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 | 05.05.2024 |
|  | дата, подпись студента |
| баллы |  |
|  | дата, подпись руководителя |

Омск-2024

**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

**Контрольные вопросы:**

1. Чем стандартный ввод-вывод отличается от нестандартного, привести примеры использования того и другого.

2. Зачем в общем случае закрывать файл, к каким возможным последствиям и когда приводит не закрытие файла.

3. Какой код ошибки возвращает функция открытия файла, если в ней задано ограничение по одновременной работе с файлом и этот файл в момент выполнения попытки открытия уже захвачен для работы другим вычислительным процессом.

**Решение**

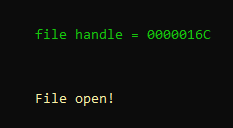
****

Рисунок 1. Вывод программы

1. Стандартный ввод-вывод (stdin/stdout) обычно используется для взаимодействия программы с пользователем через консоль. Нестандартный ввод-вывод, например, работа с файлами, предназначен для обработки данных из источников или записи данных в приемники, отличные от консоли. Например, функции printf() и scanf() используются для стандартного ввода-вывода, а функции fopen(), fread() и fwrite() - для нестандартного ввода-вывода, такого как чтение и запись данных в файлы.

2. Обычно закрытие файла позволяет освободить ресурсы, используемые программой для работы с ним, и сохранить все внесенные изменения. Если файл не закрыт, это может привести к утечке ресурсов или потере данных, если операционная система не сможет автоматически освободить ресурсы или сохранить изменения после завершения работы программы.

3. Если файл уже используется другим процессом и недоступен для открытия, функция открытия файла вернет код ошибки EBUSY (занятость).

**Листинг программы**

#include <Windows.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

HANDLE output, file\_handle;

void gotoxy(int x, int y) {

    COORD coordinates;

    coordinates.X = x;

    coordinates.Y = y;

    SetConsoleCursorPosition(output, coordinates);

}

int main() {

    output = GetStdHandle(STD\_OUTPUT\_HANDLE);

    DWORD length;

    int error = 0;

    char temp[255];

    RECT rect = {0};

    GetWindowRect(GetConsoleWindow(), &rect);

    int y = (rect.bottom - rect.top) / 50;

    int x = (rect.right - rect.left) / 20;

    do {

        system("cls");

        file\_handle = CreateFile("file.txt", GENERIC\_READ, FILE\_SHARE\_READ, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);

        gotoxy(x, y);

        sprintf(temp, "file handle = %p", file\_handle);

        SetConsoleTextAttribute(output, 10);

        WriteFile(output, temp, strlen(temp), NULL, NULL);

        if (file\_handle == INVALID\_HANDLE\_VALUE) {

            error = GetLastError();

            gotoxy(x, y + 2);

            sprintf(temp, "error = %d", error);

            SetConsoleTextAttribute(output, 4);

            WriteFile(output, temp, strlen(temp), NULL, NULL);

            if (error == 2) {

                gotoxy(x, y + 4);

                SetConsoleTextAttribute(output, 4);

                sprintf(temp, "File not found");

                WriteFile(output, temp, strlen(temp), NULL, NULL);

                Sleep(7000);

                return 0;

            }

            if (error == 32) {

                gotoxy(x, y + 4);

                SetConsoleTextAttribute(output, 4);

                sprintf(temp, "File is used");

                WriteFile(output, temp, strlen(temp), NULL, NULL);

            }

        } else {

            break;

        }

        Sleep(100);

    } while (1);

    gotoxy(x, y + 4);

    SetConsoleTextAttribute(output, 14);

    sprintf(temp, "File open!");

    WriteFile(output, temp, strlen(temp), NULL, NULL);

    ReadFile(file\_handle, temp, 100, &length, NULL);

    gotoxy(x, y + 6);

    SetConsoleTextAttribute(output, 15);

    WriteFile(output, temp, length, &length, NULL);

    Sleep(7000);

    CloseHandle(file\_handle);

    return 0;

}