

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И.УЛЬЯНОВА (ЛЕНИНА)**

Кафедра ВТ

**ОТЧЕТ
по лабораторной работе №1
по дисциплине «Операционные системы»
Тема: Управление файловой системой**

Студент гр. 8307

Репин С.А.

Преподаватель

Тимофеев А.В.

Санкт-Петербург
2020

СОДЕРЖАНИЕ

Цель работы	3
Введение	3
1. Управление дисками, каталогами и файлами.	4
1.1. Вывод списка дисков	4
1.2. Вывод информации о диске	4
1.3. Создание и удаление директорий	5
1.4. Создание файлов	7
1.5. Копирование и перемещение файлов с учетом перемещения на место существующих файлов	7
1.6. Анализ и изменение атрибутов файлов	9
2. Копирование файла с помощью операций перекрывающегося ввода-вывода	14
Вывод	17

ЦЕЛЬ РАБОТЫ

Исследовать управление файловой системой с помощью Win32 API.

ВВЕДЕНИЕ

При выполнении лабораторной работы на языке программирования C стандарта C11 было разработано консольное приложение, управление которым происходит через различные меню, содержащие подпункты и подменю, соответствующие пунктам заданий. Исходный код приложения доступен на GitHub ¹.

Файл	Описание
menu.c	Определение типов и функций для работы с меню
main.c	Точка входа в программу; объявления конкретных меню и переходов между ними
actions.c	Реализация функций непосредственно выполняющих требования заданий (другими словами, callback'и конечных пунктов меню)
error.c	Описание номеров ошибок, а также функции отображения сообщений об ошибках

Таблица 1 Описание файлов в проекте

Сборка проекта производится с помощью Powershell-скрипта *build.ps1* (следует создать папку *build* и запускать скрипт из нее). Также потребуется пакет Build Tools for Visual Studio 2019.

¹https://github.com/stnrepin/os_labs/tree/master/lab1


```
PS C:\Users\aaa\Projects\os_labs\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 2
Enter drive root (e.g. 'C:/'): Z:/
Drive type: remote (network)
Volume name: VBox SharedFolder
Volume serial number: 0000fe01
Max component length: 255
File system flags: 00000006
File system name: VBoxSharedFolderFS
Sectors per cluster: 4096
Bytes per sector: 1
Number of free clusters: 28643115
Total number of clusters: 44624152
Free space: 109.26GiB
Done
```

Рис. 1.2 Результат выполнения программы

1.3. Создание и удаление директорий

Используя функции *CreateDirectory* и *RemoveDirectory*, программа создает и удаляет директории соответственно.

Заметим, что в программе также добавлена возможность передачи аргумента командной строки, при получении которого программа меняет текущую директорию при запуске с помощью функции *SetCurrentDirectory*.

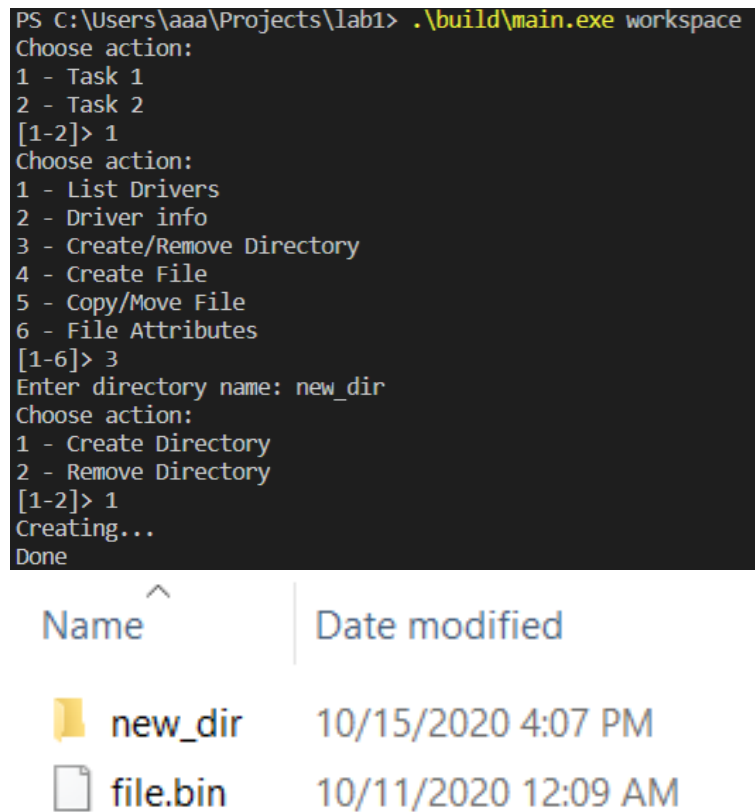


Рис. 1.3 Создание директории и проверка с помощью Проводника

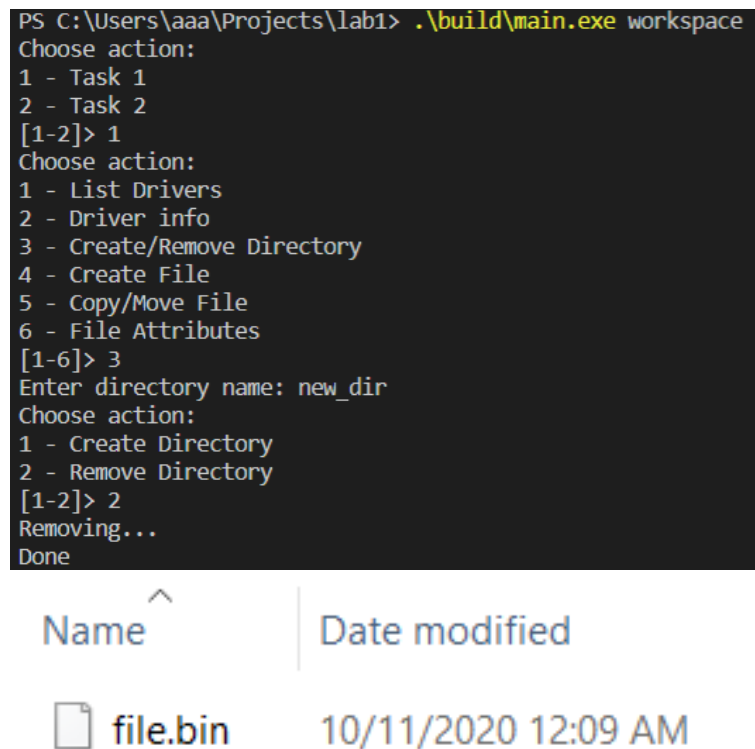


Рис. 1.4 Удаление директории и проверка с помощью Проводника

1.4. Создание файлов

Используя функцию *CreateFile* с модификатором *CREATE_NEW*, программа создает файлы.

```
PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 4
Enter file path: new_file.txt
Creating...
Done
```

Name	Date modified	Size
file.bin	10/11/2020 12:09 AM	30,720 KB
new_file.txt	10/15/2020 4:19 PM	0 KB

Рис. 1.5 Создание файла и проверка с помощью Проводника

1.5. Копирование и перемещение файлов с учетом перемещения на место существующих файлов

Используя функции *CopyFile*, *MoveFile* и *MoveFileEx*, программа копирует, перемещает и перемещает с заменой файлы и директории соответственно.

```

PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 5
Enter source file/directory path: file.bin
Enter destination file/directory path: file_copy1.bin
Choose action:
1 - Copy File or Directory
2 - Move File or Directory
3 - Move File or Directory (Replace Existing)
[1-3]> 1
Copying...
Done

```

Name	Date modified	Size
file.bin	10/11/2020 12:09 AM	30,720 KB
file_copy1.bin	10/11/2020 12:09 AM	30,720 KB
new_file.txt	10/15/2020 4:19 PM	0 KB

Рис. 1.6 Копирование файла и проверка с помощью Проводника

```

PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 5
Enter source file/directory path: file_copy1.bin
Enter destination file/directory path: file_copy2.bin
Choose action:
1 - Copy File or Directory
2 - Move File or Directory
3 - Move File or Directory (Replace Existing)
[1-3]> 2
Moving...
Done

```

Name	Date modified	Size
file.bin	10/11/2020 12:09 AM	30,720 KB
file_copy2.bin	10/11/2020 12:09 AM	30,720 KB
new_file.txt	10/15/2020 4:19 PM	0 KB

Рис. 1.7 Перемещение файла и проверка с помощью Проводника


```
PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 5
Enter source file/directory path: file_copy2.bin
Enter destination file/directory path: new_file.txt
Choose action:
1 - Copy File or Directory
2 - Move File or Directory
3 - Move File or Directory (Replace Existing)
[1-3]> 3
Moving...
Done
```



Name	Date modified	Size
 file.bin	10/11/2020 12:09 AM	30,720 KB
 new_file.txt	10/11/2020 12:09 AM	30,720 KB

Рис. 1.8 Перемещение файла на место существующего файла и проверка с помощью Проводника

1.6. Анализ и изменение атрибутов файлов

Используя функции *GetFileAttributes* и *SetFileAttributes*, программа получает и устанавливает атрибуты файлов (заметим, что набор атрибутов, доступных для получения, и набор атрибутов, доступный для установки, отличаются количественно в пользу первого). Получение информации о файле в ФС производится функцией *GetFileInformationByHandle*. Для получения и установки времени создания, последнего доступа и последнего изменения файла в программе используются функции *GetFileTime* и *SetFileTime* (обратите внимание, что эти функции не учитывают текущий часовой пояс, поэтому результат этих функций может отличаться от той информации, которую предоставляет система).

```

PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 6
Enter file name: new_file.txt
Choose action:
1 - Get File Attributes
2 - Set File Attributes
3 - Get File Information
4 - Get File Time
5 - Set File Time
[1-5]> 2
1 - Archive                2 - Hidden
3 - Normal                4 - Not content indexed
5 - Offline               6 - Readonly
7 - System                8 - Temporary
Enter file attribute index (0 to stop): 1
Set Archive attribute
Enter file attribute index (0 to stop): 4
Set Not content indexed attribute
Enter file attribute index (0 to stop): 8
Set Temporary attribute
Enter file attribute index (0 to stop): 10
Enter file attribute index (0 to stop): 0
Done

```

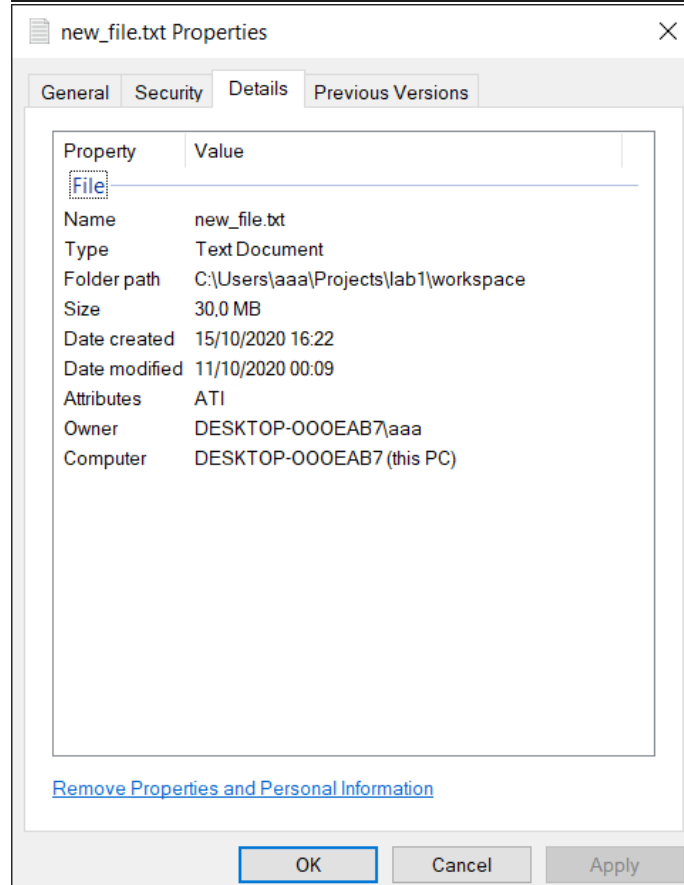


Рис. 1.9 Установка атрибутов файла и проверка с помощью Проводника

```

PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 6
Enter file name: new_file.txt
Choose action:
1 - Get File Attributes
2 - Set File Attributes
3 - Get File Information
4 - Get File Time
5 - Set File Time
[1-5]> 1
File attributes: Archive Not content indexed Temporary
Done

```

Рис. 1.10 Получение ранее установленных атрибутов из программы

```

PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 6
Enter file name: file.bin
Choose action:
1 - Get File Attributes
2 - Set File Attributes
3 - Get File Information
4 - Get File Time
5 - Set File Time
[1-5]> 3
File index: 1407374883556413
File size: 30MiB
Number of links: 1
Done

```

Рис. 1.11 Получение информации о файле

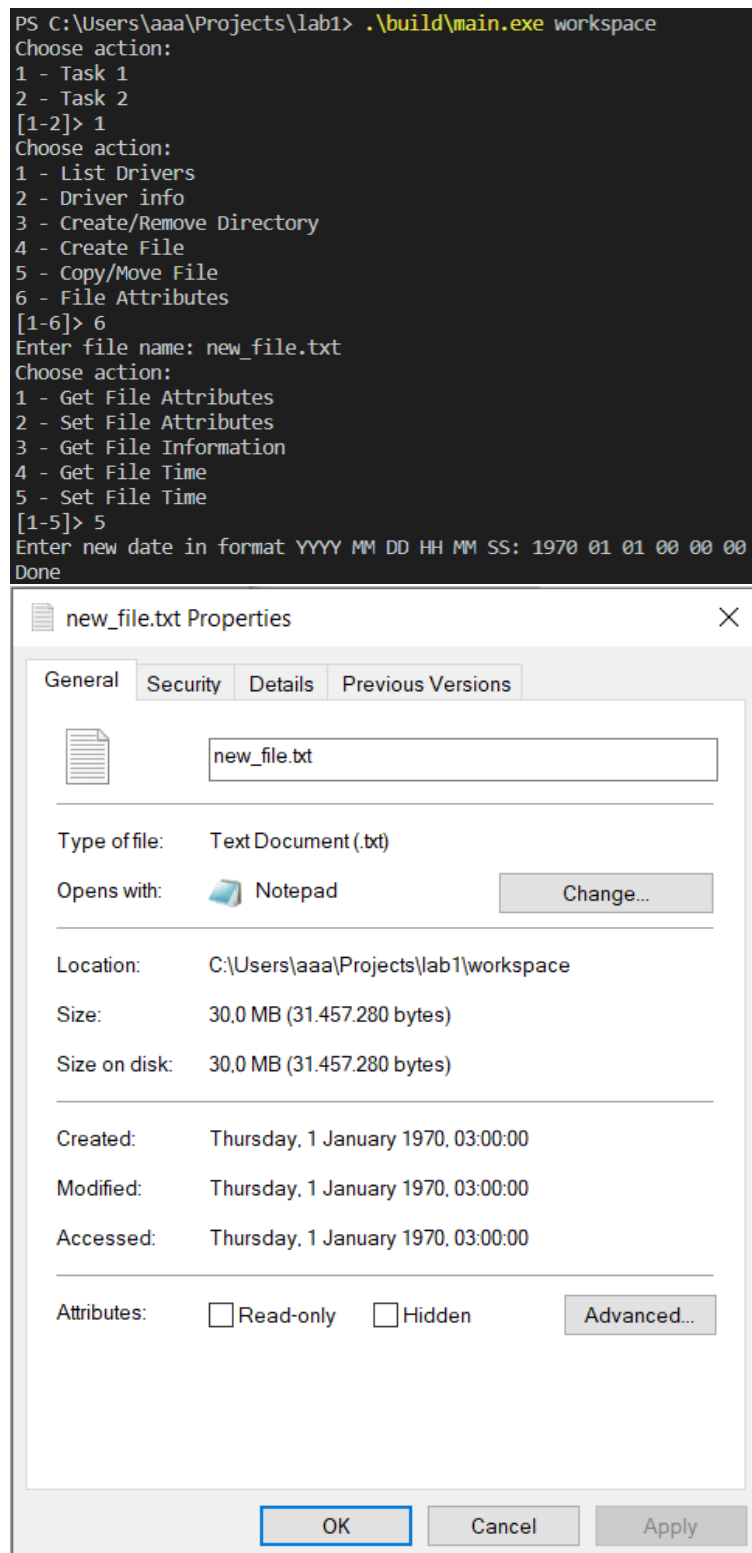


Рис. 1.12 Установка времени создания, времени доступа и времени изменения файла, и проверка с помощью Проводника

```
PS C:\Users\aaa\Projects\lab1> .\build\main.exe workspace
Choose action:
1 - Task 1
2 - Task 2
[1-2]> 1
Choose action:
1 - List Drivers
2 - Driver info
3 - Create/Remove Directory
4 - Create File
5 - Copy/Move File
6 - File Attributes
[1-6]> 6
Enter file name: new_file.txt
Choose action:
1 - Get File Attributes
2 - Set File Attributes
3 - Get File Information
4 - Get File Time
5 - Set File Time
[1-5]> 4
Creation time (UTC): 2020-10-15 13:22:01:954
Last access time (UTC): 2020-10-15 14:46:46:745
Last write time (UTC): 2020-10-10 21:09:42:178
Done
```

Рис. 1.13 Получение ранее установленных времени создания, времени доступа и времени изменения файла из программы

2. КОПИРОВАНИЕ ФАЙЛА С ПОМОЩЬЮ ОПЕРАЦИЙ ПЕРЕКРЫВАЮЩЕГОСЯ ВВОДА-ВЫВОДА

В этом задании с помощью функций *ReadFileEx*, *WriteFileEx*, *SleepEx* в программу была добавлена возможность копирования указанного файла операциями перекрывающегося ввода-вывода. Пользователю предоставляется возможность указывать количество одновременных операций ввода-вывода и множитель размера кластера, определяющий размер блоков, которыми копируется файл. Произведем эксперимент по замеру скорости копирования в зависимости от этих параметров.

Сперва, воспользуемся утилитой *dd* из пакета GNU Core Utilities и сгенерируем файл размером 30 MiB:

```
dd if=/dev/random of=file.bin bs=30MiB count=1
```

Зафиксируем количество количество операций ввода-вывода (возьмем 1) и замерим время выполнения копирования при различных размерах блока. Произведем три замера, найдем среднее арифметическое и построим график.

№	1	2	4	8	12	16	20
1	10266	5015	2609	2110	1640	1532	1110
2	10094	5594	2938	2156	1641	1547	813
3	9703	5250	3297	2094	1625	1532	1093
\bar{S}	10021	5286	2948	2120	1635	1537	1005

Таблица 2.1 Результаты изменения времени

После каждого опыта была проведена проверка корректности выполнения операции копирования с помощью утилиты *FS*.

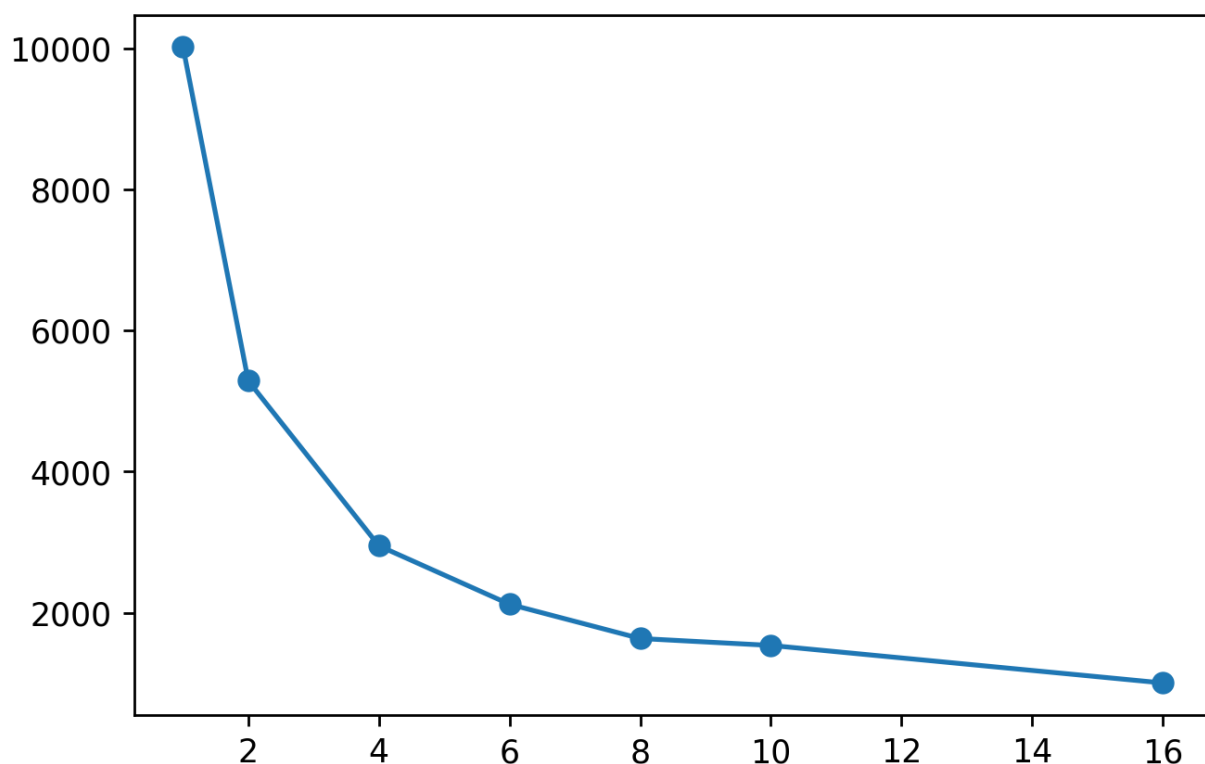


Рис. 2.1 График зависимости времени копирования от размера блока (при одной операции ввода-вывода)

На графике видно, что скорость прямо пропорциональна размеру блока. Это происходит из-за обратной зависимости между размером блока и количеством операций ввода-вывода, тем самым большой размер блока позволяет минимизировать число дорогих операций.

Зафиксируем размер блоков (например, возьмем 4 размера кластера) и замерим время выполнения копирования при различном количестве операций ввода-вывода. Аналогично предыдущему опыту произведем три замера, найдем среднее арифметическое и построим график.

№	1	2	4	8	12	16	20
1	4000	2375	1921	1391	1000	1343	1609
2	3313	2031	1734	1359	984	1219	1329
3	2890	2703	2031	1359	906	1390	1475
\bar{S}	3401	2369	1895	1369	963	1317	1471

Таблица 2.2 Результаты изменения времени

После каждого опыта была проведена проверка корректности выполнения операции копирования с помощью утилиты *FS*.

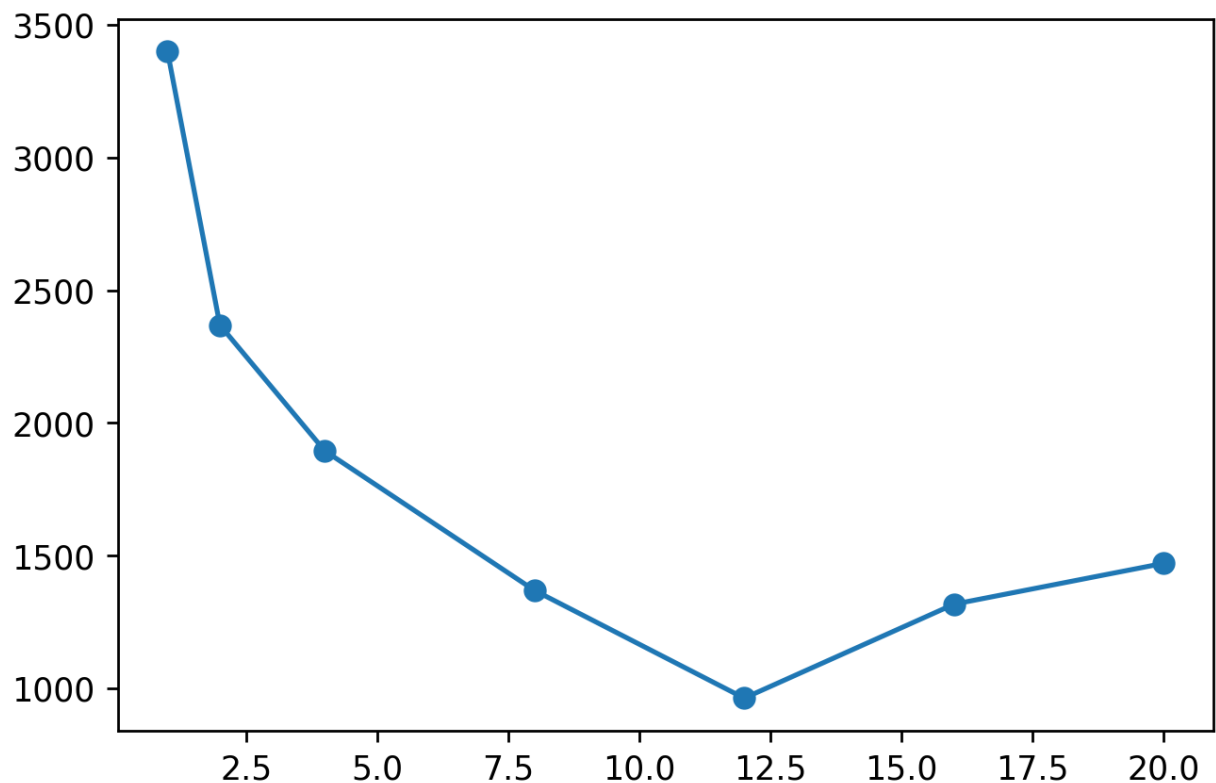


Рис. 2.2 График зависимости времени копирования от количества операций ввода-вывода (при размере блока равным 4 размерам кластера)

На графике видно, что максимальная скорость достигается при 12 операциях ввода-вывода. Увеличение времени при дальнейшем возрастании числа операций объясняется возрастанием числа системных вызовов, что приводит к частым переключениям контекста, которые занимают ощутимое время.

ВЫВОД

В ходе выполнения данной лабораторной работы были получены знания и практические навыки низкоуровневой работы с файловой системой при помощи Windows API. Было продемонстрировано, что такой подход дает программисту широкие возможности для работы с файлами и директориями, намного большие, чем обычно предоставляемые стандартной библиотекой языков, хотя с другой стороны, и приходится жертвовать переносимостью программы. Параллельно, был изучен подход Windows API в работу с Unicode, а также был получен опыт работы с мощной системой документации MSDN.

Отдельно стоит отметить, получение умений в работе с асинхронным выполнением операций чтения и записи файлов. Как было показано в работе, этот метод дает возможность многократного увеличения скорости чтения и записи файлов.