write(struct file *f, const char __user *buf, size_t len, loff_t
*off)
{
    char *kbuffer;
    int i = 0;
    int local_count = 0;
    kbuffer = kmalloc(len + 1, GFP_KERNEL);
    if (!kbuffer)
        return -ENOMEM;
    if (copy_from_user(kbuffer, buf, len)) {
        kfree(kbuffer);
        return -EFAULT;
    }
    kbuffer[len] = '\0';
    for (i = 0; i < len; i++) {
        if (isalpha(kbuffer[i]))
            local_count++;
    }
    total_count += local_count;
    if (write_index < 100) {
        local_counts[write_index++] = local_count;
    }
    printk(KERN_INFO "Driver: write()", local_count, total_count);
    kfree(kbuffer);
    return len;
}
```

Выполнение и демонстрация работы

Репозиторий с исходным кодом:

<https://github.com/kkettch/input-output-systems/tree/main/lab2>

Вывод в терминале для ввода “hello” и “123bye456 “:



```
[ubuntu@ubuntu:~$ sudo echo "hello" >/dev/var3
[ubuntu@ubuntu:~$ sudo echo "123bye456 " >/dev/var3
[ubuntu@ubuntu:~$ sudo cat /dev/var3
Total written characters: 8
Write 1: 5 characters
Write 2: 3 characters
ubuntu@ubuntu:~$
```

Рисунок 1. Демонстрация выполнения

Вывод

В ходе данной лабораторной работы я научилась писать свои драйверы для Linux, конкретно был реализован драйвер для подсчета количества введенных букв при записи в файл и при чтении – вывод получившегося числа.