

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования  
"Омский государственный технический университет"

Кафедра «Информатика и вычислительная техника»

**Отчет**  
**по лабораторной работе № 4**  
дисциплина «Операционные системы»

Выполнил: Шмидт А.В.  
гр. ИВТ - 244  
Проверил: асс. Кононова В.В.

*З. Кононова*

*58.*

*02.12.2024*

**Цель работы:**

Изучение многопользовательской блокировки файлов в Linux.

**Задание:**

Разработать программу для Linux, которая должна запускаться двух экземплярах, каждый со своей виртуальной консоли. Программа использует заранее подготовленный текстовый файл. Она открывает этот файл, при невозможности этого действия выдается сообщение и прекращает выполнение. После успешного открытия файла делается попытка установить на весь файл многопользовательскую блокировку по записи. По результатам попытки выполнения блокировки – на экран выдается сообщение о его реализации или текущей невозможности это сделать. При невозможности установить блокировку сразу, программа задает блокировку с ожиданием ее выполнения. По установлении блокировки доступа программа читает из этого файла все находящиеся в нем данные и выводит их на экран. Затем программа делает задержку выполнения («засыпает») на 7 или 8 секунд, после чего снимает блокировку. Сообщения должны выводиться цветные и в середине экрана.

## Исходный код:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <cstring>
#include <errno.h>

using namespace std;

void printCentered(const string& msg, const string& color, int& row) {
    struct winsize w;
    ioctl(STDOUT_FILENO, TIOCGWINSZ, &w);
    int col = (w.ws_col - msg.length()) / 2;
    cout << "\033[" << row << ";" << col << "H" << color << msg << "\033[0m" << endl;
    row++;
}

void printFileContent(int fd, int& row) {
    string content;
    char buffer[1024];
    ssize_t bytes_read;
    while ((bytes_read = read(fd, buffer, sizeof(buffer))) > 0) {
        content.append(buffer, bytes_read);
    }

    vector<string> lines;
    stringstream ss(content);
    string line;
    while (getline(ss, line)) {
        lines.push_back(line);
    }

    if (!content.empty() && content.back() != '\n') {
        lines.push_back("");
    }

    for (const auto& l : lines) {
        cout << "\033[" << row << ";1H" << l << endl;
        row++;
    }
}

int main() {
    const char* filename = "test.txt";
    int sleep_time = 7;
    int row = 1;
```

```

cout << "\033[2J\033[H";

int fd = open(filename, O_RDWR);
if (fd == -1) {
    string err_msg = "Error opening file: " + string(strerror(errno));
    printCentered(err_msg, "\033[31m", row);
    return 1;
}

struct flock fl;
fl.l_type = F_WRLCK;
fl.l_whence = SEEK_SET;
fl.l_start = 0;
fl.l_len = 0;

if (fcntl(fd, F_SETLK, &fl) == -1) {
    if (errno == EAGAIN || errno == EACCES) {
        printCentered("Cannot lock now, waiting...", "\033[33m", row);

        if (fcntl(fd, F_SETLKW, &fl) == -1) {
            string err_msg = "Error setting lock: " + string(strerror(errno));
            printCentered(err_msg, "\033[31m", row);
            close(fd);
            return 1;
        }
    }
    else {
        string err_msg = "Error attempting lock: " + string(strerror(errno));
        printCentered(err_msg, "\033[31m", row);
        close(fd);
        return 1;
    }
}
else {
    printCentered("Lock acquired successfully", "\033[32m", row);
}

row++;

lseek(fd, 0, SEEK_SET);
printCentered("File content:", "\033[34m", row);
printFileContent(fd, row);
    row++;

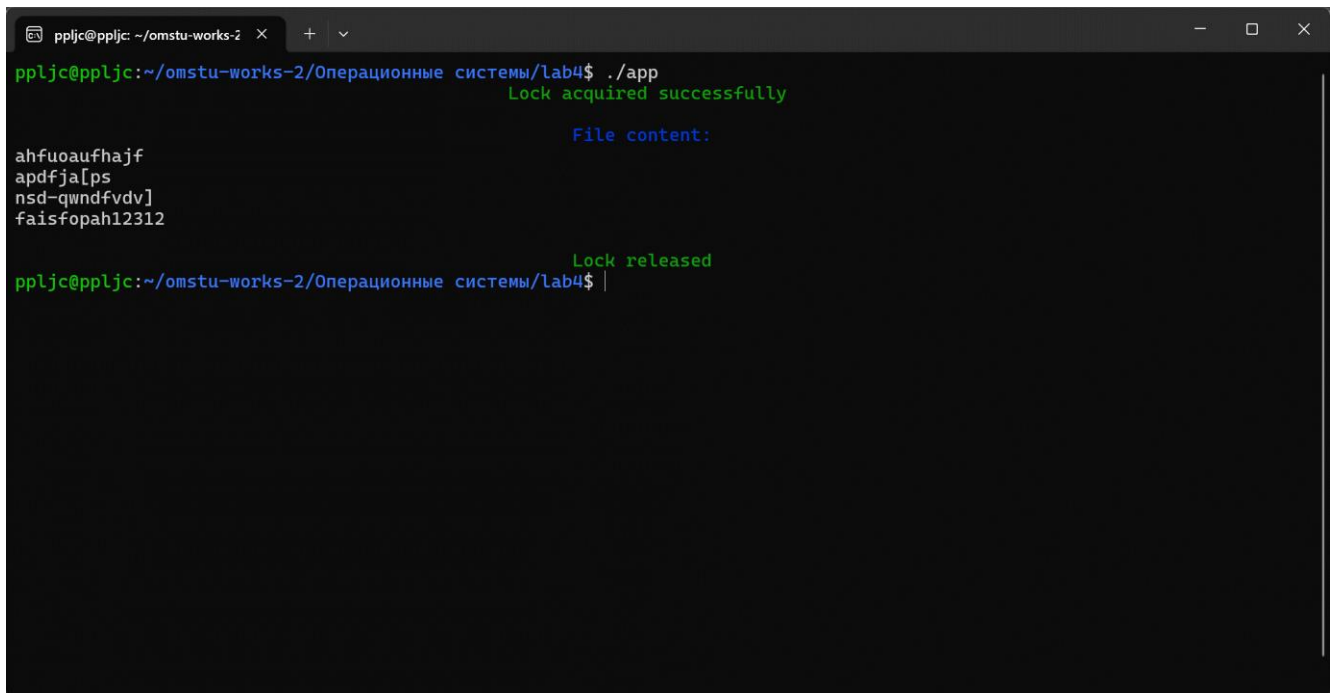
sleep(sleep_time);

fl.l_type = F_UNLCK;
if (fcntl(fd, F_SETLK, &fl) == -1) {
    string err_msg = "Error unlocking: " + string(strerror(errno));
    printCentered(err_msg, "\033[31m", row);
}
else {

```

```
    printCentered("Lock released", "\033[32m", row);  
}  
  
close(fd);  
return 0;  
}
```

## Результат выполнения:

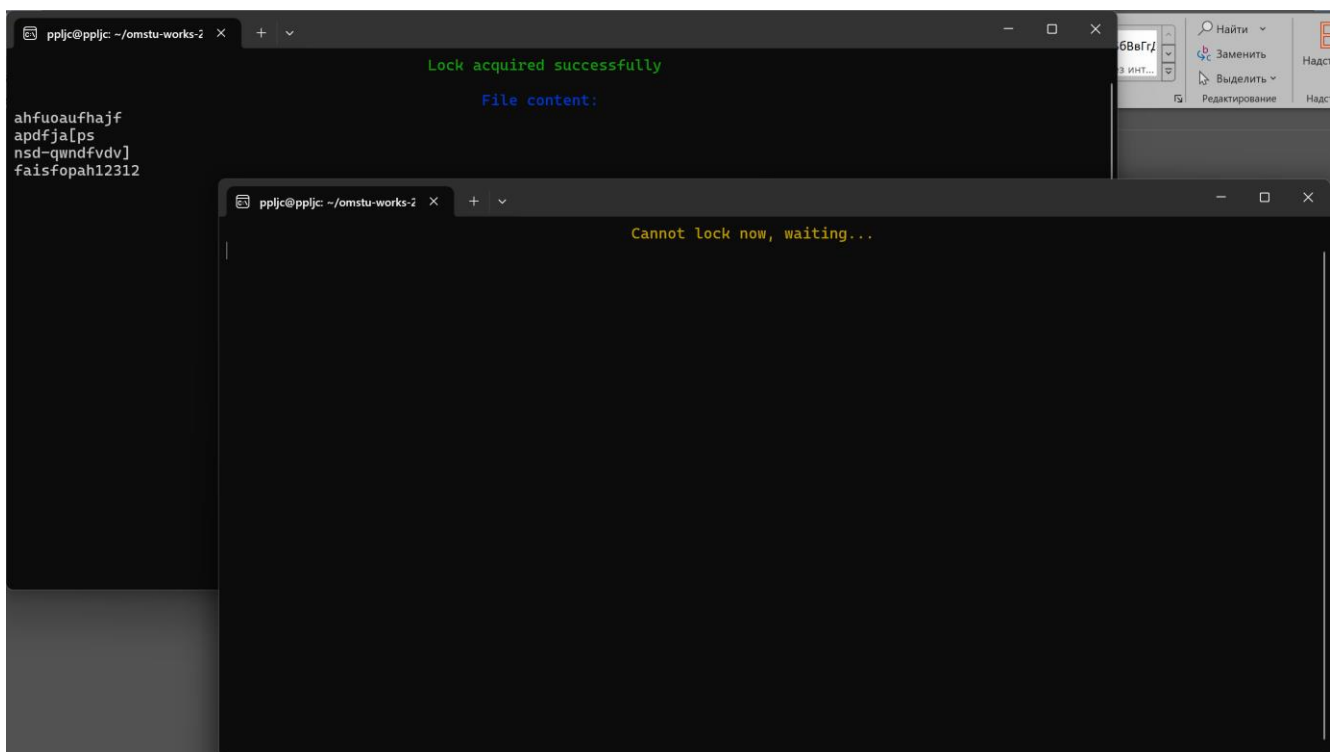


```
ppljc@ppljc: ~/omstu-works-2 x + v
ppljc@ppljc:~/omstu-works-2/Операционные системы/lab4$ ./app
Lock acquired successfully

File content:
ahfuoauafhjf
apdfja[ps
nsd-qwndfvdv]
faisfopah12312

Lock released
ppljc@ppljc:~/omstu-works-2/Операционные системы/lab4$ |
```

Рисунок 1 – Результат работы единственной копии программы.



```
ppljc@ppljc: ~/omstu-works-2 x + v
Lock acquired successfully

File content:
ahfuoauafhjf
apdfja[ps
nsd-qwndfvdv]
faisfopah12312

ppljc@ppljc: ~/omstu-works-2 x + v
Cannot lock now, waiting...
```

Рисунок 2 – Результат работы второй копии программы, запущенной параллельно с первой.

```
ppljc@ppljc: ~/omstu-works-2 x + v
Lock acquired successfully
File content:
ahfuoaufhajf
apdfja[ps
nsd-qwndfvdv]
faisfopah12312

ppljc@ppljc:~/omstu-works-2
ppljc@ppljc:~/omstu-works-2 x + v
Cannot lock now, waiting...
File content:
ahfuoaufhajf
apdfja[ps
nsd-qwndfvdv]
faisfopah12312
Lock released
ppljc@ppljc:~/omstu-works-2/Операционные системы/lab4$ |

else {
    string err;
    printCentered(err);
    close(fd);
    return 1;
}
else {
    printCentered(
```

Рисунок 3 – Результат работы второй копии программы, запущенной параллельно с первой, после завершения выполнения первой копии.

**Вывод:**

Было изучено применение механизмов многопользовательской блокировки файлов в операционной системе Linux. Освоены системные средства установки и снятия блокировок на файлы с использованием функций работы с файловыми дескрипторами. Исследовано поведение программы при попытке установить блокировку в условиях конкурентного доступа из нескольких процессов. Реализовано ожидание освобождения ресурса при невозможности немедленной блокировки. Получены навыки организации синхронного доступа к общим данным, обработки ошибок открытия и блокировки файлов, а также вывода диагностической информации о состоянии доступа.