

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Гомельский государственный университет имени Франциска Скорины»
Факультет физики и информационных технологий
Кафедра общей физики

Отчёт по лабораторной работе №5

«Настройка сетевых подключений ОС»

Выполнил:

студент группы КИ-22: Аксёнов И. С.

Проверил: Грищенко.В.В

Гомель 2025

Цель работы: Изучить основные принципы работы с каталогами и файлами в операционной системе Windows.

Краткие сведения из теории

Файловая система (англ. file system) – порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах. Файловая система определяет формат содержимого и способ физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имен файлов (и каталогов), максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

Любая операционная система может работать с разными файловыми системами – например. Задача операционной системы заключается в предоставлении пользователю стандартного интерфейса, позволяющего ему обращаться к каждой файловой системе, не вникая в ее особенности.

Файловая система связывает носитель информации с одной стороны и API для доступа к файлам – с другой. Когда прикладная программа обращается к файлу, она не имеет никакого представления о том, каким образом расположена информация в конкретном файле, так же, как и о том, на каком физическом типе носителя (CD, жёстком диске, магнитной ленте, флеш-памяти или другом) он записан. Всё, что знает программа — это имя файла, его размер и атрибуты. Эти данные она получает от драйвера файловой системы. Именно файловая система устанавливает, где и как будет записан файл на физическом носителе (например, жёстком диске).

С точки зрения операционной системы (ОС), весь диск представляет собой набор кластеров (как правило, размером 512 байт и больше). Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

Основные функции любой файловой системы нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс работы с файлами для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;

- организация устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;

- содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).

В многопользовательских системах появляется ещё одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя, а также обеспечение совместной работы с файлами, к примеру, при открытии файла одним из пользователей, для других этот же файл временно будет доступен в режиме «только чтение».

Файл — это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные, а также собственно хранимые в этой области данные и набор атрибутов, позволяющие ОС манипулировать этими данными.

Файлы представляют собой абстрактные объекты. Их задача — хранить информацию, скрывая от пользователя детали работы с устройствами. Когда процесс создает файл, он дает ему имя. После завершения процесса файл продолжает существовать и через свое имя может быть доступен другим процессам.

Правила именования файлов зависят от ОС. Многие ОС поддерживают имена из двух частей (имя+расширение), например, prog.c (файл, содержащий текст программы на языке Си) или autoexec.bat (файл, содержащий команды интерпретатора командного языка). Тип расширения файла позволяет ОС организовать работу с ним различных прикладных программ в соответствии с заранее оговоренными соглашениями. Обычно ОС накладывают некоторые ограничения, как на используемые в имени символы, так и на длину имени файла. В соответствии со стандартом POSIX, популярные ОС оперируют удобными для пользователя длинными именами (до 255 символов).

Файлы предназначены для хранения информации с возможностью ее последующего извлечения. Разные системы предоставляют различные операции, позволяющие сохранять и извлекать информацию. Далее рассматриваются наиболее распространенные системные вызовы, относящиеся к работе с файлами.

- Create (Создать). Создает файл без данных. Цель вызова состоит в объявлении о появлении нового файла и установке ряда атрибутов.

- Delete (Удалить). Когда файл больше не нужен, его нужно удалить, чтобы освободить дисковое пространство. Именно для этого и предназначен этот системный вызов.

– Open (Открыть). Перед использованием файла процесс должен его открыть. Цель системного вызова open – дать возможность системе извлечь и поместить в оперативную память атрибуты и перечень адресов на диске, чтобы ускорить доступ к ним при последующих вызовах.

– Close (Закреть). После завершения всех обращений к файлу потребность в его атрибутах и адресах на диске уже отпадает, поэтому файл должен быть закрыт, чтобы освободить место во внутренней таблице. Многие системы устанавливают максимальное количество открытых процессами файлов, определяя смысл существования этого вызова. Информация на диск пишется блоками, и закрытие файла вынуждает к записи последнего блока файла, даже если этот блок и не заполнен.

– Read (Произвести чтение). Считывание данных из файла. Как правило, байты поступают с текущей позиции. Вызывающий процесс должен указать объем необходимых данных и предоставить буфер для их размещения.

– Write (Произвести запись). Запись данных в файл, как правило, с текущей позиции. Если эта позиция находится в конце файла, то его размер увеличивается. Если текущая позиция находится где-то в середине файла, то новые данные пишутся поверх существующих, которые утрачиваются навсегда.

– Append (Добавить). Этот вызов является усеченной формой системного вызова write. Он может лишь добавить данные в конец файла. Как правило, у систем, предоставляющих минимальный набор системных вызовов, вызов append отсутствует, но многие системы предоставляют множество способов получения того же результата, и иногда в этих системах присутствует вызов append.

– Seek (Найти). При работе с файлами произвольного доступа нужен способ указания места, с которого берутся данные. Одним из общепринятых подходов является применение системного вызова seek, который перемещает указатель файла к определенной позиции в файле. После завершения этого вызова данные могут считываться или записываться с этой позиции.

– Get attributes (Получить атрибуты). Процессу для работы зачастую необходимо считать атрибуты файла. К примеру, имеющаяся в UNIX программа make обычно используется для управления проектами разработки программного обеспечения, состоящими из множества сходных файлов. При вызове программа make проверяет время внесения последних изменений всех исходных и объектных файлов и для обновления проекта обходится компиляцией лишь минимально необходимого количества файлов. Для этого ей

необходимо просмотреть атрибуты файлов, а именно время внесения последних изменений.

- `Set attributes` (Установить атрибуты). Значения некоторых атрибутов могут устанавливаться пользователем и изменяться после того, как файл был создан. Такую возможность дает именно этот системный вызов. Характерным примером может послужить информация о режиме защиты. Под эту же категорию подпадает большинство флагов.

- `Rename` (Переименовать). Нередко пользователю требуется изменить имя существующего файла. Этот системный вызов помогает решить эту задачу. Необходимость в нем возникает не всегда, поскольку файл может быть просто скопирован в новый файл с новым именем, а старый файл затем может быть удален. Обычно в файловой системе для упорядочения файлов имеются каталоги или папки, которые сами по себе являются файлами. Допустимые системные вызовы для управления каталогами имеют большее количество вариантов от системы к системе, чем системные вызовы, управляющие файлами. Рассмотрим примеры, дающие представление об этих системных вызовах и характере их работы (взяты из системы UNIX).

- `Create` (Создать каталог). Каталог создается пустым, за исключением точки и двойной точки, которые система помещает в него автоматически (или в некоторых случаях при помощи программы `mkdir`).

- `Delete` (Удалить каталог). Удалить можно только пустой каталог. Каталог, содержащий только точку и двойную точку, рассматривается как пустой, поскольку они не могут быть удалены.

- `Opendir` (Открыть каталог). Каталоги могут быть прочитаны. К примеру, для вывода имен всех файлов, содержащихся в каталоге, программа `ls` открывает каталог для чтения имен всех содержащихся в нем файлов. Перед тем как каталог может быть прочитан, он должен быть открыт по аналогии с открытием и чтением файла.

- `Closedir` (Закрыть каталог). Когда каталог прочитан, он должен быть закрыт, чтобы освободить пространство во внутренних таблицах системы.

- `Readdir` (Прочитать каталог). Этот вызов возвращает следующую запись из открытого каталога. Раньше каталоги можно было читать с помощью обычного системного вызова `read`, но недостаток такого подхода заключался в том, что программист вынужден был работать с внутренней структурой каталогов, о которой он должен был знать заранее. В отличие от этого, `readdir` всегда возвращает одну запись в стандартном формате независимо от того, какая из возможных структур каталогов используется.

– Rename (Переименовать каталог). Во многих отношениях каталоги подобны файлам и могут быть переименованы точно так же, как и файлы.

– Link (Привязать). Привязка представляет собой технологию, позволяющую файлу появляться более чем в одном каталоге. В этом системном вызове указываются существующий файл и новое имя файла в некотором существующем каталоге и создается привязка существующего файла к указанному каталогу с указанным новым именем. Таким образом, один и тот же файл может появиться в нескольких каталогах, возможно, под разными именами. Подобная привязка, увеличивающая показания файлового счетчика iузла (предназначенного для отслеживания количества записей каталогов, в которых фигурирует файл), иногда называется жесткой связью, или жесткой ссылкой (hard link).

– Unlink (Отвязать). Удалить запись каталога. Если отвязываемый файл присутствует только в одном каталоге (что чаще всего и бывает), то этот вызов удалит его из файловой системы. Если он фигурирует в нескольких каталогах, то он будет удален из каталога, который указан в имени файла. Все остальные записи останутся. Фактически системным вызовом для удаления файлов в UNIX (как ранее уже было рассмотрено) является unlink.

В приведенном списке перечислены наиболее важные вызовы, но существуют и другие вызовы, к примеру для управления защитой информации, связанной с каталогами.

Еще одним вариантом идеи привязки файлов является символическая ссылка (symbolic link). Вместо двух имен, указывающих на одну и ту же внутреннюю структуру данных, представляющую файл, может быть создано имя, указывающее на очень маленький файл, в котором содержится имя другого файла. Когда используется первый файл, например, он открывается, файловая система идет по указанному пути и в итоге находит имя. Затем она начинает процесс поиска всех мест, где используется это новое имя. Преимуществом символических ссылок является то, что они могут пересекать границы дисков и даже указывать на имена файлов, находящихся на удаленных компьютерах. И тем не менее их реализация несколько уступает в эффективности жестким связям.

Допустимые системные вызовы для управления каталогами имеют большее количество вариантов от системы к системе, чем системные вызовы, управляющие файлами.

– Create (Создать каталог). Каталог создается пустым, за исключением точки и двойной точки, которые система помещает в него автоматически (или в некоторых случаях при помощи программы `mkdir`).

– Delete (Удалить каталог). Удалить можно только пустой каталог. Каталог, содержащий только точку и двойную точку, рассматривается как пустой, поскольку они не могут быть удалены.

– Opendir (Открыть каталог). Каталоги могут быть прочитаны. К примеру, для вывода имен всех файлов, содержащихся в каталоге, программа `ls` открывает каталог для чтения имен всех содержащихся в нем файлов. Перед тем как каталог может быть прочитан, он должен быть открыт по аналогии с открытием и чтением файла.

– Closedir (Заккрыть каталог). Когда каталог прочитан, он должен быть закрыт, чтобы освободить пространство во внутренних таблицах системы.

– Readdir (Прочитать каталог). Этот вызов возвращает следующую запись из открытого каталога. Раньше каталоги можно было читать с помощью обычного системного вызова `read`, но недостаток такого подхода заключался в том, что программист вынужден был работать с внутренней структурой каталогов, о которой он должен был знать заранее. В отличие от этого, `readdir` всегда возвращает одну запись в стандартном формате независимо от того, какая из возможных структур каталогов используется.

– Rename (Переименовать каталог). Во многих отношениях каталоги подобны файлам и могут быть переименованы точно так же, как и файлы.

– Link (Привязать). Привязка представляет собой технологию, позволяющую файлу появляться более чем в одном каталоге. В этом системном вызове указываются существующий файл и новое имя файла в некотором существующем каталоге и создается привязка существующего файла к указанному каталогу с указанным новым именем. Таким образом, один и тот же файл может появиться в нескольких каталогах, возможно, под разными именами. Подобная привязка, увеличивающая показания файлового счетчика `iuzla` (предназначенного для отслеживания количества записей каталогов, в которых фигурирует файл), иногда называется жесткой связью, или жесткой ссылкой (`hard link`).

– Unlink (Отвязать). Удалить запись каталога. Если отвязываемый файл присутствует только в одном каталоге (что чаще всего и бывает), то этот вызов удалит его из файловой системы. Если он фигурирует в нескольких каталогах, то он будет удален из каталога, который указан в имени файла.

Ход работы

```
C:\Users\Administrator> New-Item -Path "$HOME\dir1\dir2\dir3" -ItemType Directory -Force

Directory: C:\Users\Administrator\dir1\dir2

Mode                LastWriteTime         Length Name
----                -
d-----          22.05.2025   16:11             dir3
```

Рисунок 1 - Создание вложенных директорий

```
PS C:\Users\Administrator> New-Item -Path "$HOME\file1" -ItemType File

Directory: C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   16:14             0 file1

PS C:\Users\Administrator> New-Item -Path "$HOME\file2" -ItemType File

Directory: C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   16:14             0 file2
```

Рисунок 2 - Создание пустых файлов

```
PS C:\Users\Administrator> Copy-Item "$HOME\file1" "$HOME\dir1"
PS C:\Users\Administrator> Copy-Item "$HOME\file2" "$HOME\dir1"
PS C:\Users\Administrator> Copy-Item "$HOME\file2" "$HOME\dir2"
PS C:\Users\Administrator> Copy-Item "$HOME\file1" "$HOME\dir2"
PS C:\Users\Administrator> Copy-Item "$HOME\file1" "$HOME\dir3"
PS C:\Users\Administrator> Copy-Item "$HOME\file2" "$HOME\dir3"
```

Рисунок 3 - Копирование файлов в каталоги

```
PS C:\Users\Administrator> Remove-Item "$HOME\file1" -Force
PS C:\Users\Administrator> Remove-Item "$HOME\file2" -Force
```

Рисунок 4 - Удаление файлов file1 и file2


```

PS C:\Users\Administrator> New-Item -Path "$HOME\current" -ItemType Directory

Directory: C:\Users\Administrator


Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         16:20         current

PS C:\Users\Administrator> New-Item -Path "$HOME\new" -ItemType Directory

Directory: C:\Users\Administrator


Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         16:21         new

PS C:\Users\Administrator> New-Item -Path "$HOME\old" -ItemType Directory

Directory: C:\Users\Administrator


Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         16:21         old

```

Рисунок 5 - Создание трех директорий

```

PS C:\Users\Administrator> Copy-Item "C:\Users\Administrator\qqq" "$HOME\old" -Recurse -Force
PS C:\Users\Administrator> Copy-Item "C:\Users\Administrator\qqq" "$HOME\new" -Recurse -Force
PS C:\Users\Administrator> Copy-Item "C:\Users\Administrator\qqq" "$HOME\current" -Recurse -Force
PS C:\Users\Administrator> Get-ChildItem "$HOME\new" -Recurse | ForEach-Object { $_.LastWriteTime = (Get-Date).AddYears(-1) }

```

Рисунок 6 - Копирование файлов в old, current, new с изменением даты на год назад с сохранением атрибутов

```

PS C:\Users\Administrator> Compress-Archive -Path "$HOME\current", "$HOME\new", "$HOME\old" -DestinationPath "$HOME\directories.zip"

```

Рисунок 7 - Создание архивного файла из трех директорий

```

PS C:\Users\Administrator> New-Item -Path "$HOME\unsorted" -ItemType Directory

Directory: C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
d-----          22.05.2025   17:32             unsorted

PS C:\Users\Administrator> New-Item -Path "$HOME\unsorted/example.mp3" -ItemType File

Directory: C:\Users\Administrator\unsorted

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   17:33             0 example.mp3

PS C:\Users\Administrator> New-Item -Path "$HOME\unsorted/example.mp4" -ItemType File

Directory: C:\Users\Administrator\unsorted

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   17:33             0 example.mp4

PS C:\Users\Administrator> New-Item -Path "$HOME\unsorted/example.pdf" -ItemType File

Directory: C:\Users\Administrator\unsorted

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   17:33             0 example.pdf

PS C:\Users\Administrator> New-Item -Path "$HOME\unsorted/random_file.txt" -ItemType File

Directory: C:\Users\Administrator\unsorted

Mode                LastWriteTime         Length Name
----                -
-a-----          22.05.2025   17:33             0 random_file.txt

```

Рисунок 8 - Создание каталога unsorted и тестовых файлов вручную

```
PS C:\Users\Administrator> New-Item -Path "$HOME\sorted/video" -ItemType Directory

Directory: C:\Users\Administrator\sorted

Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         17:35         video

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted/audio" -ItemType Directory

Directory: C:\Users\Administrator\sorted

Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         17:35         audio

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted/books" -ItemType Directory

Directory: C:\Users\Administrator\sorted

Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         17:35         books

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted/undefined" -ItemType Directory

Directory: C:\Users\Administrator\sorted

Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         17:35         undefined
```

Рисунок 9 - Создание каталога sorted с подкаталогами

```
PS C:\Users\Administrator> Move-Item "$HOME\unsorted\*.mp3" "$HOME\sorted\audio"
PS C:\Users\Administrator> Move-Item "$HOME\unsorted\*.mp4" "$HOME\sorted\video"
PS C:\Users\Administrator> Move-Item "$HOME\unsorted\*.pdf" "$HOME\sorted\books"
```

Рисунок 10 - Перемещение файлов по категориям

```

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted\video\80x" -ItemType Directory

Directory: C:\Users\Administrator\sorted\video

Mode                LastWriteTime         Length Name
----                -
d-----          22.05.2025      17:40             80x

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted\video\200x" -ItemType Directory

Directory: C:\Users\Administrator\sorted\video

Mode                LastWriteTime         Length Name
----                -
d-----          22.05.2025      17:40             200x

PS C:\Users\Administrator> New-Item -Path "$HOME\sorted\video\latest" -ItemType Directory

Directory: C:\Users\Administrator\sorted\video

Mode                LastWriteTime         Length Name
----                -
d-----          22.05.2025      17:40             latest

```

Рисунок 11 - Создание подкаталогов в video

```

PS C:\Users\Administrator> Get-ChildItem "$HOME\sorted\video" -Filter "*198[0-9]*" | Move-Item -Destination "$HOME\sorted\video\80x"
PS C:\Users\Administrator> Get-ChildItem "$HOME\sorted\video" -Filter "*200[0-9]*" | Move-Item -Destination "$HOME\sorted\video\200x"
PS C:\Users\Administrator> Get-ChildItem "$HOME\sorted\video" -Filter "*201[4-6]*" | Move-Item -Destination "$HOME\sorted\video\latest"

```

Рисунок 12 - Сортировка видеофайлов по годам

```

PS C:\Users\Administrator> Get-ChildItem "$HOME\sorted\video" | Move-Item -Destination "$HOME\sorted\undefined"
PS C:\Users\Administrator> |

```

Рисунок 13 - Перемещение несортированных файлов

```
PS C:\Users\Administrator> Compress-Archive -Path "$HOME\sorted" -DestinationPath "$HOME\task.completed.zip"
```

```
PS C:\Users\Administrator> Get-ChildItem "$HOME" -Recurse
```

Directory: C:\Users\Administrator

Mode	LastWriteTime	Length	Name
d-r---	21.05.2025 23:06		Contacts
d-----	22.05.2025 16:29		current
d-r---	21.05.2025 23:06		Desktop
d-----	22.05.2025 16:26		dir1
d-r---	21.05.2025 23:06		Documents
d-r---	21.05.2025 23:06		Downloads
d-r---	21.05.2025 23:06		Favorites
d-r---	21.05.2025 23:06		Links
d-r---	21.05.2025 23:06		Music
d-----	22.05.2025 16:28		new
d-----	22.05.2025 16:28		old
d-r---	21.05.2025 23:06		Pictures
d-----	22.05.2025 16:26		qqq
d-r---	21.05.2025 23:06		Saved Games
d-r---	21.05.2025 23:06		Searches
d-----	22.05.2025 17:35		sorted
d-----	22.05.2025 17:38		unsorted
d-r---	21.05.2025 23:06		Videos
-a----	22.05.2025 16:14	0	dir2
-a----	22.05.2025 16:14	0	dir3
-a----	22.05.2025 16:35	306	directories.zip
-a----	22.05.2025 17:46	866	task.completed.zip

Directory: C:\Users\Administrator\current

Mode	LastWriteTime	Length	Name
d-----	22.05.2025 16:29		qqq

Directory: C:\Users\Administrator\dir1

Mode	LastWriteTime	Length	Name
d-----	22.05.2025 16:26		dir2
-a----	22.05.2025 16:14	0	file1
-a----	22.05.2025 16:14	0	file2

Directory: C:\Users\Administrator\dir1\dir2

Mode	LastWriteTime	Length	Name
d-----	22.05.2025 16:26		dir3
-a----	22.05.2025 16:14	0	file1
-a----	22.05.2025 16:14	0	file2

Directory: C:\Users\Administrator\dir1\dir2\dir3

Directory: C:\Users\Administrator\dir1\dir2\dir3

Mode		LastWriteTime	Length	Name
----		-----	-----	----
-a----	22.05.2025	16:14	0	file1
-a----	22.05.2025	16:14	0	file2

Directory: C:\Users\Administrator\Favorites

Mode		LastWriteTime	Length	Name
----		-----	-----	----
d-r---	21.05.2025	23:06		Links
-a----	21.05.2025	23:06	208	Bing.url

Directory: C:\Users\Administrator\Links

Mode		LastWriteTime	Length	Name
----		-----	-----	----
-a----	21.05.2025	23:06	518	Desktop.lnk
-a----	21.05.2025	23:06	975	Downloads.lnk

Directory: C:\Users\Administrator\new

Mode		LastWriteTime	Length	Name
----		-----	-----	----
d-----	22.05.2024	16:32		qqq

Directory: C:\Users\Administrator\old

Mode		LastWriteTime	Length	Name
----		-----	-----	----
d-----	22.05.2025	16:28		qqq

Directory: C:\Users\Administrator\sorted

Mode		LastWriteTime	Length	Name
----		-----	-----	----
d-----	22.05.2025	17:38		audio
d-----	22.05.2025	17:38		books
d-----	22.05.2025	17:44		undefined
d-----	22.05.2025	17:44		video

Directory: C:\Users\Administrator\sorted\audio

Mode		LastWriteTime	Length	Name
----		-----	-----	----
-a----	22.05.2025	17:33	0	example.mp3

Directory: C:\Users\Administrator\sorted\books

```
Directory: C:\Users\Administrator\sorted\books
Mode                LastWriteTime         Length Name
----                -
-a-----         22.05.2025         17:33             0 example.pdf

Directory: C:\Users\Administrator\sorted\undefined
Mode                LastWriteTime         Length Name
----                -
d-----         22.05.2025         17:40          200x
d-----         22.05.2025         17:40           80x
d-----         22.05.2025         17:40         latest
-a-----         22.05.2025         17:33             0 example.mp4

Directory: C:\Users\Administrator\unsorted
Mode                LastWriteTime         Length Name
----                -
-a-----         22.05.2025         17:33             0 random_file.txt

PS C:\Users\Administrator>
```

Рисунки 14-16 - Создание архива с результатами и вывод содержимого текущего каталога

Контрольные вопросы

1. Что такое файловая система?

Файловая система – это метод организации, хранения и управления данными на носителе информации (жесткий диск, SSD, флеш-накопитель и т. д.).

2. Каково назначение файловых систем ОС?

Они помогают структурировать данные, обеспечивают доступ, безопасность, целостность файлов, а также эффективное использование дискового пространства.

3. Что такое файл?

Файл – это именованный набор данных, записанных на носитель, например текстовый документ, изображение или видео.

4. Что такое каталог?

Каталог (папка) – это контейнер, в котором хранятся файлы и другие каталоги, позволяя организовать данные и структурировать их.

5. Каковы правила именования файлов?

В Windows имя файла может содержать буквы, цифры и некоторые специальные символы, но запрещены: / \ : * ? " < > |. Максимальная длина имени – 255 символов.

6. Какие бывают файловые операции?

К ним относятся создание, чтение, изменение, переименование, копирование и удаление файлов.

7. Какие операции выполняются для управления каталогами?

Можно создавать, удалять, перемещать, переименовывать и просматривать содержимое каталогов.

8. Какие основные командлеты используются для работы с файлами в Windows?

- Get-Item – получение информации о файле
- New-Item – создание файла

- Remove-Item – удаление файла
- Copy-Item – копирование файла
- Move-Item – перемещение файла

9. Какой параметр помогает работать внимательно при удалении или перемещении файлов в Windows?

В PowerShell можно использовать параметр -Confirm, который запрашивает подтверждение перед выполнением команды.

10. Какие основные командлеты используются для работы с каталогами в Windows?

- Get-ChildItem – просмотр содержимого
- New-Item -ItemType Directory – создание каталога
- Remove-Item – удаление каталога
- Set-Location – переход в каталог

11. Как определить текущий каталог в Windows?

В PowerShell – с помощью Get-Location, а в командной строке – cd без параметров.

12. Какие структуры каталогов существуют в Windows?

Windows использует иерархическую структуру, начиная с корневого (C:\) и далее подразделяя каталоги (C:\Users, C:\Program Files и т. д.).