МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

**Лабораторная работа №1**

|  |  |
| --- | --- |
| 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 в Windows программу, которая получает хэндлы стандартного ввода и вывода, выводит числовые с комментариями значения этих хэндлов, затем, используя стандартный ввод системными функциями небуферизованного ввода-вывода ReadFile или WrirwFile, делает приглашение для ввода, вводит любой текст и выводит его с предуведомлением, что он предварительно введен в программу. Предуведомление должно представлять вывод поясняющего текста с помощью отдельного оператора на основе вызова функции системного вывода. Продемонстрировать работу программы, запуская ее как с использованием стандартного ввода вывода по умолчанию, так и с переназначением этого ввода на файл для ввода исходных данных и вывода данных вместо экрана. (Базовый вариант задания).

Разработать аналогичную по действиям программу в Linux, обеспечивая в этой системе вывод приглашения на ввод данных со стандартного ввода только в случае использовании ввода с консоли, при переадресации этого ввода на входной файл приглашение отображаться не должно. При переадресации стандартного вывода в файл отображение приглашения в случае ввода с консоли должно принудительно появляться на экране, а не в файле, на который переадресуется вывод. Числовые значения хэндлов стандартных ввода и вывода в этом варианте выводить не нужно. (Дополнительное задание для максимального рейтинга).

**Базовый вариант задания**

#include <windows.h>

#include <stdio.h>

*int* *main*(){

    DWORD bytes\_read, bytes\_written;

    CHAR buffer[256];

    CHAR handles\_info[64];

    HANDLE handle\_input = *GetStdHandle*(*STD\_INPUT\_HANDLE*);

    HANDLE handle\_output = *GetStdHandle*(*STD\_OUTPUT\_HANDLE*);

    if (handle\_input *==* *INVALID\_HANDLE\_VALUE* ||

        handle\_output *==* *INVALID\_HANDLE\_VALUE*) return 1;

*wsprintf*(handles\_info, "STDIN handle = %p\nSTDOUT handle = %p\n", handle\_input, handle\_output);

*WriteFile*(handle\_output, handles\_info, *lstrlen*(handles\_info), &bytes\_written, *NULL*);

    const *char*\* promt\_to\_input = "Input text\n";

*WriteFile*(handle\_output, promt\_to\_input, *lstrlen*(promt\_to\_input), &bytes\_written, *NULL*);

    BOOL rc = *ReadFile*(handle\_input, buffer, sizeof(buffer), &bytes\_read, *NULL*);

    if (!rc){

        const *char*\* failed\_message = "Failed input";

*WriteFile*(handle\_output, failed\_message, *lstrlen*(failed\_message), &bytes\_written, *NULL*);

        return 1;

    }

    const *char*\* output\_message = "Your inputed text:";

*WriteFile*(handle\_output, output\_message, *lstrlen*(output\_message), &bytes\_written, *NULL*);

*WriteFile*(handle\_output, buffer, bytes\_read, &bytes\_written, *NULL*);

    return 0;

}

Запуск программы стандартного ввода вывода по умолчанию представлен на рисунке 1, с переназначением этого ввода на файл для ввода исходных данных и вывода данных вместо экрана представлен на рисунках 2-3.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1. Запуск программы из консоли Windows

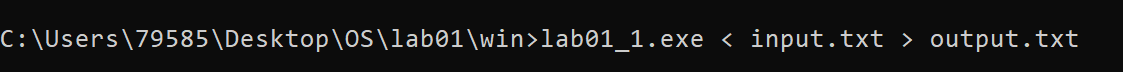
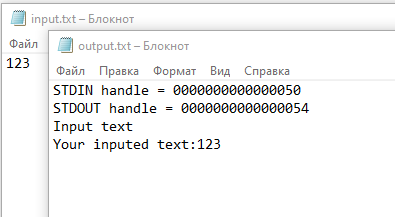


Рисунок 2

Рисунок 3

GetStdHandle(STD\_INPUT\_HANDLE) и GetStdHandle(STD\_OUT\_HANDLE) возвращают хэндлы на объекты ввода/вывода текущего процесса. При ошибке функция возвращает INVALID\_HANDLE\_VALUE, поэтому осуществляется проверка для предупреждения работы с недействительным значением. Использованы не буферизированные системные функции ввода/вывода WriteFile и ReadFile, работающие с Windows и не использующие буферизацию стандартной библиотеки C. ReadFile читает данные из источника, ассоциированного с handle\_input (консоль или файл) и возвращает количество прочитанных байт в bytes\_read.

**Дополнительное задание**

#include <unistd.h>

#include <string.h>

int main(){

char buffer[256];

ssize\_t bytes\_read;

int stdin\_is\_tty = isatty(STDIN\_FILENO);

int stdout\_is\_tty = isatty(STDOUT\_FILENO);

if (stdin\_is\_tty){

const char\* prompt\_to\_input\_message = "Input your message: ";

if (stdout\_is\_tty) {

write(STDOUT\_FILENO, prompt\_to\_input\_message,

strlen(prompt\_to\_input\_message));

} else {

write(STDERR\_FILENO, prompt\_to\_input\_message,

strlen(prompt\_to\_input\_message));

}

}

bytes\_read = read(STDIN\_FILENO, buffer, sizeof(buffer)-1);

if (bytes\_read < 0) {

const char \*failed\_message = "Failed to read input\n";

write(STDERR\_FILENO, failed\_message, strlen(failed\_message));

return 1;

}

buffer[bytes\_read] = '\n';

const char\* output\_message = "Your input: ";

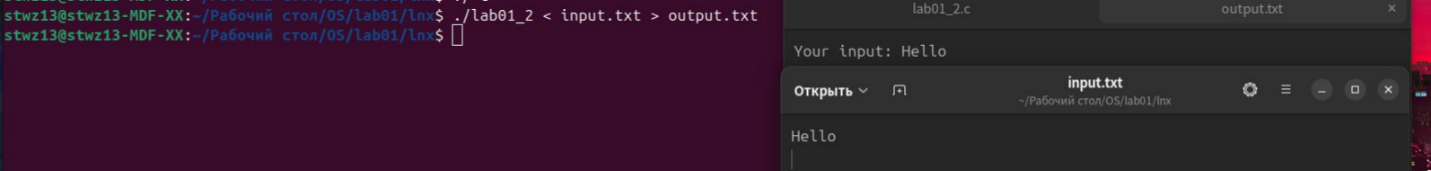
write(STDOUT\_FILENO, output\_message, strlen(output\_message));

write(STDOUT\_FILENO, buffer, bytes\_read);

return 0;

}

Isatty возвращает ненулевое значение, если переданный хэндл связан с терминалом. Для проверки перенаправления ввода из файла проверяем isatty(STDIN\_FILENO). Если stdout – терминал, то вывод приглашения происходит в терминал, но если stdout перенаправлен в файл, а ввод с происходил с терминала, то приглашение отображается на экране. Для этого вывод приглашения происходит в stderr, который по умолчанию остаётся связан с терминалом. Системные вызовы read и write дают не буферизованный доступ к файловым дескрипторам.

Рисунок 4. Перенаправленный ввод-вывод на Linux