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 | 08.05.2024 |
|  | дата, подпись студента |
| баллы |  |
|  | дата, подпись руководителя |

Омск-2024

**Задание**

Программа для Windows должна открыть текстовый файл для чтения и все его содержимое вывести в текстовую консоль (текстовое окно на экране). Затем в программе запускается опрос мыши в консольном режиме. По щелчке мышью на любом слове, находящемся в текстовой консоли, в последней строке этого текстового окна выводится буква, на которой осуществлен щелчок, и координаты позиции, на которой он произошел, выраженные в виде номера строки и столбца текстового режима. Эта информация выводится только для не пустого позиции (для пробелов выводить ничего не нужно). Такие действия программ может осуществлять многократно. Завершение программы осуществляется по нажатию правой клавиши мыши.

1. Чем событийное программирование отличается от жестко алгоритмического, опишите сферу применений того и другого?

2. Чем отличается программная обработка нажатий на управляющую клавишу от аналогичной обработки нажатий на алфавитно-цифровую клавишу при использовании функции ReadConsoleInput в Windows?

3. Почему в современных ОС для получения информации от мыши в прикладных программах используется событийно организованное программирование, а для получения информации от клавиатуры - не всегда?

**Решение**

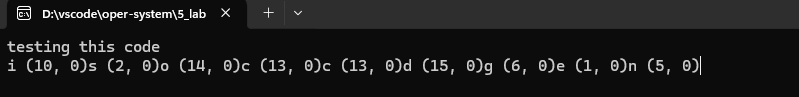
****

Рисунок 1. Вывод программы

1. Событийное программирование заключается в обработке событий, таких как нажатие клавиш или движение мыши, и динамическом реагировании на них. Этот подход широко применяется в графических интерфейсах пользователя (GUI), играх и веб-приложениях для обработки взаимодействий пользователя. В отличие от него, жестко алгоритмическое программирование сосредоточено на выполнении заданной последовательности операций для достижения конкретной цели, например, в математических расчетах или алгоритмах сортировки данных.

2. При программной обработке нажатий управляющих клавиш (таких как Enter или Esc), событие генерируется только при фактическом нажатии клавиши. Однако, при работе с алфавитно-цифровыми клавишами (например, 'A' или '1'), событие также создается при удержании клавиши, что может привести к множественным повторяющимся событиям нажатия.

3. Событийное программирование применяется для получения информации от мыши в современных операционных системах, так как оно позволяет оперативно реагировать на различные действия пользователя, такие как клики и движения курсора. Этот подход более гибок и эффективен для обработки взаимодействий с графическим интерфейсом. В отличие от этого, для получения информации с клавиатуры событийное программирование используется не всегда, поскольку зачастую достаточно просто получать текстовый ввод от пользователя по запросу или после завершения ввода.

**Листинг программы**

#include <windows.h>

#include <stdio.h>

void main() {

    char\* buffer;

    char symbol;

    DWORD bytesRead, bufferSize;

    HANDLE fileHandle, stdinHandle, stdoutHandle;

    INPUT\_RECORD inputBuffer;

    COORD symbolCoord;

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

    if (fileHandle == INVALID\_HANDLE\_VALUE) {

        return;

    }

    bufferSize = GetFileSize(fileHandle, NULL);

    buffer = (char\*)malloc(bufferSize \* sizeof(char));

    stdinHandle = GetStdHandle(STD\_INPUT\_HANDLE);

    if (stdinHandle == INVALID\_HANDLE\_VALUE) {

        CloseHandle(fileHandle);

        return;

    }

    stdoutHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);

    if (stdoutHandle == INVALID\_HANDLE\_VALUE) {

        CloseHandle(fileHandle);

        return;

    }

    ReadFile(fileHandle, buffer, bufferSize, &bytesRead, NULL);

    WriteFile(stdoutHandle, buffer, bytesRead, NULL, NULL);

    SetConsoleMode(stdinHandle, ENABLE\_EXTENDED\_FLAGS | ENABLE\_MOUSE\_INPUT);

    printf("\n");

    while (1) {

        if (ReadConsoleInput(stdinHandle, &inputBuffer, 1, &bytesRead)) {

            if (inputBuffer.EventType == MOUSE\_EVENT) {

                if (inputBuffer.Event.MouseEvent.dwButtonState == RIGHTMOST\_BUTTON\_PRESSED) {

                    break;

                }

                if (inputBuffer.Event.MouseEvent.dwButtonState == FROM\_LEFT\_1ST\_BUTTON\_PRESSED) {

                    symbolCoord.X = inputBuffer.Event.MouseEvent.dwMousePosition.X;

                    symbolCoord.Y = inputBuffer.Event.MouseEvent.dwMousePosition.Y;

                    ReadConsoleOutputCharacter(stdoutHandle, &symbol, 1, symbolCoord, &bytesRead);

                    if (symbol != ' ') {

                        printf("%c (%d, %d)", symbol, symbolCoord.X, symbolCoord.Y);

                    }

                }

            }

        }

    }

    free(buffer);

    CloseHandle(fileHandle);

}