**САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**КУРСОВАЯ РАБОТА**

**«Использование пространств хранения данных (unix и window)»**

Выполнил: студент 3 курса ИВЧТ-31

Кузнецов А.А.

Проверил:

**Саратов 2024**

**Содержание**

Глава 1. Типы систем хранения данных Unix и Windows

1.1 Основные типы систем хранения в Unix (ext… https://selectel.ru/blog/directory-structure-linux/)

1.2 Типы систем хранения данных в Windows (ntfs, fat более простая)

1.3 Сравнение архитектурных особенностей систем хранения Unix и Windows

Глава 2. Управление данными в Unix и Windows

2.1 Схожесть управления данными (mkdir, rm, …

2.2 Различия

2.3 Сравнение подходов к управлению данными в Unix и Windows (когда какой будет удобнее)

Глава 3. Производительность и безопасность систем хранения

3.1 Показатели производительности систем хранения в Unix

3.2 Безопасность и защита данных в Windows

3.3 Сравнительный анализ производительности и безопасности Unix и Windows

Глава 4. Современные тенденции в использовании пространств хранения:

Облачные хранилища, Виртуализация, Новые типы устройств хранения: флеш-память(тут fat если памяти мало), SSD(и то и то удобно), NVMe

**Содержание**

Введение

Глава 1. Типы систем хранения данных Unix и Windows

1.1 Основные типы систем хранения в Unix

1.2 Типы систем хранения данных в Windows

1.3 Сравнение архитектурных особенностей систем хранения Unix и Windows

Глава 2. Управление данными в Unix и Windows

2.1 Методы управления данными в Unix

2.2 Особенности управления данными в Windows

2.3 Сравнение подходов к управлению данными в Unix и Windows

Глава 3. Производительность и безопасность систем хранения

3.1 Показатели производительности систем хранения в Unix

3.2 Безопасность и защита данных в Windows

3.3 Сравнительный анализ производительности и безопасности Unix и Windows

Заключение

Список литературы

**Введение**

Операционная система (operating system) – это комплекс программ, предоставляющий пользователю удобную среду для работы с компьютерным оборудованием.

Операционная система позволяет запускать пользовательские программы; управляет всеми ресурсами компьютерной системы – процессором (процессорами), оперативной памятью, устройствами ввода вывода; обеспечивает долговременное хранение данных в виде файлов на устройствах внешней памяти; предоставляет доступ к компьютерным сетям.

Файлы в файловых системах хранятся в директориях (каталогах, «папках»). Большая часть этой информации, особенно та, которая непосредственно относится к хранению, управляется файловой системой. Сама директория – это тоже служебный файл, к которому можно получить доступ при помощи различных административных процедур.

Самым большим преимуществом современных файловых систем хранения является их интуитивная понятность: интерфейс выглядит также, как и файловый менеджер на любом компьютере, и принцип иерархичности (вложенность) папок с файлами тоже взят из обыденной жизни.

Система достаточно легко масштабируется(не все) (до определенных пределов). Совместный доступ пользователей внутри масштабов системы практически ничем не ограничен. К достоинствам также можно отнести относительно невысокую цену.

Файлы в файловой системе ищутся достаточно быстро, если ее масштаб не запределен.(опять зависит от типа) Система имен файлов позволяет разным собственникам иметь файлы с одинаковым именем – с точки зрения системы это будут разные файлы – даже в том случае, если содержимое файлов с одинаковыми владельцами будет одинаковым.

К недостаткам файловой системы хранения следует отнести наличие определенных пределов масштабирования. При росте объема системы навигация становится более сложной, а время доступа к файлам увеличивается. То есть файлы будут открываться медленнее.

В данной курсовой работе будет проведен сравнительный анализ использования пространств хранения данных в операционных системах unix и windows.

**Глава 1. Типы систем хранения данных Unix и Windows**

Взрывной рост объемов данных – одна из основных тенденций развития современных ИТ-систем. Помимо роста объемов, наблюдается рост значимости данных, и, как следствие, появляется необходимость в бесперебойном доступе к данным, к защите от аварий и катастроф. При этом на передний план выходят так же вопросы стоимости хранения данных и сохранения инвестиций в инфраструктуру. Все эти задачи успешно решаются за счет применения передовых принципов дизайна систем хранения, таких как консолидации и виртуализации вычислительных ресурсов.

**Система хранения данных (СХД)** — комплекс аппаратных и программных средств, который предназначен для хранения и оперативной обработки информации, как правило, большого объема. Информация — это файлы, в том числе медиа, структурированные (СУБД) и неструктурированные данные (big data), резервные копии, архивы. В качестве носителей информации используются жесткие диски, в основном SSD (системы All Flash Array), а также гибридные решения, сочетающие SSD- и HDD-накопители в одной СХД. СХД позволяет получить доступ к хранящимся данным из разных узлов сети, обеспечивает высокую скорость передачи информации и легко масштабируется. Она широко используются в корпоративных сетях и центрах обработки данных для обеспечения надежного и эффективного хранения информации.

### **Уровни хранения**

Сюда бы вводный абзац

#### **Блочное хранилище**

СХД используется как обычный диск, который можно форматировать, устанавливать на него ОС, создавать логические диски. Данные хранятся не файлами, а блоками, что ускоряет операции ввода-вывода. Чаще используется в сетях типа SAN (Storage Attached Network). Подходит для высокопроизводительных вычислений, СУБД, хранения больших объемов данных, в качестве сред разработки (Dev/Test). Из недостатков: а) сложность настройки и обслуживания, которые требуют соответствующей квалификации; б) высокая стоимость.

#### **Файловое хранилище**

Данные хранятся в виде файлов, которые размещаются в каталогах. Такая СХД используется для хранения «холодной» информации, которая не требуется для операционных вычислений. На файловых хранилищах, как правило, строятся NAS (Network Attached Storage). Недостатки: при накоплении больших объемов данных усложняется иерархия папок, и скорость работы СХД постепенно снижается. Не подходит для нагрузок, которые требуют высокой скорости отклика.

#### **Объектное хранилище**

Тип СХД, который ориентирован на работу с большими неструктурированными данными объемом до петабайтов. Информация хранится не в виде файлов, а в виде «объектов» с уникальными идентификатором и метаданными. Поэтому объектное хранилище похоже по структуре на БД. Используется в аналитике, big data, машинном обучении, для хранения «тяжелых» медиа-файлов и резервных копий, разработки и эксплуатации приложений в облаке, хостинга веб-сайтов. По скорости уступает блочному хранилищу в задачах, связанных с транзакционными нагрузками.

**1.1 Основные типы систем хранения в Unix**

В зависимости от того, что необходимо, — быстродействие, высокая гарантия восстановления данных или производительность, можно выбрать в системе Linux стандартную файловую систему для конкретного раздела, специальную или виртуальную.

## Файловые системы в операционной системе Linux. Базовые понятия

ОС Linux предоставляет выбор еще на стадии установки: в ядро системы встроены разные файловые системы (ФС). При этом пользователь должен выбрать ту, что отвечает его требованиям и задачам. Перед теми, кто использует Windows, такой вопрос не стоит — эту современные версии этой ОС можно установить только на NTFS. Отличается от Windows и иерархическое устройство самих ФС, и структура каталогов.

Linux поддерживает деление жесткого диска на разделы. Для подсчета и определения физических границ используется специальная таблица разделов — GPT или MBR. Она содержит метку и номер раздела, а также адреса физического расположения точек начала и конца раздела.

## Организация файловой системы Linux

В Linux на каждый раздел можно установить свою ФС, которая отвечает за порядок и способ организации информации. В основе файловых систем лежит набор правил, определяющий, где и каким образом хранятся данные. Следующий «слой» ФС — практический (технический) способ организации информации на каждом конкретном типе носителя (опять же, учитывая правила, заложенные в основу системы).

От выбора файловой системы зависят:

* скорость работы с файлами;
* их сохранность;
* скорость записи;
* размер файлов.

Тип ФС также определяет, будут ли данные храниться в оперативной памяти (ОП) и как именно пользователь сможет изменить конфигурацию ядра.

**Файловая система** (ФС) — архитектура хранения данных, которые могут находиться в разделах жесткого диска и ОП. Определяет, какую структуру принимают файлы в каждом из разделов, создает правила для их генерации, а также управляет файлами в соответствии с особенностями каждой конкретной ФС.

ФС Linux — пространство раздела, поделенное на блоки определенного размера. Он определяется кратностью размеру сектора. Соответственно, это могут быть 1024, 2048, 4096 или 8120 байт. Важно помнить, что размер каждого блока известен изначально, ограничен максимальным размером ФС и зависит от требований, которые выдвигает пользователь к каждому из блоков.

Для обмена данными существует сразу два способа. Первый из них — виртуальная файловая система (VFS). С помощью данного типа ФС происходит совместная работа ядра и приложений, установленных в системе. VFS позволяет пользователю работать, не учитывая особенности каждой конкретной ФС. Второй способ — драйверы файловых систем. Именно они отвечают за связь между «железом» и софтом.

Список файловых систем, которые поддерживаются ядром, находится в файле **/proc/filesystems**:

user@user-VirtualBox:~$ cat /proc/filesystems

nodev sysfs

nodev rootfs

nodev ramfs

nodev bdev

nodev proc

nodev cpuset

nodev cgroup

nodev cgroup2

nodev tmpfs

nodev devtmpfs

nodev debugfs

nodev tracefs

nodev securityfs

nodev sockfs

nodev bpf

nodev pipefs

nodev hugetlbfs

nodev devpts

ext3

ext2

ext4

squashfs

vfat

nodev ecryptfs

fuseblk

nodev fuse

nodev fusectl

nodev pstore

nodev mqueue

nodev autofs

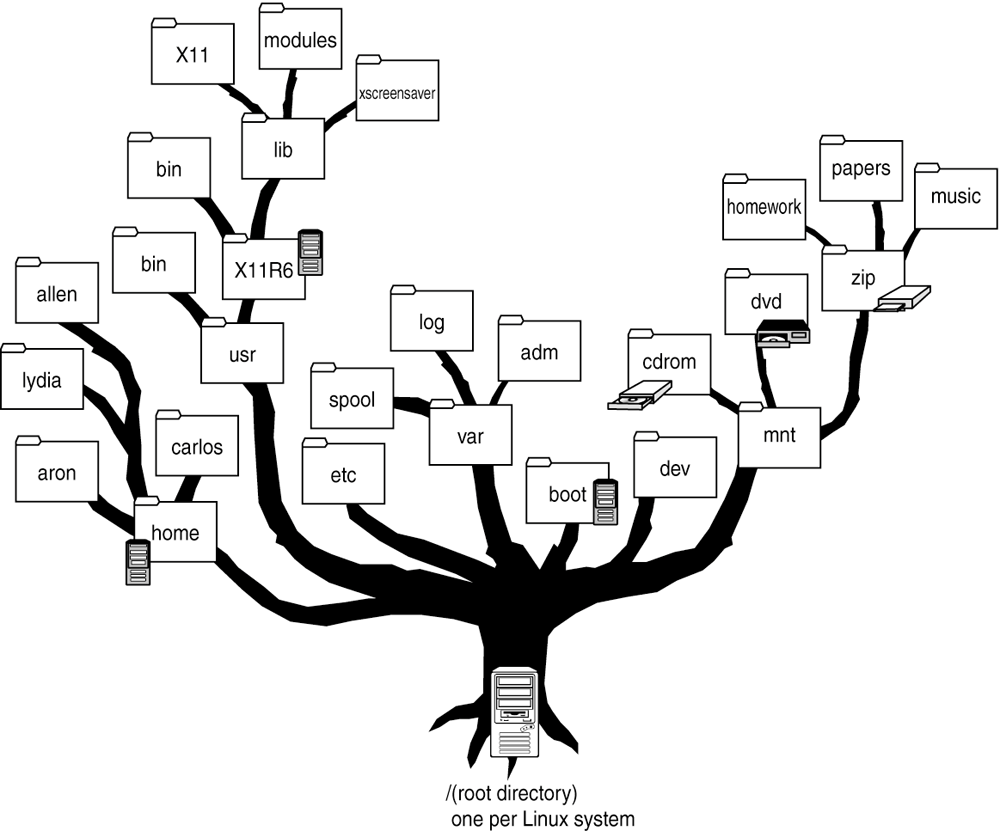
## Структура и иерархия файловой системы. Структура каталога

Файловая система в Linux определяет также организацию расположения файлов, по сути представляя собой иерархическую структуру «дерева»: начинается с корневого каталога «/» и разрастается ветвями в зависимости от работы системы.

ФС также характерно понятие целостности: в такой системе изменения, внесенные в один файл, не приведут к изменению другого файла, не связанного с первым. У всех данных есть собственная физическая память. В Linux целостность ФС проверяется специальной командой — ***fsck***.

Типы файлов условно можно разделить на несколько групп. Некоторые из них такие же, как и в ОС Windows, — текстовые документы, медиа и изображения. Отличия начинаются с каталогов, которые являются отдельным типом файлов. Жесткие диски относят к блочным устройствам. Принтеры — к символьным. Отдельную группу составляют символические ссылки, о которых речь пойдет ниже.

Тип файла определяется с помощью команды **ls** (параметр -l).

Структура каталогов в Linux

В ФС каждый файл определяется конкретным индексом — Inode (от англ. index node — «индексный узел»). Но при этом один файл (речь о физическом размещении) может иметь сразу несколько имен (или путей). И если в структуре ФС файлы будут отличаться, то на жестком диске им может соответствовать один файл. Это означает, что ФС Linux перекрестно-иерархична, а ветви дерева могут пересекаться.

Корневой раздел в Linux один — «/» (root, «корень)». Разделы называются подкаталогами, примонтированными к соответствующим каталогам. Типовая структура каталогов (первых двух уровней), примонтированных к корневому каталогу Linux:

.

├── bin -> usr/bin

├── boot

│ ├── grub

│ └── lost+found

├── dev

│ ├── block

│ └── ...

├── etc

│ ├── ...

│ ├── update-manager

│ ├── update-motd.d

│ └── xdg

├── home

├── lib -> usr/lib

├── lib32 -> usr/lib32

├── lib64 -> usr/lib64

├── libx32 -> usr/libx32

├── lost+found

├── media

├── mnt

├── opt

├── proc

│ ├── ...

│ └── tty

├── root

├── run

│ └── ...

├── sbin -> usr/sbin

├── srv

├── sys

├── tmp

│ └── ...

├── usr

└── var

При монтировании происходит ассоциирование каталога с устройством, содержащим ФС (драйвер). Соответствующая ссылка на устройство передается драйверу. Именно он и определяет ФС. Если процедура завершается успешно, ядро заносит информацию (каким драйвером обслуживаются и где расположены файлы и каталоги) в таблицу монтирования. Она находится в файле /proc/mounts.

Данные о каждом файле содержит Inode — специфичный для UNIX-систем индексный дескриптор, хранящий различную метаинформацию (владелец файла, последнее время обращения, размер и так далее).

Когда файл (каталог) перемещается в другую ФС, его Inode тоже создается заново. И только потом удаляется исходный (в рамках той же системы меняется только путь файла). Также отметим, что файл (каталог) существует до того момента, пока хранится информация о его имени или пути к нему. После удаления всей информации блоки, отведенные под файл, становятся свободными (для выделения под другой файл).

Общая информация о ФС хранится в суперблоке. Сюда относится суммарное число блоков и Inode, число свободных блоков, их размеры и так далее. Важно, чтобы суперблок сохранял свою целостность, поскольку от этого зависит стабильность и работоспособность системы в целом. В ОС создается сразу несколько копий, чтобы можно было восстановить всю необходимую информацию.

При загрузке ядро автоматически монтирует разделы после того, как корень уже примонтирован на чтение. Информацию ядро считывает из конфигурации /etc/fstab.

### **Команды для работы с файлами в Linux**

|  |  |
| --- | --- |
| **Команда** | **Что делает** |
| ls | Просматривает содержимое текущего каталога |
| touch file\_name | Создает файл file\_name |
| mkdir directory\_name | Создает директорию directory\_name |
| cat file\_name | Показывает содержимое файла file\_name в терминале |
| less file\_name | Обеспечивает просмотр файла с помощью скроллинга |
| rm file\_name | Удаляет файл file\_name |
| rm -r | Удаляет рекурсивно все файлы из директории |
| rmdir directory | Удаляет папку directory, которая находится в текущей папке |
| ln -s /home/user/directory\_name/ /home/user/test/ | Создает жесткие и символические ссылки на файлы или папки. Для создания символической ссылки используется опция -s |
| pwd | Выводит каталога, в котором находится пользователь |
| which program | Выводит каталог, в котором установлена программа |
| mc | Запускает полнофункциональный файловый менеджер с псевдографическим интерфейсом на основе ncurses. Требуется установка mc в Ubuntu |
| cd directory\_name | Переходит в директорию directory\_name |
| cp file\_name directory\_name | Копирует file\_name в директорию directory\_name |
| nano | Инициирует запуск простейшего текстового редактора командной строки Linux |
| mv file\_name directory\_name | Перемещает file\_name в директорию directory\_name |
| mv old\_name new\_name | Переименовывает файл/директориюold\_name в new\_name |
| locate file\_name | Выполняет быстрый поиск файла |
| chmod num file\_name | Изменяет права доступа к файлу или каталогу |

## Типы файловых систем Linux. Какая из файловых систем используется ядром Linux

Как уже говорилось ранее, в Linux несколько предустановленных и доступных ФС. В зависимости от выбора пользователя будут меняться методы работы с файлами, обращения к конфигурации ядра и способы хранения данных в ОП. В зависимости от целей и задач пользователя (а также достоинств и недостатков самих ФС) можно выбрать любую файловую систему, доступную в дистрибутиве ОС.

Список основных файловых систем:

* Ext2;
* Ext3;
* Ext4;
* JFS;
* ReiserFS;
* XFS;
* Btrfs;
* ZFS.

ФС может являться корневой в различных разделах, Linux позволяет использовать разные системы одновременно.

### **Два типа файловых систем**

**Журналируемые** — данный тип ФС сохраняет историю действий пользователя, а также план проверки системы в специальном файле. Особенности: устойчивость к сбоям и сохранение целостности информации.

**Не журналируемые** — не предусматривают хранение логов. Особенности: работают быстрее, но не гарантируют сохранность данных.

Чтобы узнать тип ФС, существует команда ***file -s***.

**JFS** Журналируемая ФС — первая альтернатива для ФС группы Ext. Ее разработали в IBM специально для операционной системы AIX UNIX. Главные плюсы этой системы: стабильность и минимальные требования для работы. Разработчики JFS ставили перед собой цель создать ФС, которая бы эффективно работала на многопроцессорных компьютерах. Кроме того, эта система также относится к журналируемым ФС. Но есть и очевидные недостатки. Если случится непредвиденный сбой в работе системы, ФС может использовать версии файлов, которые уже устарели. Причина заключается в том, что журнал сохраняет только метаданные.

**ReiserFS** Эта ФС разработана под руководством Ганса Райзера и названа в честь него. Подходит исключительно под Linux, чаще всего ее используют в качестве возможной замены Ext3. Главные особенности: увеличенная производительность и более широкие возможности. Изменяющийся размер блока дает пользователю возможность объединять небольшие файлы в один блок, таким образом удается избежать фрагментации и повысить качество работы ФС в целом. Размер разделов можно менять прямо в процессе работы, однако эта ФС может показать нестабильные результаты и потерять данные, например, при отключении энергии.

**XFS** Еще одна журналируемая ФС. Однако, в отличие от аналогов, в логи записывает исключительно те изменения, которые претерпевают метаданные. Разработана для ОС в Silicon Graphics. Важные особенности: быстро работает с файлами сравнительно большого размера, умеет выделять место в отложенном режиме, а также менять размеры разделов в процессе работы. Часто встречается в дистрибутивах на основе Red Hat. Минусы: нельзя уменьшить размер разделов, сложно восстанавливать данные и можно потерять информацию при отключении питания.

**Btrfs** Современная ФС, главной особенностью которой является высокая отказоустойчивость. Из дополнительных «бонусов»: удобна для сисадминов и поддерживает сравнительно простой процесс восстановления данных. Поддерживает подтома, разрешает менять размеры разделов в динамическом режиме и позволяет делать снапшоты. Отличается высокой производительность. Применяется как ФС, установленная по умолчанию, в OpenSUSE и SUSE Linux. Главный минус — нестабильность (нарушена обратная совместимость, сложная для поддержки и так далее).

**F2FS** Flash-Friendly File System входит в состав ядра ОС Linux и предназначена для использования с хранилищем на основе флеш-памяти. Разработчик — корпорация Samsung. F2FS разбивает носитель на части, которые снова делятся, и так далее. Эти миниатюрные зоны используются вместо повторного использования одних и тех же размеченных участков.

**OpenZFS** OpenZFS — ветвь ZFS, о которой мы пишем ниже). Разработчик — компания Sun для ОС Solaris. В 2016 году Ubuntu включила ее поддержку по умолчанию. Главные плюсы: защита от повреждения данных, поддержка больших файлов и автоматическое восстановление.

Традиционные для Windows ФС NTFS, FAT, HFS применяются в Linux, но пользователь не сможет установить в такие разделы корень, поскольку структура этих ФС для этого не приспособлена.

## Специальные файловые системы

Для решения задач, связанных с предоставлением доступа пользователю или программам к настройкам ядру ОС, используются так называемые специальные файловые системы. Ядро использует несколько типов специальных ФС:

* **tmpfs** — записывает файлы в ОП. Для этого создается блочное устройство определенного объема, после чего оно подключается к папке.
* **procfs** — хранит данные о системных процессах и ядре.
* **sysfs** — изменяет настройки ядра ОС.

## Виртуальные файловые системы: EncFS, Aufs, NFS и ZFS

Если пользователю необходимо решить задачи, которые не требуют непременного наличия ФС в ядре, применяется модуль FUSE (filesystem in userspace). Он создает ФС в пространстве пользователя. Виртуальные ФС, как правило, поддерживают шифрование и сетевое администрирование. Сегодня на рынке существует целый спектр виртуальных ФС для ряда задач:

* **EncFS** — шифрует файлы, а затем выполняет сохранение зашифрованных файлов в необходимую пользователю директорию.
* **Aufs (AnotherUnionFS)** — объединяет несколько ФС (то же самое может делать с папками) в одну.
* **NFS (Network Filesystem)** — выполняет монтирование ФС удаленно.
* **ZFS (Zettabyte File System)** — ФС, созданная для ОС Solaris. Главные плюсы: отсутствие фрагментации, управление снапшотами и пулами хранения, изменяющийся размер блоков. Используется посредством FUSE.

**1.2 Типы систем хранения данных в Windows**

Форматирование флешки или системного накопителя — стандартная задача компьютерного пользователя. В Windows процесс достаточно прост. Система самостоятельно определяет, какие настройки подходят определенному устройству и какую файловую систему выбрать при форматировании. Так, Windows форматирует системные накопители в NTFS, а флешки превращает в FAT32. Чем отличаются эти файловые системы и зачем диску нужен «формат»?

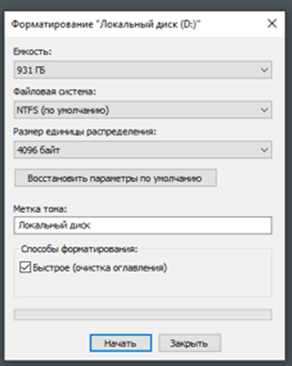
Если говорить простым языком, то компьютерный накопитель — это подобие библиотеки, в которой хранятся тысячи книг. Библиотека может быть устроена в виде небольшого стеллажа или многоэтажной полки с лестницей, а также в виде кластеров — огромных помещений с десятками шкафов и сотнями полок. Чтобы найти в таком масштабе интересующую книгу, необходимо ориентироваться по условным опознавательным знакам, буквам или цифрам.

## NTFS — New Technology File System

**NTFS** — фирменная файловая система Microsoft, которую разработчики начали внедрять в операционную систему Windows, начиная с версии NT 3.1. Несмотря на хрупкость и низкую отказоустойчивость системы, NTFS считается самым лучшим и удачным решением для работы актуальных операционных систем Windows. Конечно, как и любая другая система, NTFS не лишена недостатков — это слишком сложное устройство ФС, особенно по современным меркам. Ведь известно — чем сложнее устройство, тем больше в нем уязвимостей.

**Структура и фрагментация**

Файловая система NTFS делит пространство накопителя на кластеры — блоки, размером от 512 байт до 64 КБ. По умолчанию Windows делит блоки по 4 КБ каждый.



Способ организации файлового пространства на диске с NTFS подразумевает наличие специального раздела, в котором ФС хранит сервисные данные о своей работе. А именно, ведет некий каталог, в котором записываются различные данные о файлах и разделах. Это раздел MFT (Master File Table) — свободное пространство с метафайлом, под который система выделяет 12% от общего объема.

MFT является динамическим разделом — по мере накопления информации на диске, он может сокращаться, чтобы освободить место под пользовательские файлы. Однако при первом же свободном гигабайте на диске, раздел MFT снова заберет свое «законное» место, при этом новая часть метафайла может фрагментироваться и оказаться уже не в начале диска, а в конце или в середине. Отсюда существует распространенная проблема фрагментации файловой системы, когда части каталогов разбросаны по всему диску. Тогда, чтобы найти какой-либо файл, диск судорожно ищет их по всей поверхности, отсюда снижение скорости доступа и общей производительности компьютера. Фрагментация — не самая сильная сторона NTFS.

**Файлы и каталоги**

Организация данных в этой ФС имеет структуру бинарного дерева: каждый элемент в системе обрабатывается не иерархически, а через бинарные запросы. Например, чтобы найти файл с именем «К» среди тысячи других файлов, система делит каталог на две части и начинает поиск с середины. Например, узнает, в какой части необходимо искать данный файл, если за середину каталога принят файл с названием «Т»? В таком случае система ответит — ищите среди тех файлов, которые идут до файла с именем «Т». То есть, имея отсортированный по алфавиту каталог, система понимает, что файл с необходимым именем находится в одной из двух частей, и время на поиск файла сокращается в два раза — это улучшает скорость работы с мелкими одиночными файлами.

Все файлы в этой системе существуют в виде потоков. Фактически, для того, чтобы превратить блоки с данными в единый файл, этой ФС необходим только файл с метаданными. Это своего рода инструкция по сборке файлов из кусочков данных, которые хранятся в ячейках по всей поверхности накопителя. Благодаря гибкой файловой структуре, объекты NTFS могут принимать множество дополнительных свойств. Например, содержать в названии до 65535 различных символов Unicode. При этом максимальная длина имени файла достигает 255 символов.

**Журналирование**

Современные операционные системы работают на базе журналируемых файловых систем. Это необходимо для того, чтобы в случае системного сбоя и аварийного завершения работы (вынули вилку питания ПК из розетки) файловая система компьютера смогла восстановиться до последнего рабочего состояния без потери файлов.

В журналируемой файловой системе работа с данными происходит по принципу транзакций — действие совершается полностью или не совершается совсем. Например, при записи системного файла на диск, компьютер делает пометки в метафайл в разделе MTF и ведет мини-журнал процесса копирования до тех пор, пока файл полностью не запишется в необходимый раздел диска. Если устройство перезагрузится во время записи, то при следующем включении система обратится к журналу, узнает о совершенных и несовершенных транзакциях и оставит существовать только те, которые помечены как завершенные. Остальные транзакции будет вычеркнуты, а файлы удалены или возвращены на место.

Как правило, такая система работает наиболее эффективно только с системными файлами, тогда как пользовательские данные могут повредиться или исчезнуть при сбое. Работу журналирования можно проверить с помощью контрольных точек восстановления — компьютер периодически создает слепки состояния системы, по которым позже может восстановиться до этих состояний.

**Файловая система NTFS также отличается:**

* Поддержкой больших томов и файлов — до 8 ПБ;
* Несколькими уровнями безопасности, в том числе, поддержкой шифрования;
* Распределением прав доступа к файлам;
* Возможностью сжатия;
* Поддержкой до 4 294 967 295 (232−1) файлов.

Вывод: система NTFS «заточена» под работу с операционной системой, а также для накопителей с большим объемом и несколькими разделами.

## FAT32 — File Allocation Table

Обновленная файловая система пришла на смену устаревшей FAT16. Ее также разработали специалисты Microsoft, но, в отличие от NTFS, она распространяется в виде открытого исходного кода. Поэтому разработчики любого софта могут беспрепятственно компилировать и внедрять драйвер в свое ПО. Например, поддержка FAT32 есть не только в «родной» операционной системе, но и в [любой другой](https://club.dns-shop.ru/blog/t-123-vneshnie-hdd/27824-kak-podkluchit-vneshnii-jestkii-disk-k-televizoru/) — linux, MacOS, Android, даже в таких проприетарных системах, как iOS.

В ранних версиях ОС Windows файловая система FAT32 даже использовалась в качестве основной ФС для системного раздела. Но позже разработчики отказались от этого решения в пользу новой и прогрессивной NTFS. Впрочем, несмотря на некоторые особенности FAT32, эта ФС все еще повсеместно используется во флешках и картах памяти.

Файловая система FAT32 максимально упрощена. Здесь нет продвинутых систем безопасности и шифрования, система не умеет журналировать свою работу. Это частично сказывается на производительности — в некоторых случаях скорость чтения или записи может быть выше, чем у NTFS.

Однако, файловая система не умеет работать с объемными разделами. Например, штатные средства ОС Windows не позволяют создавать на диске с FAT32 разделы, объем которых превышает 32 ГБ. К этим недостаткам относятся и ограничения по максимальному размеру файла. Максимальный размер файла, который запоминает накопитель, составляет 4 ГБ.

Все это, конечно же, влияет на популярность файловой системы и ее удобство. Особенно заметны недостатки устаревшей системы стали после того, как почти у каждого пользователя в арсенале появились флешки с объемом от 64 ГБ.

**Структура**

Еще больше красок в устаревание вносит древняя структура: файлы в FAT32 хранятся иерархически, а не в виде бинарного дерева, где каждый отдельный объект может быть доступен независимо от остальных.

Несмотря на перечисленные особенности, FAT32 все еще неплохо справляется со своими задачами. Например, отсутствие журналирования идет на пользу накопителям, которые [быстро изнашиваются](https://club.dns-shop.ru/blog/t-107-jestkie-diski/49522-diagnostika-nakopitelei-opisanie-parametrov-s-m-a-r-t/) от частых перезаписей ячеек. К тому же, работа с объемными файлами и разделами на обычной флешке мало кого интересует. Как правило, они «переносят» легковесные офисные файлы, фотографии, короткие видеоматериалы и установочные файлы программ. Вряд ли кто-то попытается загружать образ фильма в формате Blu-ray на флешку: для этого больше подойдет внешний жесткий диск или твердотельный накопитель с большим объемом.

Вывод: если выбирать файловую систему, то лучше ориентироваться на практичность. Для серьезных задач и под системные нужды обязательно выделять накопитель с NTFS на борту. В то же время, для флешки с маленьким объемом будет достаточно и FAT32 — эта ФС широко поддерживается всеми возможными устройствами.

**3.3 Сравнительный анализ производительности и безопасности Unix и Windows**

Иерархия каталогов: Структура каталогов в Linux обычно более строгая и согласованная, чем в Windows. Например, в Linux файлы программ находятся в /usr/bin или /bin, а пользовательские данные - в /home, в то время как в Windows файлы программ могут быть распределены по различным папкам, например, C:\Program Files.

Командная строка: Linux традиционно ориентирована на командную строку, что обеспечивает больше возможностей для управления файлами и настройками. Windows также имеет командную строку, но она не так распространена и мощна.

Системы прав доступа: Linux использует более детализированную систему прав доступа (владелец, группа, другие), предоставляя более точный контроль над доступом к файлам.

Управление дисками: Linux использует более гибкие методы управления дисками и разделами, позволяя пользователям легко создавать, удалять и изменять разделы.

Инструменты управления файлами: Linux предлагает более широкий спектр инструментов для управления файлами, такие как find, grep, sed, которые облегчают поиск, фильтрацию и обработку данных.

Безопасность: Linux в целом считается более безопасной операционной системой, чем Windows, благодаря открытому исходному коду и более строгому контролю над правами доступа.