**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Операционные системы»**

**Тема: «Управление файловой системой»**

Студентки гр. 3312 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шахов К.С.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тимофеев А.В.

Санкт-Петербург

2025

**Цель работы.**

исследовать управление файловой системой с

помощью Win32 API.

**Задание 1.1.**

Управление дисками, каталогами и файлами.

Указания к выполнению.

1. Создайте консольное приложение с меню (каждая выполняемая

функция и/или операция должна быть доступна по отдельному пункту меню),

которое выполняет:

∙ вывод списка дисков (функции Win32 API – GetLogicalDrives,

GetLogicalDriveStrings);

∙ для одного из выбранных дисков вывод информации о диске и размер

свободного пространства (функции Win32 API –

GetDriveType, GetVolumeInformation, GetDiskFreeSpace); ∙

создание и удаление заданных каталогов (функции Win32 API –

CreateDirectory, RemoveDirectory);

∙ создание файлов в новых каталогах (функция Win32 API –

CreateFile);

∙ копирование и перемещение файлов между каталогами с возможностью

выявления попытки работы с файлами, имеющими совпадающие

имена (функции Win32 API – CopyFile, MoveFile, MoveFileEx);

∙ анализ и изменение атрибутов файлов (функции Win32 API –

GetFileAttributes, SetFileAttributes, GetFileInformationByHandle,

GetFileTime, SetFileTime).

**Описание программы**

**Назначение**

Данный код предназначен для работы с файловой системой в Windows. Он позволяет:

* Отображать доступные диски
* Показывать информацию о дисках
* Создавать и удалять директории
* Создавать, копировать и перемещать файлы
* Просматривать и изменять атрибуты и временные метки файлов

**Основные функции**

1. **displayDrives()** - Получает список доступных логических дисков в системе, используя API-функцию GetLogicalDrives(). Эта функция возвращает битовую маску, где каждый бит представляет букву диска (например, 0x3 означает, что доступны диски A: и B:). Затем с помощью цикла выводятся найденные диски.
2. **displayDriveDetails()** - Определяет размер, свободное место и файловую систему выбранного диска, используя:

GetDiskFreeSpaceEx() для получения общего, свободного и доступного пространства.

GetVolumeInformation() для извлечения метаданных диска, таких как файловая система (NTFS, FAT32 и т. д.).

Выводит полученные данные в удобном формате.

1. **createDirectory()** - Создает новую папку по указанному пути с помощью CreateDirectory(). Перед этим проверяет существование директории с помощью GetFileAttributes(). Если директория уже существует, выводится сообщение об ошибке.
2. **removeDirectory()** - Удаляет указанную папку с помощью RemoveDirectory(). Перед этим проверяет, существует ли директория (GetFileAttributes()) и пуста ли она (FindFirstFile() + FindNextFile() для проверки наличия файлов внутри).
3. **createFile()** - Создает новый файл в указанной директории, используя CreateFile() с флагом CREATE\_NEW. Функция выполняет:

Проверку существования файла (GetFileAttributes()).

Создание файла (CreateFile() с GENERIC\_WRITE, FILE\_ATTRIBUTE\_NORMAL).

Закрытие дескриптора файла (CloseHandle()).

1. **copyFile()** - Копирует файл из одного места в другое с помощью CopyFile(). Перед копированием проверяет:

Существует ли исходный файл (GetFileAttributes()).

Не существует ли уже файл по целевому пути (если не используется параметр bFailIfExists = FALSE).

При успешном копировании выводит сообщение об успехе.

1. **moveFile()** - Перемещает файл в новое расположение, используя MoveFile() или MoveFileEx(). Перед этим выполняет:

Проверку существования файла (GetFileAttributes()).

Перемещение с возможностью замены существующего файла (MOVEFILE\_REPLACE\_EXISTING).

Обработку ошибок (например, если файл используется другим процессом).

1. displayFileAttributes() - Определяет текущие атрибуты файла (FILE\_ATTRIBUTE\_HIDDEN, FILE\_ATTRIBUTE\_READONLY и т. д.) с помощью GetFileAttributes() и декодирует их в удобочитаемый вид.
2. showFileTimestamps() - Получает временные метки файла с помощью GetFileTime(), включая:

Дату создания файла.

Дату последнего изменения.

Дату последнего доступа.

Преобразует полученные FILETIME в удобочитаемый формат (SystemTimeToFileTime(), FileTimeToLocalFileTime()).

1. modifyFileTimestamps() - Позволяет пользователю изменить временные метки файла, используя SetFileTime(). Функция:

Запрашивает новые значения у пользователя.

Преобразует их в FILETIME (SystemTimeToFileTime()).

Применяет изменения к файлу (SetFileTime()).

1. modifyFileAttributes() - Позволяет изменять атрибуты файла с помощью SetFileAttributes(), например:

Сделать файл скрытым (FILE\_ATTRIBUTE\_HIDDEN).

Установить атрибут "только для чтения" (FILE\_ATTRIBUTE\_READONLY).

Снять все атрибуты перед изменением (GetFileAttributes() → SetFileAttributes()).

**Основные элементы кода**

* Использование Windows API для работы с файловой системой (например, GetLogicalDrives, GetFileAttributes, SetFileAttributes, CreateFile, MoveFile и т. д.).
* Работа с широкими строками (std::wstring) для поддержки Unicode.
* Использование std::wcout для корректного вывода данных в консоль.
* Обработка ошибок при взаимодействии с файловой системой.

**Вывод**

Разработанный код предоставляет удобный инструмент для работы с файлами и директориями в Windows. Возможны дальнейшие улучшения, такие как:

* Добавление поддержки GUI.
* Улучшение обработки ошибок (например, подробные коды ошибок GetLastError()).
* Добавление возможности работы с сетевыми дисками (WNetOpenEnum, WNetEnumResource).
* Расширение функциональности, например, получение прав доступа (GetSecurityInfo()).

Код демонстрирует основы работы с файловой системой Windows с использованием C++ и Windows API.