МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Прикладная математика и фундаментальная информатика»

**Домашнее задание**

|  |  |  |
| --- | --- | --- |
| по дисциплине | Операционные системы |  |

|  |  |
| --- | --- |
| Студента | Зензина Егора Николаевича |
|  | фамилия, имя, отчество полностью |
| Курс | 2 Группа ФИТ-222 |
| Направление | 02.03.02 Фундаментальная информатика |
|  | и информационные технологии |
|  | код, наименование |
| Руководитель | старший преподаватель |
|  | должность, ученая степень, звание |
|  | Карабцов Р.Д. |
|  | фамилия, инициалы |
| Выполнил | 06.05.2024 |
|  | дата, подпись студента |
| баллы |  |
|  | дата, подпись руководителя |

Омск-2024

**Задание**

Разработать программу для Linux, которая должна запускаться в 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. (Базовый вариант)

1. Чем блокировка файла по записи в Windows отличается от блокировки по записи в Linux.

2. Чем блокировка файла по чтению в Windows отличается от блокировки по чтению в Linux.

3. Сформулируйте наиболее существенное отличие управлением цветом вывода на консоль в Windows и Linux, оцените в каких условиях какое из этих решений кажется вам предпочтительней.

**Решение**

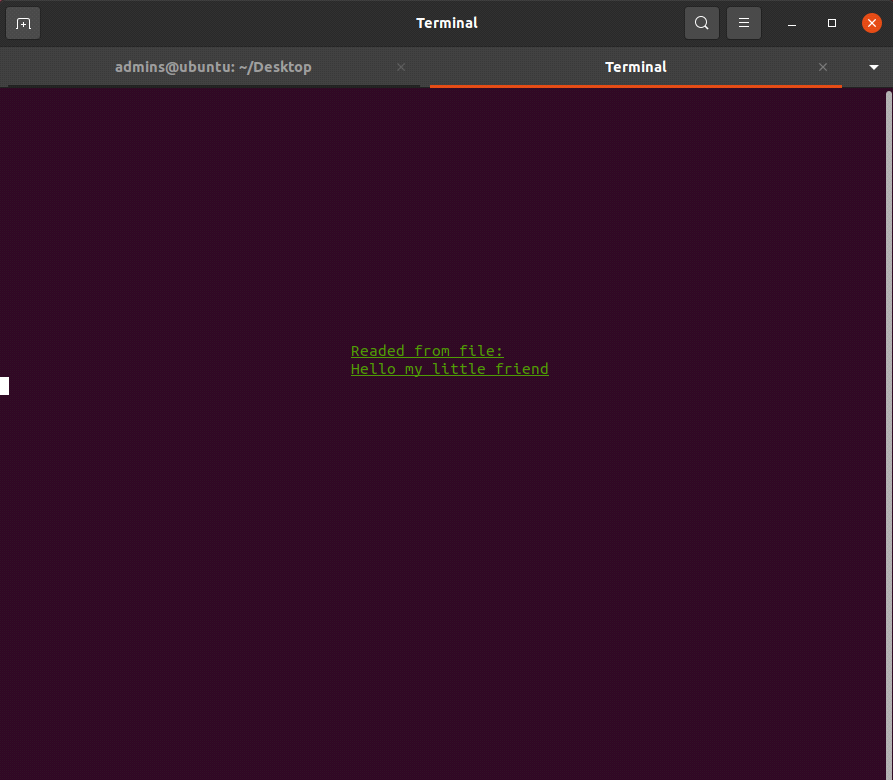


Рисунок 1. Старт программы

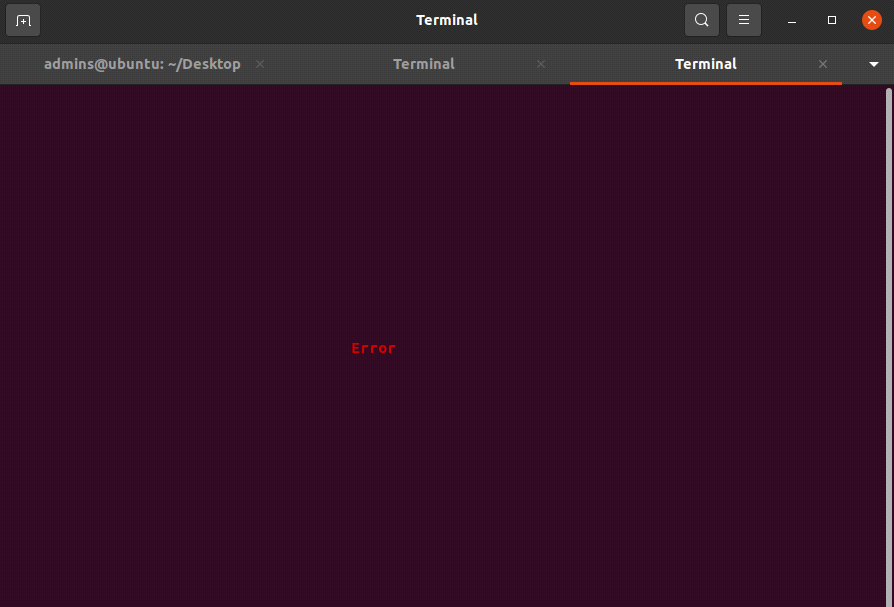


Рисунок 2. Блокировка и завершение программы

1. Блокировка файла для записи в Windows и Linux осуществляется разными методами. В Windows часто используется функция LockFile, которая позволяет заблокировать конкретный диапазон байтов внутри файла. В Linux для этой цели применяется системный вызов fcntl с аргументом F\_SETLK, что позволяет устанавливать блокировку на весь файл или его часть.

2. Блокировка файла для чтения в Windows и Linux осуществляется по-разному. В Windows для этой цели часто используется функция LockFileEx, которая позволяет установить блокировку только для чтения. В Linux аналогичная функциональность достигается с помощью системного вызова fcntl с аргументом F\_SETLK и указанием типа блокировки как F\_RDLCK.

3. Управление цветом вывода на консоль в Windows и Linux осуществляется по-разному. В Windows используются функции из Windows API, такие как SetConsoleTextAttribute, для изменения цвета текста. В Linux для изменения цвета текста используются ANSI Escape-коды, которые добавляются непосредственно к тексту.

**Листинг программы**

Файл *4\_laba.c:*

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#define RED "\033[1;31m" // красный цвет

#define GREEN "\033[4;32m" // зеленый цвет

int main() {

char name[] = "file"; // название файла для открытия

char buffer[255]; // буфер

int file; // хэндл файла

int lenght; // размер полученной строки чтения

struct flock lock={F\_WRLCK, SEEK\_SET, 0, 0};

file = open(name, O\_RDWR, 0600); // открываем файл

while (fcntl(file, F\_SETLK, &lock) == -1){ // проверяем на ошибку доступа

sprintf(buffer,"\033[15;40H%sError",RED);

write(1, buffer, strlen(buffer)); // выводим ошибку

sleep(1+(rand()%3)); // случайная задержка перед следующей проверкой файла

}

sprintf(buffer,"\033[15;40H%sReaded from file:\033[16;40H",GREEN); // указываем что прочитали файл, записавая это в буфер

write(1,buffer,strlen(buffer)); // выводим буфер в который записали что прочитан файл

lenght = read(file, buffer, 80); // чтение с файла

write(1, buffer, lenght); // вывод в консоль

sleep(7); // задержка в 7 секунд

close(file); // закрытие файла

return 0;

}