Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

**Лабораторная работа №1.**

**КОПИРОВАНИЕ ФАЙЛОВ В Win32.**

По дисциплине «Системное программирование»

Выполнил: ст. гр. ИВТ-42-21

Матвеев Л.Е.

Проверил: Яковлев С. В.

г. Чебоксары, 2022

**Задание.**

1. Ознакомиться со средой разработки приложений Visual C++. Создать консольное приложение “Hello, Word!”. Выполнить его.

2. Написать, ввести, построить и выполнить три программы копирования файлов **cpC, cpW, cpCF** с указанной строкой запуска:

**cpC**: копирование файлов с использованием стандартной библиотеки С, которая поддерживает объекты потокового ввода-вывода типа FILE, подобные объектам Win32 типа HANDLE:

// cрC файл1 файл2: копирует файл1 в файл2 - программа 1.1

**cpW**: Копирование файлов с использованием функций Win32 CreateFile, ReadFile, WriteFile:

// cрW файл1 файл2: копирует файл1 в файл2 - программа 1.2

**cpCF**: Копирование файлов с использованием функции-полуфабриката CopyFile:

// cpCF файл1 файл2: копирует файл1 в файл2 - программа 1.3

3. Выяснить, как поведет себя программа **cpC**, если входной файл открыть в двоичном режиме, а выходной - в текстовом, и наоборот. Каков будет результат в MS DOS?

4. Измерить быстродействие программ копирования на файлах размером 10К, 100К, 1М, 10М, 100М, возможно, с применением функции DWORD GetTickCount(VOID) (заголовочный файл <windows.h.>). Результат оформить в виде таблицы.

**Ход работы.**

**Алгоритм программы cpC.**

1. **Открытие файлов**:
   * Исходный файл открывается в **двоичном режиме** ("rb")
   * Целевой файл открывается в **текстовом режиме** ("wt")
2. **Чтение и запись данных**:
   * Создать буфер фиксированного размера (например, 4 КБ).
   * В цикле считывать данные из исходного файла в буфер с помощью fread().
   * Записывать содержимое буфера в целевой файл через fwrite().
   * Цикл продолжается, пока fread() возвращает ненулевое количество байт.
3. **Закрытие файлов**:
   * Закрыть исходный и целевой файлы с помощью fclose().

**Алгоритм программы cpW.**

1. **Открытие исходного файла**:
   * Использовать CreateFile() с параметрами:
     + GENERIC\_READ — доступ на чтение.
     + FILE\_SHARE\_READ — разрешить совместное чтение.
     + OPEN\_EXISTING — открыть существующий файл.
     + FILE\_FLAG\_SEQUENTIAL\_SCAN — оптимизация для последовательного чтения.
2. **Создание целевого файла**:
   * Использовать CreateFile() с параметрами:
     + GENERIC\_WRITE — доступ на запись.
     + OPEN\_EXISTING — открыть существующий файл.
     + FILE\_ATTRIBUTE\_NORMAL — стандартные атрибуты.
3. **Копирование данных**:
   * Создать буфер (например, 4 КБ).
   * В цикле:
     + Читать данные из исходного файла через ReadFile().
     + Записывать данные в целевой файл через WriteFile().
     + Проверять, что количество записанных байт (bytesWritten) совпадает с прочитанными (bytesRead).
4. **Закрытие файлов**:
   * Закрыть дескрипторы файлов с помощью CloseHandle().

**Алгоритм программы cpCF.**

**Вызов функции CopyFile**:

* + Использовать CopyFile() с параметрами:
    - lpszExistingFile — путь к исходному файлу.
    - lpszNewFile — путь к целевому файлу.
    - fFailIfExists = FALSE — разрешить перезапись целевого файла.

**Программа cpC.**

* Если исходный файл в двоичном режиме ("rb"), а выходной в текстовом ("wt"):

Данные читаются как есть, включая символы \r\n (CR+LF). Затем каждый символ \n (LF) автоматически заменяется на \r\n, а существующие \r сохраняются.

В итоге, если исходный файл содержит \r\n, при записи они превратятся в \r\r\n.

* Если исходный файл в текстовом режиме ("rt"), выходной в двоичном ("wb"):

Последовательности \r\n преобразуются в \n. Символы \n сохраняются как есть (без добавления \r).

В итоге выходной файл будет содержать \n вместо \r\n (UNIX-формат), что может быть некорректно для DOS-программ.

**Быстродействие программ.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | cpC | cpW | cpCF |
| 10 КБ | 0 | 0 | 15 |
| 100 КБ | 0 | 0 | 15 |
| 1 МБ | 0 | 0 | 15 |
| 10 МБ | 47 | 31 | 15 |
| 100 МБ | 500 | 300 | 47 |