МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«Чувашский государственный университет И.Н. Ульянова»
Факультет информатики и вычислительной техники
Кафедра вычислительной техники

Системное программирование
Лабораторная работа 4
«Копирование файлов на стандартное устройство вывода»

Выполнил:

Студент группы ИВТ-41-20
Галкин Д.С.

Проверил:

Яковлев С.В.

Чебоксары, 2024

Цель работы:

Создать пустое консольное приложение и добавить к нему служебные функции:

1. Напишите программу **cat**, которая копирует один или более указанный файл (либо стандартный ввод, если не указан файл) на стандартный вывод:

```
cat [-s] [файлы]

где -s подавляет сообщение об ошибке, если один из файлов не существует
```

2. Для разбора командной строки и определения позиции первого файла добавьте в проект файл со служебной функцией **Options,** которая выделяет из командной строки слова с префиксом "-", проверяет отдельные символы, устанавливает логические параметры и возвращает номер параметра с именем файла:

```
DWORD Options(int argc, LPCTSTR argv[], LPCTSTR optStr, ...)

где argv – командная строка. Опции, начинающиеся с "–", передаются в argv[1], argv[2]
```

3. **OptStr** - текстовая строка, содержащая все возможные опции во взаимно-однозначном соответствии с адресом булевых переменных в списке параметров-переменных (...). Эти флаги устанавливаются тогда и только тогда, когда символ соответствующей опции встречается в argv[1], argv[2], ...
4. Программа **cat** должна иметь локальную функцию

```
VOID **CatFile** (HANDLE hInFile, HANDLE hOutFile)
```

5. Проверьте работоспособность программы **cat** для всех вариантов использования.
6. Измените функцию **CatFile** так, чтобы она использовала **WriteConsole**, а не **WriteFile**, когда дескриптор стандартного вывода связан с консолью.

Полный текст программы:

write_utils.h

```
//
// Created by Dmitry Galkin on 15.02.2024.
//

#ifndef WRITE_UTILS_H
#define WRITE_UTILS_H

#include <string>

const std::string PATH_FILES_lAB4 = "../Lab4/resources/";

// Функция для разбора командной строки
int Options(int argc, char* argv[], const std::string& optStr, bool& sFlag);

void CatFile(int inFile, int outFile);

#endif //WRITE_UTILS_H
```

Lab4.cpp

```cpp
//
// Created by Dmitry Galkin on 15.02.2024.
//

#include "write_utils.h"

#include <iostream>
#include <string>
#include <unistd.h>

// Функция для разбора командной строки
int Options(int argc, char* argv[], const std::string& optStr, bool& sFlag) {
    sFlag = false; // Инициализация флага -s как false
    int fileIndex = 1; // Индекс первого файла в argv

    for (int i = 1; i < argc; ++i) {
        std::string arg = argv[i];
        if (arg[0] == '-') { // Проверка на наличие опции
            for (size_t j = 1; j < arg.length(); ++j) {
                if (arg[j] == 's') {
                    sFlag = true; // Установка флага -s
                } else {
                    std::cerr << "Неизвестная опция: " << arg[j] << std::endl;
                }
            }
        } else {
            fileIndex = i; // Индекс первого неопционного аргумента
            break;
        }
    }

    return fileIndex; // Возвращение индекса первого файла
}


void CatFile(int inFile, int outFile) {
    const size_t BUFFER_SIZE = 1024;
    char buffer[BUFFER_SIZE];
    ssize_t bytesRead;

    while ((bytesRead = read(inFile, buffer, BUFFER_SIZE)) > 0) {
        if (write(outFile, buffer, bytesRead) != bytesRead) {
            std::cerr << "Ошибка записи" << std::endl;
            return;
        }
    }

    if (bytesRead < 0) {
        std::cerr << "Ошибка чтения" << std::endl;
    }
    else {
        write(outFile, "\n", 1);
```

```
    }
}
```

main.cpp

```cpp
#include "Lab4/write_utils.h"
#include <iostream>
#include <vector>
#include <sys/termios.h>

using namespace std;

void start_lab4(int argc, char* argv[]);

int main(int argc, char* argv[]) {

    start_lab4(argc, argv);
    return 0;
}


void start_lab4(int argc, char* argv[]) {
    bool sFlag;
    int fileIndex = Options(argc, argv, "s", sFlag);

    if (argc == fileIndex) { // Если файлы не указаны, читать стандартный ввод
        CatFile(STDIN_FILENO, STDOUT_FILENO);
    } else {
        for (int i = fileIndex; i < argc; ++i) {
            std::string full_path = PATH_FILES_lAB4 + argv[i];
            int inFile = open(full_path.c_str(), O_RDONLY);
            if (inFile < 0) {
                if (!sFlag) {
                    std::cerr << "Невозможно открыть файл: " << argv[i] << std::endl;
                }
                continue;
            }

            CatFile(inFile, STDOUT_FILENO);
            close(inFile);
        }
    }
}
```

Пример работы:

```
/Users/anpulein/Documents/Projects/Chuvsu/Chuvsu_4Kurs_SystemProgramming/SystemProgramming/cmake-build-debug/SystemProgramming -s file1.txt file2.txt
data -> file1.txt
data -> file2.txt
```