МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

**Лабораторная работа №3**

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

|  |  |
| --- | --- |
| Студента | Передериной Софьи Владимировны |
|  | фамилия, имя, отчество полностью |
| Курс | 2 Группа ФИТ-242 |
| Направление | 02.03.02 Фундаментальная информатика |
|  | и информационные технологии |
|  | код, наименование |
|  |  |

Омск-2025

**Задание**

Разработать программу для Windows, которая должна запускаться в двух экземплярах - каждый в своем окне командной оболочки FAR или из ПРОВОДНИКа операционной системы. Программа использует заранее подготовленный текстовый файл. Она пытается открыть этот файл для чтения с указываемым при этом запрете для других использовать этот файл. По результатам выполнения системной функции открытия на экран выдается сообщение - удалось ли открыть файл и, если не удалось по причине отсутствия доступа к одновременно выполняемым программам, то сообщение именно об этой причине. При отсутствии указанного в программе файла после сообщения об этом отсутствии программа прекращает работу. При его наличии, но невозможности продолжения действий из-за блокировки, установленной другим экземпляром запущенной программы, выполняется ожидание освобождения файла от блокировки. При отсутствии указанной причины доступа программа должна ждать освобождения файла. В обоих случаях - ожидания освобождения или исходной доступности - программа читает из этого файла все находящиеся в нем данные и выводит их на экран. Сообщения должны выводиться цветные и в середине консольного окна. (Базовый вариант.)

Разработать командный файл (пакетный файл в терминологии Windows для автоматического запуска трех экземпляров разработанной программы с интервалами запуска через 3 или 4 секунды между ними. (Дополнительное задание для максимального рейтинга).

**Базовый вариант задания**

#include <windows.h>

#include <stdio.h>

#include <string.h>

void print\_colored\_middle\_text(const char\* text, WORD color){

CONSOLE\_SCREEN\_BUFFER\_INFO console\_screen\_info;

HANDLE out\_handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

GetConsoleScreenBufferInfo(out\_handle, &console\_screen\_info);

int console\_width = console\_screen\_info.srWindow.Right - console\_screen\_info.srWindow.Left + 1;

int text\_len = (int)strlen(text);

int position\_of\_text = (console\_width - text\_len) / 2;

COORD position;

position.X = position\_of\_text;

position.Y = console\_screen\_info.dwCursorPosition.Y;

SetConsoleCursorPosition(out\_handle, position);

SetConsoleTextAttribute(out\_handle, color);

printf("%s\n", text);

SetConsoleTextAttribute(out\_handle, FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);

}

int main(){

HANDLE input\_file\_handle;

const char\* file\_name = "input.txt";

while (1){

input\_file\_handle = CreateFileA(

file\_name,

GENERIC\_READ,

0,

NULL,

OPEN\_EXISTING,

FILE\_ATTRIBUTE\_NORMAL,

NULL

);

if (input\_file\_handle == INVALID\_HANDLE\_VALUE){

DWORD error = GetLastError();

if (error == ERROR\_FILE\_NOT\_FOUND){

print\_colored\_middle\_text("File is not founded", FOREGROUND\_RED | FOREGROUND\_INTENSITY);

return 1;

}

else if (error == ERROR\_SHARING\_VIOLATION){

print\_colored\_middle\_text("The file is locked by another process", FOREGROUND\_BLUE);

Sleep(2000);

}

else{

print\_colored\_middle\_text("An error occurred while opening the file", FOREGROUND\_RED | FOREGROUND\_INTENSITY);

return 1;

}

}

else{

print\_colored\_middle\_text("File opened successfully", FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

break;

}

}

DWORD file\_size = GetFileSize(input\_file\_handle, NULL);

if (file\_size == INVALID\_FILE\_SIZE){

print\_colored\_middle\_text("Error reading file size", FOREGROUND\_RED | FOREGROUND\_INTENSITY);

CloseHandle(input\_file\_handle);

return 1;

}

char\* buffer = (char \*)malloc(file\_size + 1);

DWORD readed\_bytes;

if (ReadFile(input\_file\_handle, buffer, file\_size, &readed\_bytes, NULL)){

buffer[readed\_bytes] = '\0';

print\_colored\_middle\_text("Text in file: ", FOREGROUND\_GREEN | FOREGROUND\_RED | FOREGROUND\_BLUE);

print\_colored\_middle\_text(buffer, FOREGROUND\_GREEN | FOREGROUND\_RED | FOREGROUND\_BLUE);

}

else{

print\_colored\_middle\_text("Error reading file", FOREGROUND\_RED | FOREGROUND\_INTENSITY);

}

free(buffer);

CloseHandle(input\_file\_handle);

return 0;

}

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1. Запуск программы

Для работы с файлом использована функция CreateFileA, позволяющая установить флаги доступа : GENERIC\_READ – только для чтения и флаг 0 – для запрета совместного доступа. Они гарантируют, что файл нельзя открыть другим процессом, пока он используются некоторым другим процессом. Реализована обработка ошибки при отсутствии файла и при попытке открыть файл, пока тот занят другим процессом. В последнем случае при помощи функции Sleep ожидается завершение работы текущего процесса и последующего повтора открытия файла. Чтение содержимого файла реализовано с помощью функции GetFileSize для выделения буфера. Для вывода цветного текста по середине консоли была реализована функция print\_colored\_middle\_text. C помощью функции GetConsoleScreenBufferInfo была получена ширина окна. При использовании функций SetConsoleCursorPosition и SetConsoleTextAttribute курсор был установлен в ранее вычисленный центр консоли и цвет текста был установлен в соответствии тому, который был передан в качестве параметра функции print\_colored\_middle\_text.

**Дополнительное задание**

start cmd /k lab03.exe

timeout /t 3

start cmd /k lab03.exe

timeout /t 3

start cmd /k lab03.exe

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.Рисунок 2. Результат работы пакетного файла

Пакетный файл запускает новые процессы в отдельном окне, выполняя в консоли lab03.exe, по завершении работы оставляя окно открытым. Timeout позволяет приостановить выполнение на 3 секунды.