МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №1

«Механизмы ОС Microsoft Windows»

по дисциплине

«Эксплуатация современных операционных систем»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Кочешков А. А.\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухоруков В.А.\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_19-В-2\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

Оглавление

[Цель работы 4](#_Toc101735168)

[Ход работы 4](#_Toc101735169)

[1. Подготовка к установке ОС Windows 4](#_Toc101735170)

[Специфика современных версий Windows для рабочих станций 4](#_Toc101735171)

[Дисковая конфигурация 6](#_Toc101735172)

[Процесс загрузки MBR 6](#_Toc101735173)

[Процесс загрузки GPT 7](#_Toc101735174)

[Различия между структурами разделов GPT и MBR 7](#_Toc101735175)

[Варианты загрузки: Установка и развертывание. Активация и обновления. 8](#_Toc101735176)

[2. Процесс загрузки Windows 10](#_Toc101735177)

[Этапы загрузки OC 10](#_Toc101735178)

[Boot manager и BCD 11](#_Toc101735179)

[Загрузка с протоколированием журнала загрузки драйверов 12](#_Toc101735180)

[3. Стартовое меню 13](#_Toc101735181)

[Режимы запуска ОС 13](#_Toc101735182)

[Сравнение возможностей ОС при нормальной загрузке, в безопасном режиме и режиме командной строки. 13](#_Toc101735183)

[Загружаемые драйверы 14](#_Toc101735184)

[Загружаемые службы 15](#_Toc101735185)

[Загруженные процессы 17](#_Toc101735186)

[Вывод по возможностям режимов 17](#_Toc101735187)

[Утилита msconfig 18](#_Toc101735188)

[Описание возможностей программы 18](#_Toc101735189)

[Пути в реестре 18](#_Toc101735190)

[4. Архитектура Windows. Состав и функции основных компонентов. 20](#_Toc101735191)

[Основные системные каталоги 20](#_Toc101735192)

[Типы файлов 21](#_Toc101735193)

[Основные типы файлов 21](#_Toc101735194)

[Таблица популярных прикладных форматов 21](#_Toc101735195)

[5. Установка, выполнение и удаление приложение 22](#_Toc101735196)

[DOS – приложения 22](#_Toc101735197)

[Win16-приложения 23](#_Toc101735198)

[Win32 приложения 23](#_Toc101735199)

[Варианты установки 23](#_Toc101735200)

[Этапы установки с помощью инсталлятора 24](#_Toc101735201)

[Типы инсталляторов 24](#_Toc101735202)

[Установка приложения 24](#_Toc101735203)

[Изменение в реестре 25](#_Toc101735204)

[Изменения в файловой системе 25](#_Toc101735205)

[Удаление приложения 26](#_Toc101735206)

[6. Графический интерфейс 26](#_Toc101735207)

[Рабочий стол 27](#_Toc101735208)

[Главное меню 27](#_Toc101735209)

[Панель задач 28](#_Toc101735210)

[Контекстное меню 28](#_Toc101735211)

[Ассоциация файлов 29](#_Toc101735212)

[7. Базовые свойства файловой структуры NTFS 29](#_Toc101735213)

[Структура файловой системы NTFS 29](#_Toc101735214)

[Просмотр свойства файлов и каталогов через контекстное меню 32](#_Toc101735215)

[Операции с томом, каталогами и файлами 33](#_Toc101735216)

[Получение информации о файловой системе тома 33](#_Toc101735217)

[Монтирование тома к точке соединения NTFS. 34](#_Toc101735218)

[Создание hardlink, symlink, junction point 35](#_Toc101735219)

[Использование команды fsutil 36](#_Toc101735220)

[Примеры использования chkdsk, diskpart, vssadmin 38](#_Toc101735221)

[8. Командный язык и работа в режиме командной строки 39](#_Toc101735222)

[Основные команды работы с каталогами 39](#_Toc101735223)

[Команды работы с файлами 39](#_Toc101735224)

[Команды системного назначения 40](#_Toc101735225)

[Перенаправление ввода-вывода 41](#_Toc101735226)

[Конвейеры команд и фильтры 42](#_Toc101735227)

[Пример командного файла 43](#_Toc101735228)

[9. Механизмы обмена данными между приложениями 44](#_Toc101735229)

[Clipboard 44](#_Toc101735230)

[DDE – технология (Dynamic Data Exchange) 44](#_Toc101735231)

[OLE (Object Linking and Embedding) 44](#_Toc101735232)

[Связывание и встраивание объекта в документ 45](#_Toc101735233)

[Вывод по сравнению встраивания и связывания 47](#_Toc101735234)

[Вывод 48](#_Toc101735235)

# Цель работы

Изучить возможности, устройство и функционирование современных операционных систем на примере ОС Windows 7.

# Ход работы

## Подготовка к установке ОС Windows

### Специфика современных версий Windows для рабочих станций

Установка ОС начинается с выбора дистрибутива. Операционная система Windows 7 имеет несколько редакций, ориентированных на выполнение разных задач. Часть версий предназначены для установки на стационарные компьютеры, ноутбуки и нетбуки производителями устройств, ПК продается вместе с установленной операционной системой. Другие редакции доступны для розничной продажи пользователям, которые самостоятельно устанавливают систему на свой компьютер.

Windows 7 выпущена в составе 6 редакций, отличающихся друг от друга по своему функционалу:

* Windows 7 Starter — Windows 7 Начальная;
* Windows 7 Home Basic — Windows 7 Домашняя базовая;
* Windows 7 Home Premium — Windows 7 Домашняя расширенная;
* Windows 7 Professional — Windows 7 Профессиональная;
* Windows 7 Enterprise — Windows 7 Корпоративная;
* Windows 7 Ultimate — Windows 7 Максимальная.

Редакция Windows 7 Starter предназначена для установки на нетбуки. Из операционной системы убрали довольно много функционала для того, чтобы облегчить эксплуатацию данной редакции на маломощных компьютерах. Начальная версия не предназначена для продажи в розничной сети. Операционную систему устанавливают на компьютер сборщики или производители устройств.

Windows 7 Home Basic предназначена для домашних пользователей с базовым набором функций. В данной версии Windows отсутствуют многие дополнительные мультимедийные возможности и инструменты для профессиональной работы. Домашняя базовая редакция распространятся в розничной продаже.

Версия Windows 7 Home Premium предназначена для домашних пользователей. В отличие от Домашней базовой версии, в Домашней расширенной редакции имеются Windows Media Center, Windows DVD Maker для записи дисков, полная поддержка Windows Aero, другие полезные функции.

Windows 7 Professional предназначена для использования на предприятиях среднего и малого бизнеса. Данная редакция продается в розничной сети для обычных пользователей. В этой версии операционной системы имеется необходимый функционал для работы бизнес-приложений, печать по сети, удаленное управление, шифрование файловой системы и т. п.

Windows 7 Enterprise выпущена для корпоративных клиентов, для крупных компаний. Эта версия операционной системы распространяется только по корпоративной лицензии, не продается в розницу. Большинство функций корпоративной редакции не понадобятся домашним пользователям, потому что они, в основном, применяются в бизнес-процессах.

Версия Windows 7 Ultimate имеет практически все возможности всех остальных версий операционной системы. Эта версия Windows 7 подойдет для использования на производстве или дома, теми пользователями, которые не терпят никаких компромиссов в плане возможностей операционной системы своего компьютера.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Редакции** | **Windows 7 Начальная** | **Windows 7 Домашняя базовая** | **Windows 7 Домашняя расширенная** | **Windows 7 Профессиональная** | **Windows 7 Максимальная** | **Windows 7 Корпоративная** |
| Способ распространения | OEM-лицензии | Розница и OEM-лицензии | Розница и OEM-лицензии | Розница, OEM-лицензии, корпоративные лицензии | Розница и OEM-лицензии | Корпоративные лицензии |
| Максимальный объем физической памяти (32-бит) | 2ГБ | 4ГБ | 4ГБ | 4ГБ | 4ГБ | 4ГБ |
| Максимальный объем физической памяти (64-бит) | Нет | 8ГБ | 16ГБ | 192ГБ | 192ГБ | 192ГБ |
| Центр восстановления Windows | Нет поддержки домена | Нет поддержки домена | Нет поддержки домена | Да | Да | Да |
| Домашняя группа | Только присоединение | Только присоединение | Да | Да | Да | Да |
| Удаленный рабочий стол | Только клиент | Только клиент | Только клиент | Да | Да | Да |
| Несколько мониторов | Нет | Да | Да | Да | Да | Да |
| Быстрое переключение пользователей | Нет | Да | Да | Да | Да | Да |
| Windows Aero | Нет | Базовая тема оформления | Да | Да | Да | Да |
| Диспетчер окон рабочего стола | Нет | Да | Да | Да | Да | Да |
| Windows Media Center | Нет | Нет | Да | Да | Да | Да |
| Windows DVD Maker | Нет | Нет | Да | Да | Да | Да |
| Родительский контроль | Да | Да | Да | Да | Да | Да |
| Windows Defender | Да | Да | Да | Да | Да | Да |
| Windows Firewall | Да | Да | Да | Да | Да | Да |
| EFS (Шифрованная файловая система) | Нет | Нет | Нет | Да | Да | Да |
| Печать с учетом сетевого расположения | Нет | Нет | Нет | Да | Да | Да |
| Windows XP Mode | Нет | Нет | Нет | Да | Да | Да |
| Подсистема для UNIX-приложений | Нет | Нет | Нет | Нет | Да | Да |
| Пакет мульти язычного пользовательского интерфейса | Нет | Нет | Нет | Нет | Да | Да |
| Создание и подключение VHD | Да | Да | Да | Да | Да | Да |
| Загрузка с VHD | Нет | Нет | Нет | Нет | Да | Да |
| Запуск системных оснасток | Нет | Нет | Нет | Да | Да | Да |

***Выбор редакции.***Для домашних компьютеров наиболее предпочтительным будет выбор Home Premium, если пользователь является нетребовательным, то достаточным решением будет Home Basic. В рамках лабораторной работы используем редакцию Home Premium.

### Дисковая конфигурация

При включении компьютера начинается процесс, который в итоге приведёт к загрузке операционной системы в память. Первая команда зависит от того, какова структура разделов на жёстком диске.

Есть два вида структур разделов: MBR (Master Boot Record) и GPT (GUID (Globally Unique Identifier) Partition Table). Структура разделов на диске определяет три вещи:

* Структура данных на диске.
* Код, который используется при загрузке, если раздел загрузочный.
* Где начинается и заканчивается раздел.

#### Процесс загрузки MBR

Если в системе используется структура разделов MBR, то первый процесс выполнения загрузит BIOS. Базовая структура ввода-вывода (Basic Input/Output System) включает в себя микропрограмму загрузчика. Микропрограмма загрузчика содержит низкоуровневые функции, такие как ввод с клавиатуры, доступ к видеодисплею, осуществление дисковых операций ввода-вывода и код для загрузки начальной стадии загрузчика. До того, как BIOS может определить загрузочное устройство, он выполняет последовательность функций системной конфигурации, начиная со следующих:

* Самотестирование при включении питания.
* Обнаружение и инициализация видеокарты.
* Отображение стартового экрана BIOS.
* Осуществление быстрой проверки памяти (RAM).
* Конфигурация устройств plug and play.
* Определение загрузочного устройства.

Как только BIOS определил загрузочное устройство, он считывает первый дисковый сектор этого устройства в память. Первый сектор диска — это главная загрузочная запись (MBR) размером 512 байт. В этот размер поместились три объекта:

* Первая стадия загрузчика (446 байт).
* Таблица разделов диска (16 байт на раздел × 4 раздела) — MBR поддерживает только четыре раздела, подробнее об этом ниже.
* Подпись (2 байта).

На этом этапе MBR сканирует таблицу разделов и загружает в оперативную память загрузочный сектор — Volume Boot Record (VBR).

VBR обычно содержит начальный загрузчик программ — Initial Program Loader (IPL), этот код инициирует процесс загрузки. Начальный загрузчик программ включает в себя вторую стадию загрузчика, который затем загружает операционную систему. На системах семейства Windows NT, таких как Windows XP, начальный загрузчик программ сначала загружает другую программу под названием NT Loader (аббревиатура NTLDR), которая затем загружает операционную систему.

#### Процесс загрузки GPT

GPT использует UEFI, в котором нет такой как у MBR процедуры хранения в загрузочном секторе первой стадии загрузчика с последующим вызовом второй стадии загрузчика. UEFI — унифицированный расширяемый интерфейс прошивки (Unified Extensible Firmware Interface) — является более продвинутым интерфейсом, чем BIOS. Он может анализировать файловую систему и даже сам загружать файлы.

После включения компьютера UEFI сначала выполняет функции системной конфигурации, также, как и BIOS. Это управление энергопотреблением, установка дат и других компонентов управления системой.

Затем UEFI считывает GPT — таблицу разделов GUID. GPT располагается в первых секторах диска, сразу после сектора 0, где хранится главная загрузочная запись для Legacy BIOS.

GPT определяет таблицу разделов на диске, на которой загрузчик EFI распознает системный раздел EFI. Системный раздел содержит загрузчики для всех операционных систем, установленных на других разделах жёсткого диска. Загрузчик инициализирует менеджер загрузки Windows, который затем загружает операционную систему.

#### Различия между структурами разделов GPT и MBR

GPT — более новая и продвинутая структура разделов. MBR используется давно, она стабильная и обладает максимальной совместимостью. Хотя GPT со временем может вытеснить MBR, поскольку предлагает более продвинутые функции, но в некоторых случаях можно использовать только MBR.

Master Boot Record

MBR — традиционная структура для управления разделами диска. Поскольку она совместима с большинством систем, то по-прежнему широко используется. Главная загрузочная запись расположена в первом секторе жёсткого диска. Она содержит таблицу разделов — информацию об организации логических разделов на жёстком диске.

MBR также содержит исполняемый код, который сканирует разделы на предмет активной ОС и инициализирует процедуру загрузки ОС.

Преимущества:

* Совместима с большинством систем.

Недостатки:

* Допускает только четыре раздела, с возможностью создания дополнительных подразделов на одном из основных разделов.
* Ограничивает размер раздела двумя терабайтами.
* Информация о разделе хранится только в одном месте — в главной загрузочной записи. Если она повреждена, то весь диск становится нечитаемым.

Таблица разделов GUID (GPT)

GPT — более новый стандарт для определения структуры разделов на диске. Для определения структуры используются глобальные уникальные идентификаторы (GUID).

Преимущества:

* Допускает неограниченное количество разделов. Лимит устанавливает операционная система.
* Не ограничивает размер раздела.
* GPT хранит копию раздела и загрузочных данных и может восстановить данные в случае повреждения основного заголовка GPT.
* GPT хранит значения контрольной суммы по алгоритму циклического избыточного кода (CRC) для проверки целостности своих данных (используется для проверки целостности данных заголовка GPT). В случае повреждения GPT может заметить проблему и попытаться восстановить повреждённые данные из другого места на диске.

Недостатки:

* Может быть несовместима со старыми системами.

***Выбор файловой системы.*** В рамках лабораторной работы удобнее использовать структуру MBR.

### Варианты загрузки: Установка и развертывание. Активация и обновления.

Варианты установки:

* **Установка с использованием внешнего носителя** - в BIOS выбирается приоритетный носитель, содержащий образ ОС, с которого будет выполняться установка.
* **Обновление или установка новой ОС из текущей операционной системы** – установка с помощью специальных программ, предназначенных для обновления ОС. Если платформа ОС не меняется, чаще всего возможно выполнить обновление. При этом настройки конфигурации системы, приложения и данные сохраняются. Файлы старой ОС просто заменяются файлами новой.

Развертывание операционной системы позволяет создать образы операционной системы и развернуть эти образы на целевых компьютерах. В ходе развертывания операционной системы возникают последовательности задач, облегчающие развертывание образов операционной системы и других пакетов программного обеспечения Configuration Manager.

Варианты развёртывания:

* **PXE инициировал развертывание**: развертывание, инициированное PXE, позволит клиентским компьютерам запрашивать развертывание по сети. В этом методе развертывания изображение операционной системы и Windows загрузки PE отправляются в точку рассылки, настроенную для получения запросов загрузки PXE.

**PXE (Preboot eXecution Environment)** — среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных (жёсткого диска, USB-накопителя и т. п.).

* **Сделать операционные системы доступными в Центре программного обеспечения**. Вы можете развернуть операционную систему и сделать ее доступной в Центре программного обеспечения. Клиенты Configuration Manager могут инициировать установку операционной системы из Центра программного обеспечения.
* **Многоуровневые развертывания**: многоуровневые развертывания сохраняют пропускную способность сети, одновременно отправляя данные нескольким клиентам вместо отправки копии данных каждому клиенту по отдельному подключению. В этом методе развертывания изображение операционной системы отправляется в точку распространения. Это, в свою очередь, развертывает изображение, когда клиентские компьютеры запрашивают развертывание.
* **Развертывание мультимедиа для загрузки**. Развертывание мультимедиа для загрузки позволяет развертывать операционную систему после начала работы компьютера назначения. Когда компьютер назначения запускается, он извлекает последовательность задач, изображение операционной системы и любой другой необходимый контент из сети. Так как этот контент не включен в мультимедиа, его можно обновить без повторного создания мультимедиа.

***Выбор варианта загрузки.*** Загрузка с помощью установки является более простым способом для загрузки на один компьютер. Поскольку операционная система будет установлена в виртуальной машине, то установим с помощью приоритетного внешнего носителя, выбранного в BIOS.

***Активация и обновления.*** Для выполнения лабораторной работы будет достаточно времени пробного периода, который предоставляется не активированной ОС. Обновления ОС будут отключены т.к. они не требуются.

## Процесс загрузки Windows

### Этапы загрузки OC

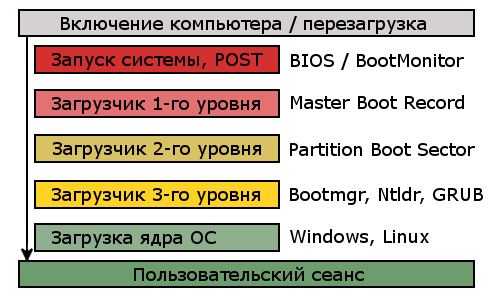


Схема процесса загрузки ОС

* **Включение компьютера, POST, BootMonitor.** Начальный этап загрузки операционной системы после включения компьютера начинается в BIOS (Basic Input/Output System — базовая система ввода-вывода). В настройках BIOS мы указываем загрузочное устройство, или ряд загрузочных устройств в порядке их приоритета. Возможны различные варианты загрузки и их комбинации: с жесткого диска, CD/DVD – диска, USB-flash и другие.

Сразу после прохождения POST (Power-On Self-Test — самотестирование после включения) BIOS компьютера начнет поочередно перебирать указанные загрузочные устройства до тех пор, пока на одном из них не найдет подходящую специальную запись, в которой содержится информация о дальнейших действиях.

* **Загрузчик 1-го уровня. Master Boot Record.** Master Boot Record — главная загрузочная запись, расположена в первых физических секторах загрузочных устройств хранения. Она содержит таблицу разделов (Partition Table) и исполняемый код.

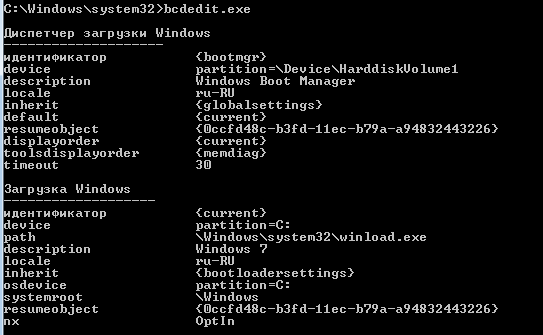
Главной задачей программы, записанной в MBR, является поиск активного системного раздела диска и передача управления его загрузочному сектору. Таким образом, эту стадию можно назвать подготовительной, в силу того, что непосредственно загрузки самой ОС еще не происходит.

Системным принято называть раздел диска (устройства хранения) на котором расположены файлы операционной системы, отвечающие за процесс загрузки ОС (сама операционная система может размещаться в другом разделе). В принципе, системных разделов может быть несколько, поэтому один из них отмечается как активный. Именно его ищет программа, загруженная с MBR.

* **Загрузчик 2-го уровня. Partition Boot Sector.** Следующим этапом загрузки компьютера является передача управления исполняемому коду, записанному в PBS (Partition Boot Sector — загрузочный сектор активного раздела). PBS расположен в первом секторе (секторах) соответствующего раздела диска. В коде PBS прописано имя файла загрузчика операционной системы, которому и передается управление на этом этапе.
* **Начальный этап загрузки операционной системы. Менеджер загрузки ОС.** В версиях Windows до Vista, например, Windows XP будет загружен Ntldr. Он, в свою очередь, считывает информацию из текстового файла Boot.ini, в котором записана информация об установленных операционных системах.
* **Загрузка ядра операционной системы.** Завершающим этапом загрузки операционной системы является загрузка ядра ОС и передача ему управления.

### Boot manager и BCD

В Windows Vista или Windows 7 / 8 /10 диспетчер загрузки называется Bootmgr. Он читает список установленных операционных систем не из текстового файла, boot.ini, а из системного хранилища BCD (Boot Configuration Data), доступ к которому осуществляется посредством специальной утилиты bcdedit.exe.



Использование утилиты bcdedit.exe

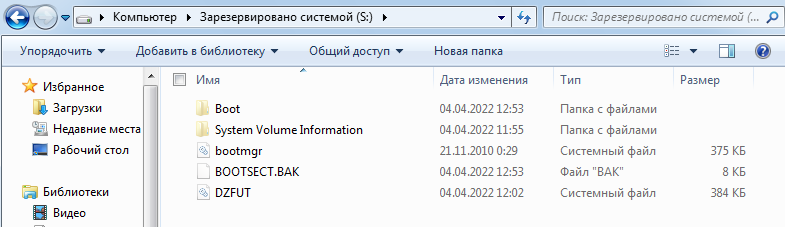
Процесс стандартной инсталляции операционной системы Windows Vista / 7 / 8 создает в начале диска дополнительный раздел «Зарезервировано системой». Ему не присваивается буква диска, благодаря чему, если не сделать дополнительных шагов, и сам раздел и его содержимое будут скрыты от пользователя. Этот основной раздел имеет статус «Системный» и «Активный» и, следовательно, удалить его тоже нельзя. В Windows 7 размер такого раздела составляет 100 МБ, из которых занято около 30 МБ.

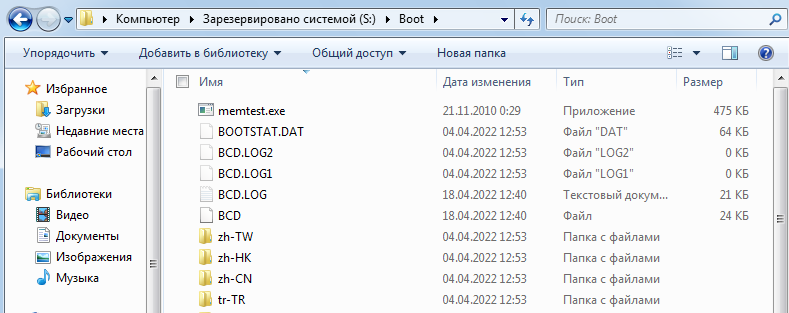
Откроем консоль «Управление компьютером» и перейдём в меню «Управление дисками» и убедимся в наличии раздела.



Меню «Управление дисками»

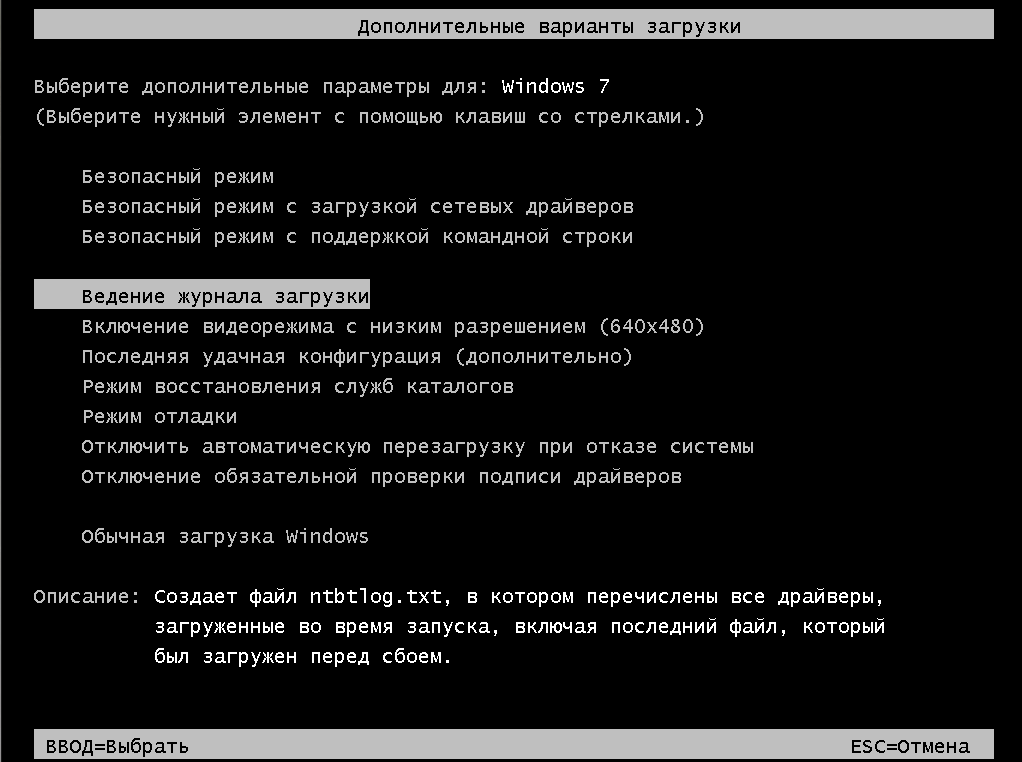
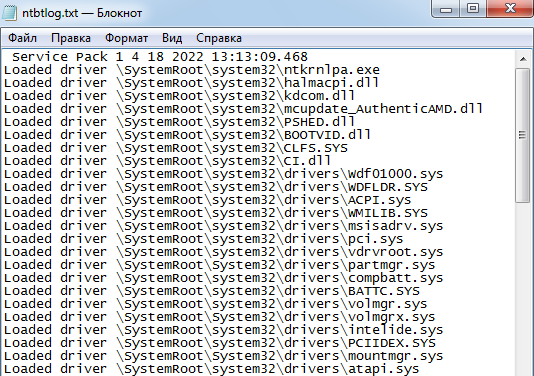
Посмотрим содержимое раздела «Зарезервировано системой». Для этого назначим ему букву диска в меню «Управление дисками». В панели управления компьютером включим отображение скрытых папок и файлов.

В корне раздела находится файл bootmgr, который является загрузчиком.

В каталоге Boot находится база хранилища конфигураций загрузки BCD, «логи» и папки с языковыми файлами.

Раздел диска, который содержит загрузчик, называется системным диском, а раздел, содержащий основные файлы операционной системы – загрузочным. Системный и загрузочный диски могут находится на одном или разных физических дисках.

### Загрузка с протоколированием журнала загрузки драйверов

Для перехода к вариантам загрузки необходимо нажать клавишу F8 до загрузки в операционную систему. Выберем пункт «Ведение журнала загрузки». При загрузке в данном режиме будет создан файл C:\Windows\ntbtlog.txt с описанием драйверов, загруженных во время запуска.

Меню boot manager Содержимое файла ntbtlog.txt

## Стартовое меню

### Режимы запуска ОС

С помощью boot manager можно произвести загрузку в разных режимах:

* **Безопасный режим** - это способ загрузки операционной системы, предусматривающий работу в среде Windows только с самыми необходимыми драйверами. Поскольку никакие дополнительные программы не загружаются, то диагностика неполадок программ и драйверов значительно упрощается. Безопасный режим предназначен для исправления некоторых проблем в операционной системе. Он также широко используется для удаления вирусов.
* **Безопасный режим с поддержкой сетевых драйверов** –  запуск Windows в безопасном режиме и загрузка сетевых драйверов и служб, необходимых для доступа к Интернету или к другим компьютерам в локальной сети. кроме основных драйверов загружаются драйверы сети.
* **Безопасный режим с поддержкой командной строки** — запуск Windows в безопасном режиме с окном командной строки вместо обычного интерфейса Windows. Этот вариант предназначен для ИТ-специалистов и администраторов.
* **Ведение журнала загрузки** – запуск Windows, при котором в файл ntbtlog.txt ведётся протоколирование загрузки драйверов. Эта информация может оказаться полезной для расширенной диагностики.
* **Включение видеорежима с низким разрешением (640\*480)** - запуск Windows с использованием встроенного видеодрайвера от Microsoft. Устанавливаются низкие значения разрешения и частоты обновления изображения. Этот режим можно использовать для сброса параметров дисплея.
* **Последняя удачная конфигурация (дополнительно)** - запуск Windows с использованием последней успешно работавшей конфигурации реестра и драйверов. Этот режим используется, если система отказывается запускаться и безопасный режим не помогает. При старте этого режима подгружаются резервные копии рабочих драйверов и некоторых системных файлов, что особенно удобно в случае, если причиной сбоя является изменение драйверов или повреждение файлов системы.
* **Режим восстановления служб каталогов** - запуск контроллера домена Windows, на котором выполняется Active Directory, для восстановления службы каталогов.
* **Режим отладки** - запуск Windows в режиме расширенной диагностики. Отладочная информация может быть послана по последовательному кабелю на другой компьютер с запущенным отладчиком.
* **Отключить автоматическую перезагрузку при отказе системы** - запрет автоматического перезапуска Windows в случае сбоя в работе системы. Этот вариант следует выбирать только в том случае, если Windows вошла в цикл, когда сбой в работе системы вызывает перезапуск, и система продолжает перезапускаться снова и снова.
* **Отключить обязательную проверку подписи драйверов**- позволяет устанавливать драйверы, содержащие неправильные подписи.
* **Обычная загрузка Windows**- запуск Windows в обычном режиме.

### Сравнение возможностей ОС при нормальной загрузке, в безопасном режиме и режиме командной строки.

Для сравнения возможностей ОС, необходимо узнать какие драйверы будут загружены при запуске, какие службы и процессы будут активны. Состояние загрузки драйверов будем наблюдать в файле ntbtlog. Процессы и службы в графическом режиме можно отобразить с помощью диспетчера задач, в режиме командной строки активные процессы можно узнать с помощью команды tasklist, службы с помощью команды sc.

#### Загружаемые драйверы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Драйвер** | **Описание драйвера** | **Нормальный режим** | **Безопасный режим** | **Командная строка** |
| ntkrnlpa.exe | системный файл ядра Windows | + | + | + |
| halmacpi.dll | Hardware Abstraction Layer – уровень абстракции оборудования | + | + | + |
| kdcom.dll | Kernel-Mode driver –драйвер режима ядра | + | + | + |
| PSHED.dll | platform-specific hardware error driver -драйвер аппаратных ошибок | + | + | + |
| BOOTVID.dll | VGA Boot Driver - Загрузочный драйвер VGA | + | + | + |
| CI.dll | Code Integrity – целостность кода ОС | + | + | + |
| ACPI.sys | Поддержка управления питанием и перечисление устройств Plug and Play | + | + | + |
| pci.sys | Функциональный драйвер для шины PCI | + | + | + |
| vdrvroot.sys | Драйвер виртуальных дисков | + | + | + |
| partmgr.sys | Драйвер разделов диска | + | + | + |
| compbatt.sys | Composite Battery Driver – драйвер батареи | + | + | + |
| volmgr.sys | Volume Manager Driver - Драйвер диспетчера томов | + | + | + |
| intelide.sys | Intel PCI IDE Driver – драйвер IDE устройств | + | + | + |
| mountmgr.sys | Mount Point Manager (Менеджер монтирования) – упрвление именами томов | + | + | + |
| msahci.sys | Microsoft AHCI Driver – подключение накопителей информации стандарта SATA | + | + | + |
| fltmgr.sys | Microsoft Filesystem Filter Manager | + | + | + |
| Ntfs.sys | Драйвер файловой структуры NTFS | + | + | + |
| ksecdd.sys | Kernel Security Support Provider Interface - поддержка удаленного прямого доступа к памяти в режиме ядра в сетевом адаптере | + | + | + |
| NETIO.SYS | Network I/O Subsystem – подсистема сетевого ввода вывода | + | + | + |
| tcpip.sys | Драйвер стека протоколов TCP/IP | + | + | + |
| hwpolicy.sys | Hardware Policy Driver – политика загрузки драйверов | + | + | + |
| usbhub.sys | Драйвер usb устройств | + | + | + |
| vga.sys | Видеодрайвер VGA | + | + | + |
| cdrom.sys | Драйвер дисковода | + | + | + |
| tdx.SYS | TDI Translation Driver- общий интерфейс для драйверов, предназначенный для общения с различными сетевыми транспортными протоколами. | + | - | - |
| AFD.SYS | Ancillary Function Driver for WinSock - драйвер сокета ОС. | + | - | - |
| NetBIOS.SYS | NetBIOS interface driver – драйвер службы NetBIOS | + | - | - |
| TermDD.SYS | Terminal Desktop Server Driver –драйвер терминального сервера | + | - | - |
| rdbss.SYS | Redirected Drive Buffering SubSystem Driver – подсистема буферизации перенаправленного диска | + | - | - |
| nsiproxy.SYS | NSI Proxy Service Driver – драйвер прокси сервера | + | - | - |
| discache.SYS | Cache Driver | + | - | - |
| CSC.SYS | Windows Client Side Caching Driver | + | - | - |

При запуске ОС первыми запускаются файлы ядра системы и файл уровня абстракции оборудования. Драйвера, запускаемые в безопасном режиме и режиме командной строки:

* драйвер PCI шины;
* драйвер контролеров SATA для подключения жестких дисков
* драйвер дисководов компакт-дисков (CD, DVD);
* драйвер контроллеров USB;
* видеодрайвер VGA.

Это минимальный набор драйверов, который необходим для запуска Windows.

При запуске в нормальном режиме дополнительно запускаются драйвера, связанные с сетевыми функциями.

#### Загружаемые службы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Служба** | **Описание службы** | **Нормальный режим** | **Безопасный режим** | **Командная строка** |
| CryptSvc | Службы криптографии | + | + | + |
| DcomLaunch | Модуль запуска процессов DCOM-сервера | + | + | + |
| eventlog | Журнал событий Windows | + | + | + |
| PlugPlay | Plug-and-Play | + | + | + |
| Power | Питание | + | + | + |
| ProfSvc | Служба профилей пользователей | + | + | + |
| RpcSs | Удаленный вызов процедур (RPC) | + | + | + |
| RpcEptMapper | Сопоставитель конечных точек RPC | + | + | + |
| WinDefend | Защитник Windows | + | + | + |
| wscsvc | Центр обеспечения безопасности | + | + | - |
| wuauserv | Центр обновления Windows | + | + | - |
| AeLookupSvc | Информация о совместимости приложений | + | - | - |
| Appinfo | Сведения о приложении | + | - | - |
| AudioEndpointBuilder | Средство построения конечных точек Windows Audio | + | - | - |
| Audiosrv | Windows Audio | + | - | - |
| BFE | Служба базовой фильтрации | + | - | - |
| BITS | Фоновая интеллектуальная служба передачи | + | - | - |
| Browser | Браузер компьютеров | + | - | - |
| CscService | Автономные файлы | + | - | - |
| Dhcp | DHCP-клиент | + | - | - |
| DPS | Служба политики диагностики | + | - | - |
| EventSystem | Система событий COM+ | + | - | - |
| fdPHost | Хост поставщика функции обнаружения | + | - | - |
| FDResPub | Публикация ресурсов обнаружения функции | + | - | - |
| FontCache | Служба кэша шрифтов Windows | + | - | - |
| gpsvc | Клиент групповой политики | + | - | - |
| HomeGroupProvider | Поставщик домашней группы | + | - | - |
| iphlpsvc | Вспомогательная служба IP | + | - | - |
| lmhosts | Модуль поддержки NetBIOS через TCP/IP | + | - | - |
| MMCSS | Планировщик классов мультимедиа | + | - | - |
| MpsSvc | Брандмауэр Windows | + | - | - |
| Netman | Сетевые подключения | + | - | - |
| netprofm | Служба списка сетей | + | - | - |
| NlaSvc | Служба сведений о подключенных сетях | + | - | - |
| nsi | Служба интерфейса сохранения сети | + | - | - |
| p2pimsvc | Диспетчер удостоверения сетевых участников | + | - | - |
| PNRPsvc | Протокол PNRP | + | - | - |
| SamSs | Диспетчер учетных записей безопасности | + | - | - |
| Schedule | Планировщик заданий | + | - | - |
| SENS | Служба уведомления о системных событиях | + |  |  |
| SSDPSRV | Обнаружение SSDP | + | - | - |
| upnphost | Узел универсальных PNP-устройств | + | - | - |
| UxSms | Диспетчер сеансов диспетчера окон рабочего стола | + | - | - |
| WdiServiceHost | Узел службы диагностики | + | - | - |

Локальные службы, запускаемые в безопасном режиме:

* Журнал событий Windows- позволяет просматривать и записывать события операционной системы. В безопасном режиме можно просмотреть события, которые привели к возникновению ошибок и сбоев Windows;
* Поддержка самонастраивающихся устройств – позволяет использовать устройства, подключенные к компьютеру, которые не требуют установки драйверов;
* Удаленный вызов процедур (RPC) – Обеспечивает правильную работу COM приложений;
* Службы криптографии – служба проверки подписи файлов и приложений;
* Защитник Windows – служба защиты операционной системы;
* сервисы Windows для настройки операционной системы.

Сетевые сервисы и службы, которые остановлены в безопасном режиме и могут быть активированы:

* Поддержка сетевых адаптеров и устройств, необходимых для проводного и беспроводного подключения к локальной сети;
* DHCP-клиент – служба, обеспечивающая получение и изменение IP-адреса, а также обновления DNS;
* DNS-клиент – служба, необходимая для регистрации имени компьютера в сети;
* Сетевые подключения – служба, необходимая для работы сервисов удаленный доступ и настройки локальной сети и сетевых подключений;
* Модуль поддержки NetBIOS через TCP/IP – Служба, позволяющая организовать общий доступ к папкам и принтерам компьютера, находящегося в локальной сети;
* Брандмауэр Windows – сетевой экран Windows, обеспечивающий безопасность и защиту от проникновения на компьютер с локальной сети.

При запуске в нормальном режиме дополнительно активируются службы, позволяющие полноценно использовать ОС, например, включается служба, отвечающая за звук.

#### Загруженные процессы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Процесс** | **Описание процесса** | **Нормальный режим** | **Безопасный режим** | **Командная строка** |
| smss.exe | Диспетчер сеанса Windows | + | + | + |
| csrss.exe | Процесс исполнения клиент-сервер | + | + | + |
| winlogon.exe | Программа входа в систему Windows | + | + | + |
| service.exe | Приложение служб и контроллеров | + | + | + |
| lsass.exe | Local Security Authority Process | + | + | + |
| winint.exe | Автозагрузка приложений Windows | + | + | - |
| scvhost.exe | Хост-процесс для служб Windows | + | + | - |
| explorer.exe | Проводник | + | + | - |
| dwm.exe | Диспетчер окон рабочего стола | + | - | - |

Основные для ОС процессы запущены во всех трёх режимах. В режиме командной строки нет процессов, связанных с графической оболочкой и автозагрузкой приложений. В безопасном режиме не активен процесс диспетчера окон рабочего стола.

#### Вывод по возможностям режимов

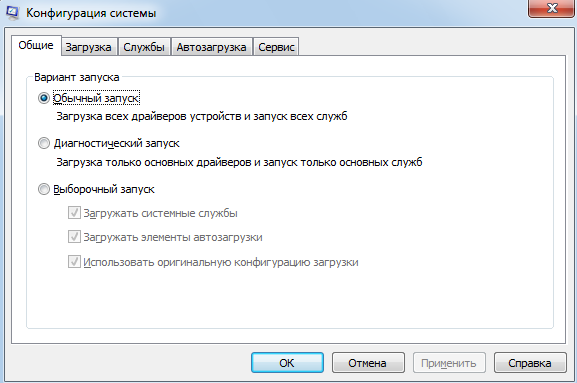
Безопасный режим запускает компьютер с минимально необходимым набором драйверов и служб. Все сторонние программы отключаются, и даже системные инструменты ограничены только необходимым набором функций.

Безопасный режим – хороший способ избавиться от вредоносных программ, блокирующих загрузку или работу компьютера. Он также обеспечивает среду, в которой можно «откатить» обновления системы или драйверов, очистить автозагрузку, удалить недавно установленные программы или провести диагностику и устранение неполадок Windows.

Безопасный режим с поддержкой командной строки обеспечивает большее быстродействие путём отключения графического интерфейса. Если пользователь умеет обращаться с командной строкой и знает её команды, то восстановление системы лучше проводить в этом режиме.

### Утилита msconfig

Это утилита для управления авто запускаемыми программами и загрузкой Windows. Начиная с Windows версии 98, Microsoft поставляет утилиту «MSConfig.exe», предоставляющую удобный интерфейс для управления файлами, запускающимися при загрузке Windows. Она находится в каталоге установки Windows. Её можно запустить из диалогового окна «Выполнить» или через командную строку.



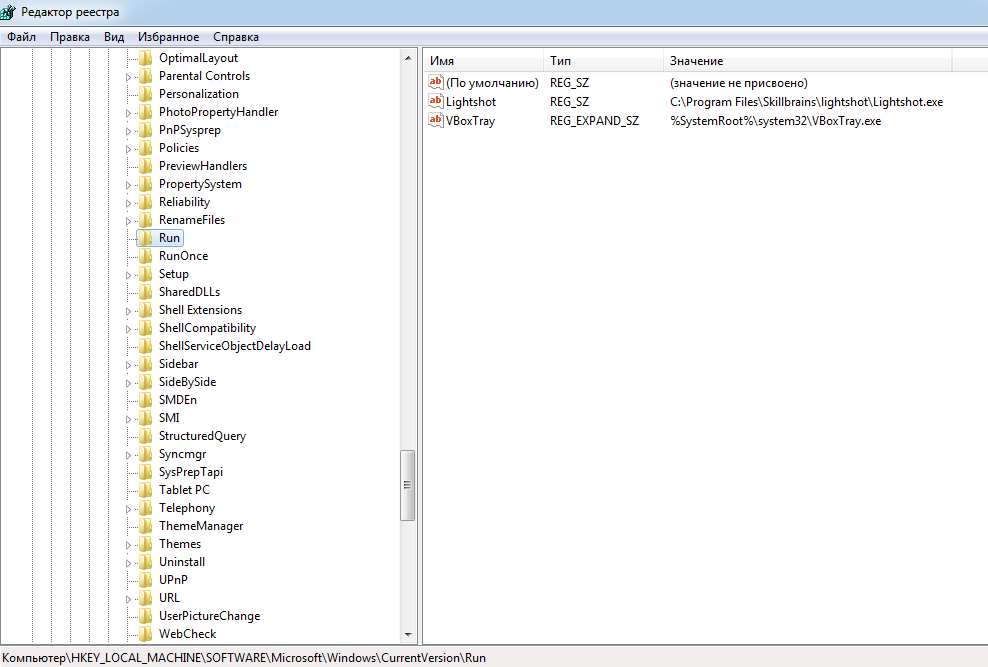
#### Описание возможностей программы

* На вкладке «Общие» есть возможно выбирать вариант загрузки: обычный, диагностический или выборочный.
* В разделе «Загрузка» можно настроить дополнительные параметры запуска ОС. Например, количество процессоров (ядер), количество памяти, включение/отключение журнала загрузки, наличие графического интерфейса.
* На вкладке «Службы» можно отключить ненужные службы, запускаемые при загрузке системы. Многие программы, такие как антивирусы и программы безопасности компьютера, запускаются через службы операционной системы. В случае отключения служб такие программы могут перестать работать.
* Вкладка «Автозагрузка» отвечает за загрузку приложений (программ) а также определённых служебных утилит, загружаемых не через службы.

#### Пути в реестре

Все настройки программы хранятся в реестре по пути HKLM\SOFTWARE\Microsoft\Shared Tools\MSConfig.

Ключи, отвечающие за автозагрузку:

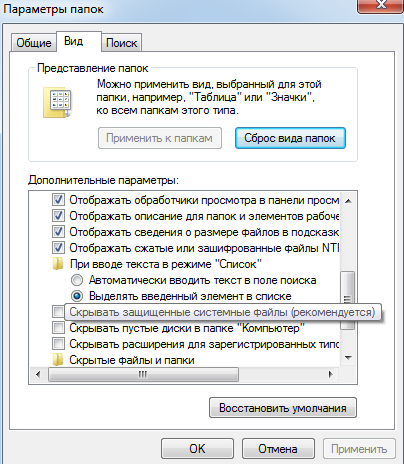


* [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] - программы, которые запускаются при входе в систему. Данный раздел отвечает за запуск программ для всех пользователей системы.
* [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce] - программы, которые запускаются только один раз при входе пользователя в систему. После этого ключи программ автоматически удаляются из данного раздела реестра. Данный раздел отвечает за запуск программ для всех пользователей системы.
* [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx] - программы, которые запускаются только один раз, когда загружается система. Этот раздел используется при инсталляции программ, например, для запуска настроечных модулей. После этого ключи программ автоматически удаляются из данного раздела реестра. Данный раздел отвечает за запуск программ для всех пользователей системы.
* [HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run] - программы, которые запускаются при входе текущего пользователя в систему
* [HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce] - программы, которые запускаются только один раз при входе текущего пользователя в систему. После этого ключи программ автоматически удаляются из данного раздела реестра.
* [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices] - программы, которые загружаются при старте системы до входа пользователя в Windows.
* [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce] - программы отсюда загружаются только один раз, когда загружается система.

## Архитектура Windows. Состав и функции основных компонентов.

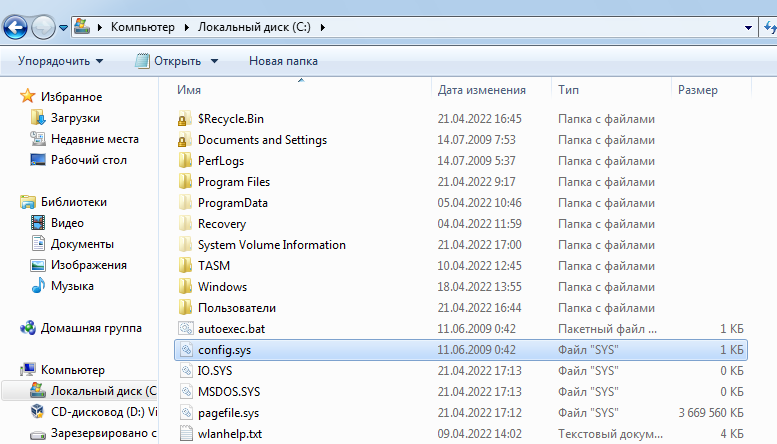
### Основные системные каталоги

Системные каталоги содержат основные системные файлы, необходимые для работы ОС. Системные файлы «Windows» – это любые файлы с включенным скрытым системным атрибутом. Их диапазон варьируется от аппаратных драйверов, конфигурационных файлов, библиотек динамической компоновки «DLL» до различных наборов файлов, составляющих реестр операционной системы «Windows». Удаление, перемещение, переименование или изменение этих файлов может привести к отказу в запуске определенных приложений, краху ее отдельных элементов, или даже к полному сбою системы. Для отображения системных файлов и каталогов необходимо в свойствах «Параметры папок» включить их отображение.



Основным каталогом является корень диска **С:\**, в нём содержатся подкаталоги с системными файлами Windows (**C:\Windows**), файлы установленных программ (**C:\Program Files** и **C:\ProgramData**), данные о профилях пользователей (**C:\Documents and Settings** и **C:\Users**), точки восстановления (**C:\Recovery)**. Также в корне загрузочного диска находятся системные файлы:

* **config.sys** — файл конфигурирования операционной системы, содержащий директивы настройки системы и команды загрузки драйверов.
* **autoexec.bat -** файл, содержащий последовательность команд, исполняется после старта командного интерпретатора, который загружается после обработки команд из файла конфигурации config.sys.
* **IO.sys -** Инициализирует драйверы для консоли, диска, последовательного порта.
* **pagefile.sys** – файл подкачки. При нехватке оперативной памяти Windows резервирует определенное место на жестком диске и использует его для увеличения своих возможностей. Иными словами, выгружает часть данных из оперативной памяти в файл pagefile.sys. Очень часто необходимые для исследователя сведения остаются только в файле подкачки.



### Типы файлов

#### Основные типы файлов

* **com** - Тип исполняемого файла, при выполнении которого данные, код и стек находятся в одном и том же 16-битном сегменте. Пример: command.com
* **exe** - Тип исполняемого файла, при выполнении которого данные, код и стек находятся в разных сегментах. Пример: cmd.exe
* **sys** - Файлы драйверов или конфигурационные файлы. Пример: pci.sys
* **bat** - Текстовый файл в, содержащий последовательность команд, предназначенных для исполнения командным интерпретатором. Пример: autoexec.bat
* **dll** - динамическая библиотека, в которой программы могут хранить функции, переменные и т. д. Пример: Hal.dll
* **reg** - Файлы данных реестра.
* **ini** -  Файл конфигурации, который содержит данные настроек. Пример: win.ini

## Установка, выполнение и удаление приложение

### DOS – приложения

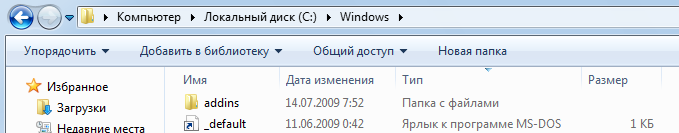
Установка DOS-программ сводится к размещению на диске, формированию переменных окружения (если для программы это необходимо). Для каждого приложения DOS ОС создает отдельную VDM. VDM работают в режиме вытесняющей многозадачности, деля процессорное время с системной виртуальной машиной. VDM не создают очередей сообщений. При выходе из программы завершается работа виртуальной машины и освобождаются ресурсы.

Загрузкой MS-DOS приложений можно управлять с помощью .pif файла, который является ярлыком на DOS программу.

В PIF-файле описываются параметры выполнения программы:

* режим выполнения (экранный или оконный)
* ресурсы памяти
* установки приоритетов для планирования процессов

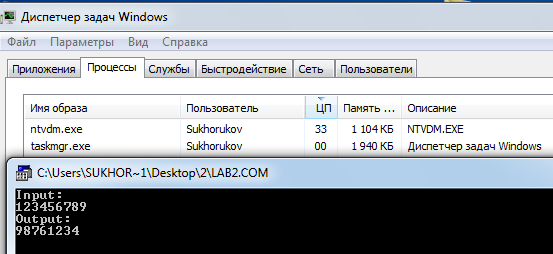
В каталоге C:\Windows есть файл **\_default.pif**, который является стандартным ярлыком для DOS программ. Если для программы не создан отдельный ярлык, то она запускается с параметрами, указанными в этом файле.



Расположение \_default.pif

Запустим DOS программу, написанную в рамках второй лабораторной курса ПиМОСПС 5 семестра. Чтобы убедиться в том, что запущена Виртуальная DOS Машина, запустим диспетчер. задач.

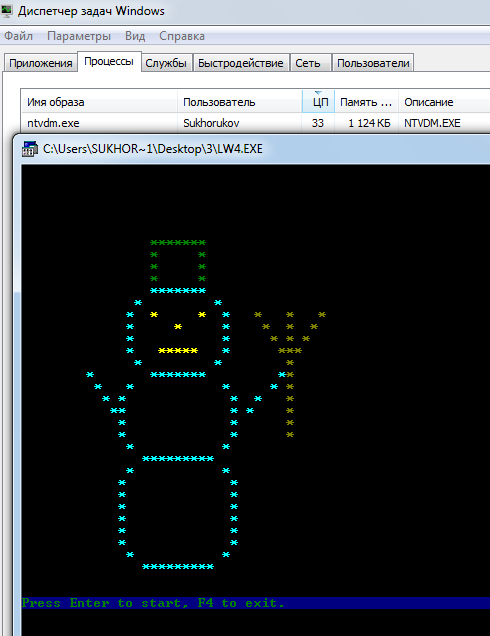
Процесс ntvdm.exe является процессом виртуальной машины.



Работа Virtual DOS Machine

### Win16-приложения

Приложения Win16 выполняются в Windows 7 в общем пространстве адресов в пределах системной виртуальной машины и имеют общую очередь сообщений. Ситуация с зависанием Win16 опасна, поскольку остальные программы Win16 перестанут получать сообщения и тоже зависнут.

Кроме того, заблокированное приложение Win16 способно повлиять и на приложение Win32, несмотря на то, что они используют независимые очереди сообщений. В случае аварийного завершения Win16 все системные ресурсы, которые были заняты этим приложением, освободятся после того, как будут выгружены все текущие программы Win16.

Работа Win16 приложения

### Win32 приложения

Для каждого приложения Win32 используется отдельная адресная область в пределах системной виртуальной машины. Приложения Win32 работают в режиме вытесняющей многозадачности, для каждого Win32-приложения и для каждого создаваемого ими потока используются отдельные очереди сообщений. Это делает ошибку в Win32-приложениях фактически безопасной для остальных приложений.

#### Варианты установки

* **Установка вручную** — установка выполняется без установщика или со значительным количеством операций, вручную выполняемых пользователем.
* **«Тихая» установка** — установка, в процессе которой не отображаются сообщения или окна.
* **Автоматическая установка** — установка, которая выполняется без вмешательства со стороны пользователя, исключая сам процесс её запуска. Процесс установки иногда требует взаимодействия с пользователем, который управляет процессом установки, делая выбор: принимая пользовательское соглашение, настраивая параметры, указывая пароли и так далее.

#### Этапы установки с помощью инсталлятора

* Анализ возможностей установки
  + - * Версия ОС – разрядность процессора
      * Состав требуемого оборудования
      * Наличие памяти на диске
* Проверка лицензионности
* Проверка целостности с помощью цифровых подписей
* Подготовка и копирование файлов
* Регистрация в реестре
* Установка в интерфейсе пользователя
* Установка параметров автозапуска
* Возможность отмены установки

#### Типы инсталляторов

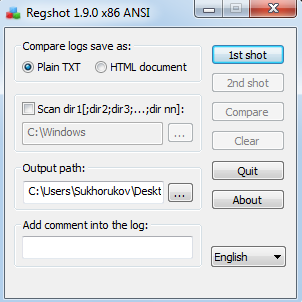
Наиболее часто используются следующие типы инсталляторов:

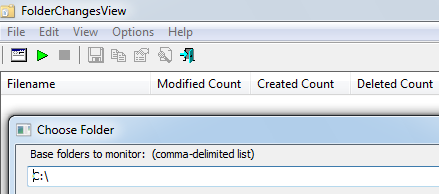
* InstallShield
* Windows Installer Service (\*.msi)
* InstallShield с MSI
* Inno Setup
* Nullsoft SuperPiMP Install System (NSIS)
* WISE Installer

### Установка приложения

Установим приложение Notepad++ (приложение для удобного редактирования исходного кода программ) и определим, какие изменения вносятся в файловую систему и реестр. Для проверки состояния реестра используем программу Regshot, файловой системы – FolderChanhesView.

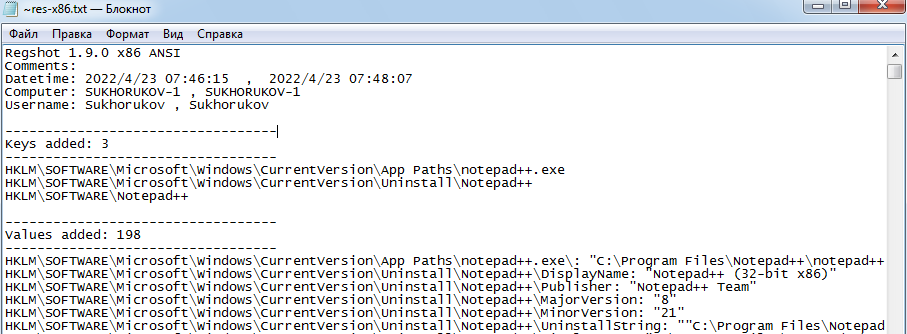
В программе Regshot сделаем снимок состояния реестра «1st shot», после установки приложения 7Zip сделаем второй снимок «2nd shot» и сравним состояния «Compare».

В программе FolderChanhesView в качестве каталога, состояние которого будет отслеживаться выберем корень диска С, после начала слежения в интерфейсе программы будут отображаться изменения.



Интерфейс программы Regshot Интерфейс программы FolderChanhesView

#### Изменение в реестре

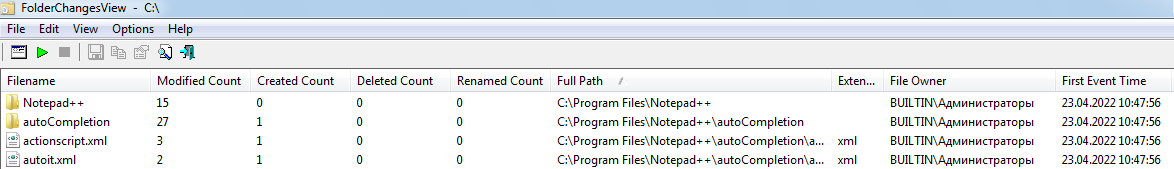
Результатом работы приложения Regshot является текстовый файл ~res-x86.txt, в котором описаны добавленные ключи и поля, измененные значения полей.

Добавлены ключи в разделы HKEY\_LOCAL\_MACHINE\SOFTWARE – раздел данных об установленных приложениях - и HKEY\_USERS\_S-1-5-21-4114… - раздел данных о пользователе.

Основными ключами являются:

* HKLM \SOFTWARE \Microsoft \Windows \CurrentVersion \App Paths \notepad++.exe\: "C:\Program Files\Notepad++\notepad++.exe" – ключ, по котором храниться путь к исполняемому файлу.
* HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Notepad++ – ключ, по которому хранится информация об удалении приложения.
* HKLM\SOFTWARE\Notepad++\: "C:\Program Files\Notepad++" - указание пути, по которому расположено приложение.
* HKU \S-1-5-21-4114700349-3829428303-2104194-1001 \Software \Microsoft \Windows \CurrentVersion \Explorer \StartPage \NewShortcuts \C: \Users \Sukhorukov \AppData \Roaming \Microsoft \Windows \Start Menu \Programs \Notepad++.lnk: 0x00000001 – добавление приложения в интерфейс пользователя (Меню «Windows» -> «Все программы» -> «7Zip»).

#### Изменения в файловой системе

Результатом работы приложения FolderChanhesView является список каталогов и файлов, которые были изменены или добавлены.

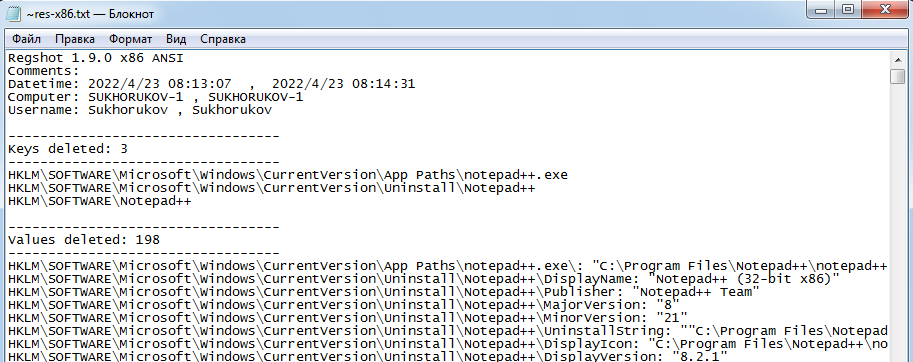
Изменения каталогов:

* C:\Program Files\Notepad++ – каталог, в котором установлено приложения, где находятся его исполняемый файл и необходимые для работы ресурсы.
* C:\ProgramData\Microsoft\Windows\Start Menu\Programs - добавление приложения в интерфейс пользователя (Меню «Windows» -> «Все программы» -> «7Zip»).
* C:\Users\Sukhorukov\AppData\Roaming – добавление пользовательских настроек для приложения. При использовании данной учётной записи пользователя на другом компьютере, настройки приложения будут сохранены.
* C:\Users\Sukhorukov\AppData\Local\ – добавление локальных пользовательских настроек приложения. Настройки сохраняются только на одном компьютере.
* C:\Users\Sukhorukov\AppData\Local\Temp – добавление временных файлов настроек приложения.
* C:\Windows\Temp – добавление временных файлов приложения.
* C:\Users\Sukhorukov\NTUSER.DAT – изменение файла реестра пользователя.
* C:\Windows\System32\config\SOFTWARE – изменение файла реестра.
* C:\Windows\System32\config\SYSTEM– изменение файла реестра.

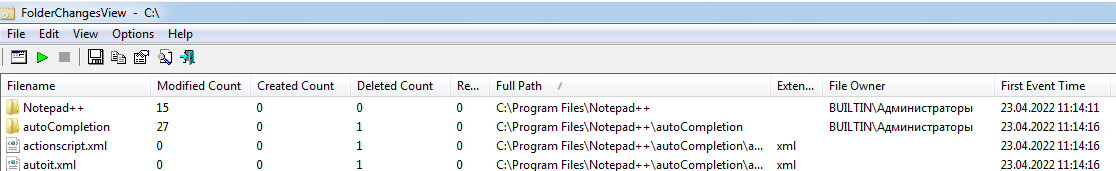
### Удаление приложения

*Изменения в реестре*

Программа Regshot зафиксировала удаление всех, созданных при установке ключей.



*Изменение файловой структуры*

Созданные каталоги, подкаталоги и временные файлы были удалены.

Программа удалилась полностью и не оставила после себя никаких файлов и «мусора».

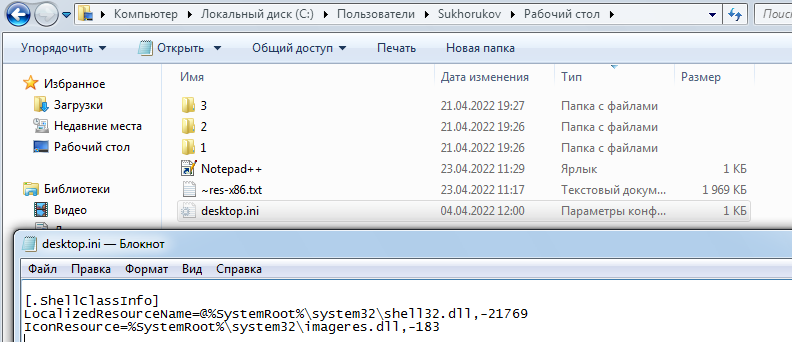
## Графический интерфейс

ОС Windows строится на методологии объектного подхода, в соответствии с которым весь мир и любая его часть рассматривается как совокупность объектов. Объекты обладают определенными свойствами и поведением. Причём разные объекты обладают разными свойствами и поведением.

Основными элементами графического интерфейса ОС Windows являются следующие объекты: Рабочий стол, окна, значки, ярлыки, кнопки, панели, меню, папки, приложения и документы.

### Рабочий стол

Рабочим столом называется основной элемент графического интерфейса, который обеспечивает эффективный доступ пользователя ко всем ресурсам компьютера, к наиболее часто используемым программам, документам и аппаратным средствам.

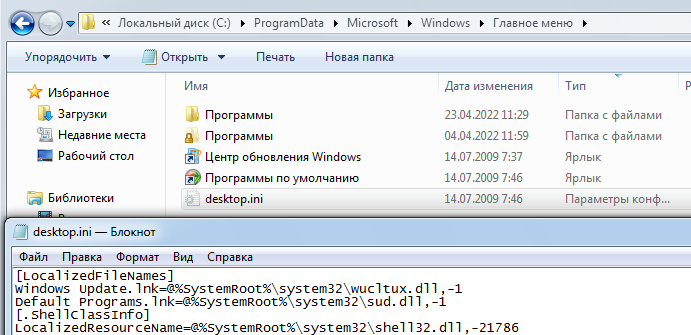
Ярлыки, каталоги и файлы, отображаемые на рабочем столе, находятся в каталоге C:\Users\<User's\_name>\Desktop. Так же в этом каталоге находится файл конфигурации desktop.ini.

### Главное меню

Меню называется элемент интерфейса пользователя, представляющий собой список альтернативных вариантов команд, действий режимов, установок и т.д., из которых пользователь может выбрать только один вариант. Отдельные варианты, из которых состоит список принято называть пунктами или строками меню.

Главное меню отображается при нажатии на значок Windows или нажатием соответствующей клавиши на клавиатуре.

Каталог главного меню - C:\ProgramData\Microsoft\Windows\Start Menu. В подкаталоге «Programs» отображается список установленных программ. В файле desktop.ini находятся пути, на которое ссылаются ярлыки.

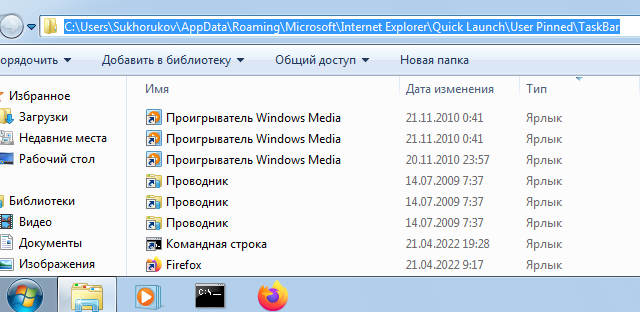


Через главное меню можно зайти в панель управления и папки текущего пользователя. Утилиты, доступные в панели управления хранятся в каталогах C:\Windows\winsxs и C:\Windows\System32, каталоги текущего пользователя - C:\Users\<User's\_name>.

### Панель задач

Панелью называется элемент графического интерфейса пользователя, который служит для объединения группы логически взаимосвязанных значков, элементов управления и индикаторов состояния операционной системы или выполняющей программы.

Панель задач располагается внизу рабочего стола. На ней отображаются запущенные приложения, ярлыки, по которым можно быстро открыть приложение, настройки времени, языка и сети.

Закрепленные ярлыки находятся в каталоге C: \Users \<User's\_name> \AppData \Roaming \Microsoft \Internet Explorer \Quick Launch \User Pinned \TaskBar.

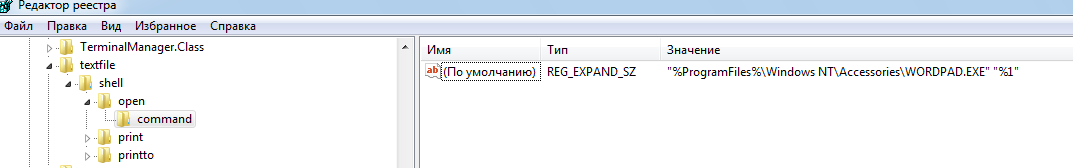
Программы, настройки времени, сети и языка находятся в каталоге C:\Windows и подкаталоге \System32.

### Контекстное меню

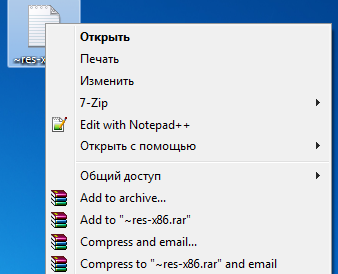
Контекстное меню — элемент графического интерфейса операционной системы, представляющий собой список команд, вызываемый пользователем для выбора необходимого действия над выбранным объектом. Команды контекстного меню относятся к тому объекту, над которым это меню было вызвано.

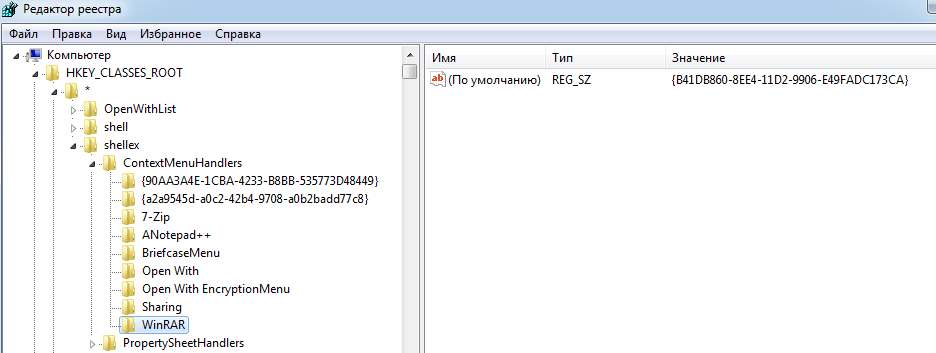
Контекстное меню папок и файлов Проводника, которое содержит такие команды, как "Открыть", "Изменить", "Открыть с помощью", "Создать" (с выбором типа файла) и т.д.. Оно может быть изменено путём настройки некоторых разделов реестра.

Информация о каждом зарегистрированном типе файлов и командах, применимых к нему, хранится в ветви реестра HKEY\_CLASSES\_ROOT. Каждому зарегистрированному типу файлов соответствуют два подраздела в разделе HKEY\_CLASSES\_ROOT. Первый подраздел имеет название, соответствующее расширению имени файла. В своём значении "по умолчанию" этот подраздел содержит идентификатор, который используется как название второго подраздела. Во втором подразделе хранятся описания команд, применяемых к данному типу файлов.

 Например, текстовому типу файлов в реестре соответствуют два подраздела - ".txt" и "txtfile". Подраздел shell раздела "txtfile" содержит в себе подразделы, определяющие действия над данным типом файлов. Каждый подраздел-глагол может содержать подраздел command. Этот подраздел содержит командную строку в параметре "по умолчанию". Параметр "%1" в командной строке будет заменён путём и именем выбранного файла.

Информация, относящаяся одновременно ко всем типам файлов, хранится в подразделе "\*" ветви реестра HKEY\_CLASSES\_ROOT. Информация, относящаяся ко всем незарегистрированным типам файлов, хранится в подразделе "Unknown" ветви реестра. Информация, относящаяся к папкам, хранится в подразделах "Directory" и "Folder" ветви реестра.

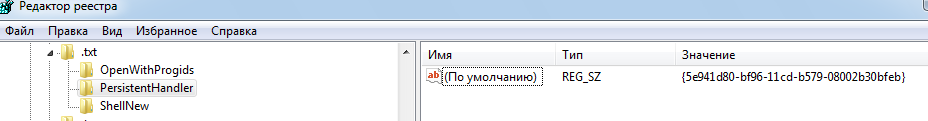
На компьютере Sukhorukov-1 установлено 3 программы, которые добавили свои действия в контекстное меню: 7Zip, Winrar, Notepad++. Информацию об этих действиях можно увидеть в реестре



Информация из реестра для контекстного меню всех типов Контекстное меню файла типа .txt

### Ассоциация файлов

Файлы одного типа могут быть открыты разными программами. Информация о доступных программа записана в реестре. Подраздел с названием типа файла, например, .txt содержит информацию о программах, которыми можно открыть данный тип файла.



Ключ содержит 3 подключа:

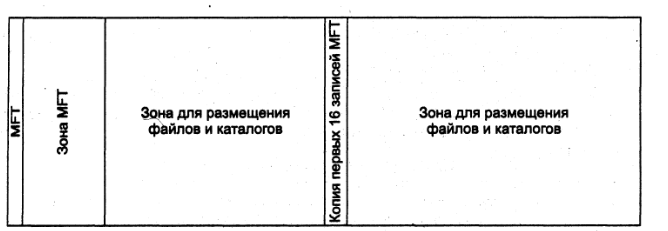
* OpenWithProgIds – идентификаторы программы, которыми можно открыть данный тип файла.
* PersistendHandler (Постоянный обработчик) – программа, которая выбрана приоритетной для данного типа файла.
* ShellNew – возможность создать данный тип файла через меню «Создать».

## Базовые свойства файловой структуры NTFS

### Структура файловой системы NTFS

Всё дисковое пространство в NTFS делится на две неравные части. Первые 12 % диска отводятся под MFT-зону – пространство, которое может занимать, увеличиваясь в размере, главный служебный метафайл MFT. Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится пустой – это делается для того, чтобы самый главный, служебный файл (MFT) по возможности не фрагментировался при своем росте. Остальные 88 % тома представляют собой обычное пространство для хранения файлов.

MFT (masterfiletable, общая таблица файлов) представляет собой централизо­ванный каталог всех остальных файлов диска, в том числе и себя самого. MFT поделен на записи фиксированного размера в 1 Кбайт, и каждая запись соответ­ствует какому-либо файлу. Первые 16 файлов но­сят служебный характер и недоступны операционной системе – они называются метафайлами, причем самый первый метафайл – сам MFT. Эти первые 16 эле­ментов MFT– единственная часть диска, имеющая строго фиксированное поло­жение. Копия этих же 16 записей хранится в середине тома для надёжности, по­скольку они очень важны. Остальные части MFT-файла могут располагаться, как и любой другой файл, в произвольных местах диска – восстановить его по­ложение можно с помощью его самого, «зацепившись» за самую основу – за пер­вый элемент MFT.



Структура файловой системы NTFS

Первые 16 файлов NTFS (метафайлы) носят служебный характер; каждый из них отвечает за какой-либо аспект работы системы. Метафайлы находятся в корневом каталоге NTFS-тома. Все они начинаются с символа имени «$».

|  |  |
| --- | --- |
| **Имя метафайла** | **Назначение метафайла** |
| $MFT | Основная таблица MFT |
| $MFTmirr | Копия первых шестнадцати записей MFT (размещенная ровно посередине тома) |
| $Boot | Загрузчик |
| . | Корневой каталог |
| $LogFile | Журнал файловой системы |
| $Volume | Служебная информация (метка и ID тома, версия файловой системы, т.д.) |
| $Bitmap | Карта свободного места тома |
| $AttrDef | Список стандартных атрибутов файлов на томе |
| $Quota | Записи с правами пользователей на использование дискового пространства (квотами) |
| $Secure | Дескрипторы безопасности файловых объектов (права доступа) |

Все файлы тома упоминаются в MFT. В этой структуре хранится вся ин­формация о файлах, за исключением данных. Имя файла, размер, положение на диске отдельных фрагментов и т. д. – всё это хранится в соответ­ствующей записи. Если для информации не хватает одной записи MFT, то ис­пользуется несколько записей, причем не обязательно идущих подряд. Файлы могут иметь не очень большой размер. Тогда данные файла хранятся прямо в MFT, в оставшемся от основных данных месте в пределах одной записи MFT. Файлы, занимающие сотни байт, обычно не имеют своего «физического» воплощения в основной файловой области – все данные такого файла хранятся в одном месте, в MFT.

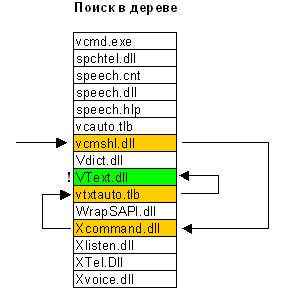
Файл в томе с NTFS идентифицируется файловой ссылкой (FileReference), которая представляется как 64-разрядное число. Файловая ссылка состоит из номера файла, который соответствует позиции его файловой записи в MFT, и номера последовательности. Последний увеличивается всякий раз, когда данная позиция в MFT используется повторно, что позволяет файловой системе NTFS выполнять внутренние проверки целостности.

Каждый файл в NTFS представлен с помощью потоков (streams). Один из потоков содержит данные файла. Большинство атрибутов файла – это тоже потоки. Таким об­разом, получается, что базовая сущность у файла только одна – номер в MFT, а всё остальное, включая и его потоки, – опционально.

Стандартные атрибуты для файлов и каталогов в томе NTFS имеют фиксиро­ванные имена и коды типа.

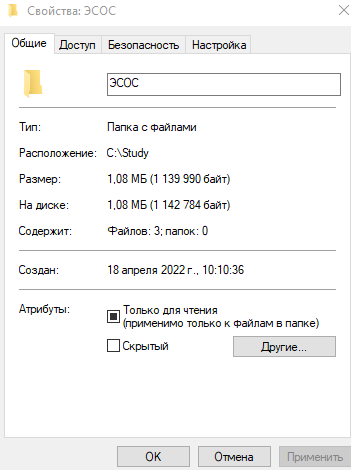
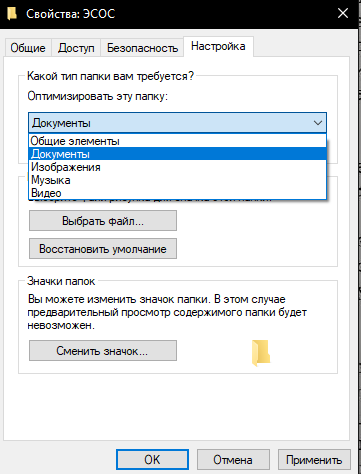
|  |  |
| --- | --- |
| **Системный атрибут** | **Описание атрибута** |
| Стандартная информация о файле | Традиционные атрибуты Read Only, Hidden, Archive, System, отметки времени, включая время создания или последней модификации, число каталогов, ссылающихся на файл |
| Список атрибутов | Список атрибутов, из которых состоит файл, и файловая ссылка на файловую запись и MFT, в которой расположен каждый из атрибутов. |
| Имя файла | Имя файла в символах Unicode. Файл может иметь несколько атрибутов – имён файла. |
| Дескриптор защиты | Структура данных защиты (ACL), предохраняющая файл от несанкционированного доступа. Атрибут «дескриптор защиты» определяет, кто владелец файла и кто имеет доступ к нему |
| Данные | Данные файла, его содержимое. В NTFS у файла по умолчанию есть один безымянный атрибут данных, и он может иметь дополнительные именованные атрибуты данных. У каталога нет атрибута данных по умолчанию, но он может иметь необязательные именованные атрибуты данных |
| Корень индекса, размещение индекса, битовая карта (только для каталогов) | Атрибуты, используемые для индексов имён файлов в больших каталогах |

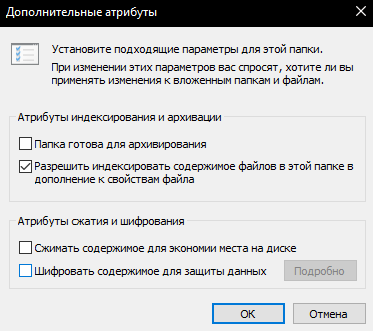
Каталог на NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске. Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT, который уже предоставляет полную информацию об элементе каталога. Внутренняя структура каталога представляет собой бинарное дерево. Для поиска файла с данным именем в линейном каталоге, таком, например, как у FAT-а, операционной системе приходится просматривать все элементы каталога, пока она не найдет нужный. Бинарное дерево располагает имена файлов таким образом, чтобы поиск файла осуществлялся более быстрым способом — с помощью получения двухзначных ответов на вопросы о положении файла. Вопрос, на который бинарное дерево способно дать ответ, таков: в какой группе, относительно данного элемента, находится искомое имя — выше или ниже? Поиск начинается с такого вопроса к среднему элементу, и каждый ответ сужает зону поиска в среднем в два раза. Файлы, скажем, просто отсортированы по алфавиту, и ответ на вопрос осуществляется очевидным способом — сравнением начальных букв. Область поиска, суженная в два раза, начинает исследоваться аналогичным образом, начиная опять же со среднего элемента.

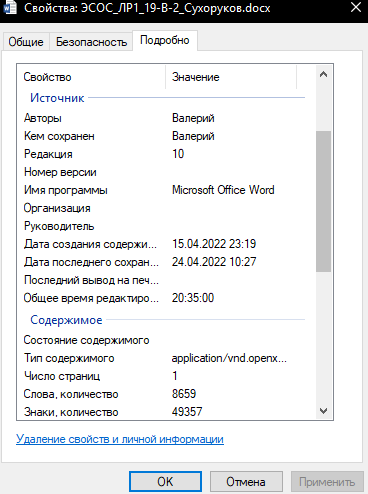
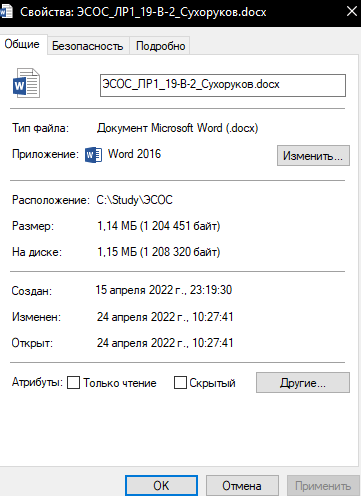


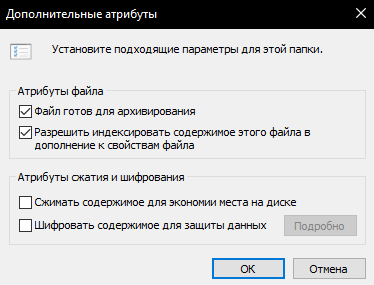
### Просмотр свойства файлов и каталогов через контекстное меню

С помощью контекстного меню определим настраиваемые свойства каталога. На вкладке «Общее» можно отображаются сведения о каталоге и присутствует возможность настройки атрибутов. На вкладках «Доступ» и «Безопасность» есть возможность настроить общий доступ к каталогу и права, которые будут у пользователей. На вкладке «Настройка» можно «оптимизировать папку» по типу содержимого.



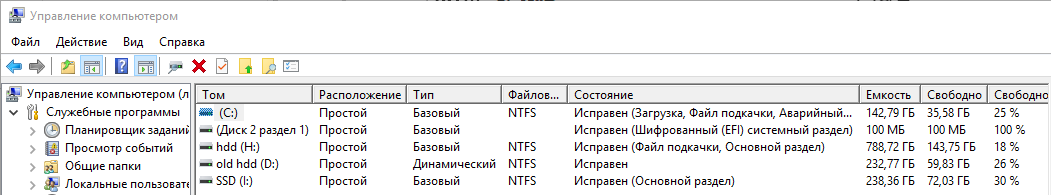


У файла также есть возможность настройки атрибутов и общего доступа, на вкладке «Подробно» можно увидеть дополнительную информацию о файле.

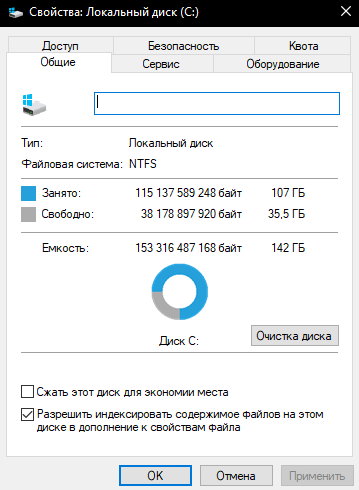


### Операции с томом, каталогами и файлами

#### Получение информации о файловой системе тома

Через консоль «Управление компьютером»:

В атрибуте «Файловая система» у каждого тома указан тип файловой системы – NTFS.

Через интерфейс проводника:

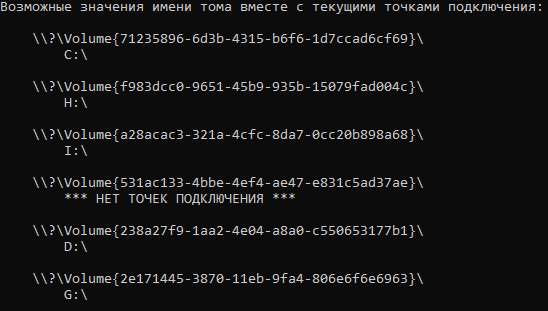
В свойствах логического диска указан тип файловой системы - NTFS.

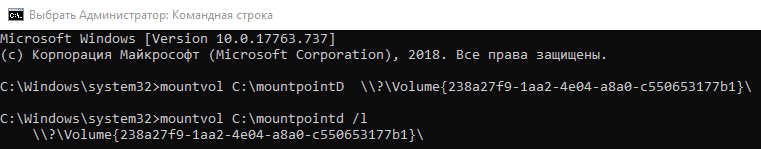
#### Монтирование тома к точке соединения NTFS.

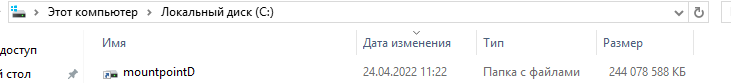
Точка монтирования (mount point) — это каталог или файл, с помощью которого обеспечивается доступ к новой файловой системе, каталогу или файлу. Точка монтирования используется для реализации возможности динамически присоединять/отсоединять разделы диска к файловой системе во время работы операционной системы.

Команда **MOUNTVOL** позволяет создавать, удалять и просматривать точки подключения томов (точки монтирования) в командной строке Windows. Точки монтирования доступны при использовании файловой системы NTFS. В среде операционных систем семейства Windows, существует два вида точек монтирования: точка монтирования каталога (junction point) и точка монтирования тома (volume mount point). Создание точек монтирования первого типа осуществляется через консольную команду mklink /J, создание точек монтирования второго типа — через команду mountvol.

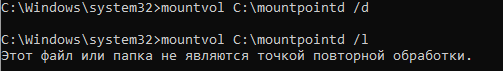
При вводе команды mountvol без параметров отображается перечень томов с точками монтирования:



* **mountvol C:\mountpointD \\?\Volume{238a27f9-1aa2-4e04-a8a0-c550653177b1}\** - подключить том с именем \\?\Volume{238a27f9-1aa2-4e04-a8a0-c550653177b1}\ в качестве папки C:\mountpointD.
* **mountvol C:\mountpointd /l** - отобразить список подключенных томов для папки C:\mountpointD

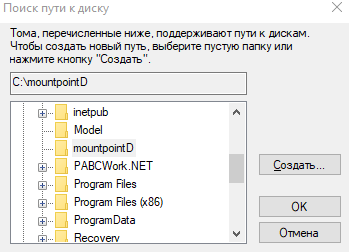
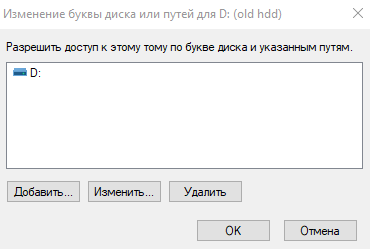


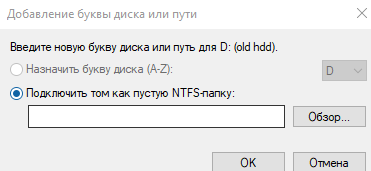
* **mountvol C:\mountpointd /d**- удалить точку монтирования тома в виде папки C:\mountpointD.

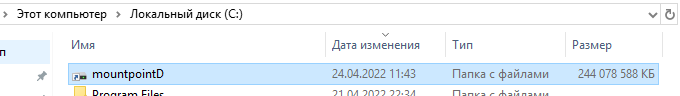


Монтировать том можно через оснастку диспетчер дисков (diskmanager). Для этого нужно:

* Выбрать целевой том, в контекстном меню выбрать команду «Изменить букву диска или путь к диску».
* Нажать кнопку «Добавить», выбрать функцию «Подключить том как пустую NTFS-папку» и указать путь к целевой папке.







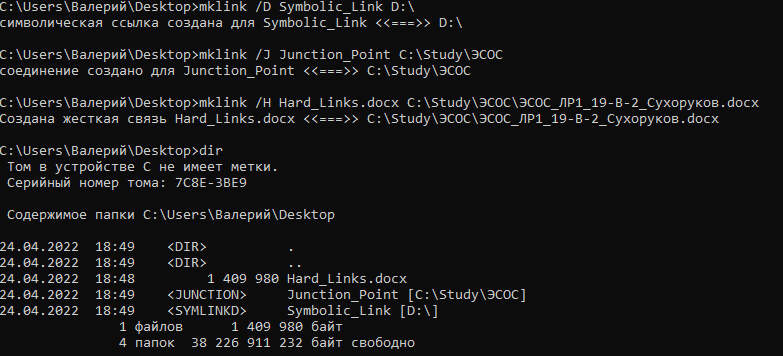
#### Создание hardlink, symlink, junction point

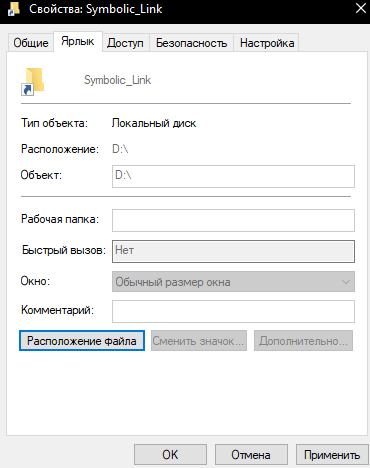
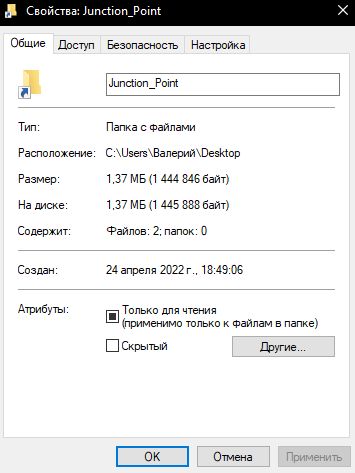
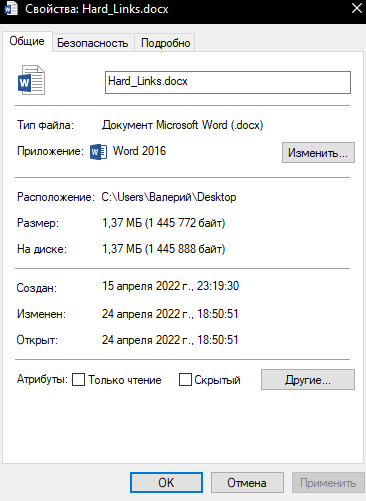
Ядро Windows поддерживает следующие виды ссылок:

* Hard Links — жёсткие ссылки. Можно создавать только на файлы.
* Junction Points — аналог символических ссылок. Можно создавать только на директории.
* Symbolic Links — символическая ссылки. Можно создавать на файлы и директории.

Для создания ссылок можно использовать программу mklink.exe в командной строке cmd. Формат командной строки MKLINK:

MKLINK [[/D] | [/H] | [/J]] Ссылка Назначение

* /D - Создание Symbolic Links на каталог. По умолчанию создается символическая ссылка на файл.
* /H - Создание Hard Links для файла.
* /J - Создание Junction Points для каталога.
* Ссылка - Имя новой символической ссылки.
* Назначение - Путь (относительный или абсолютный), на который ссылается создаваемая ссылка.



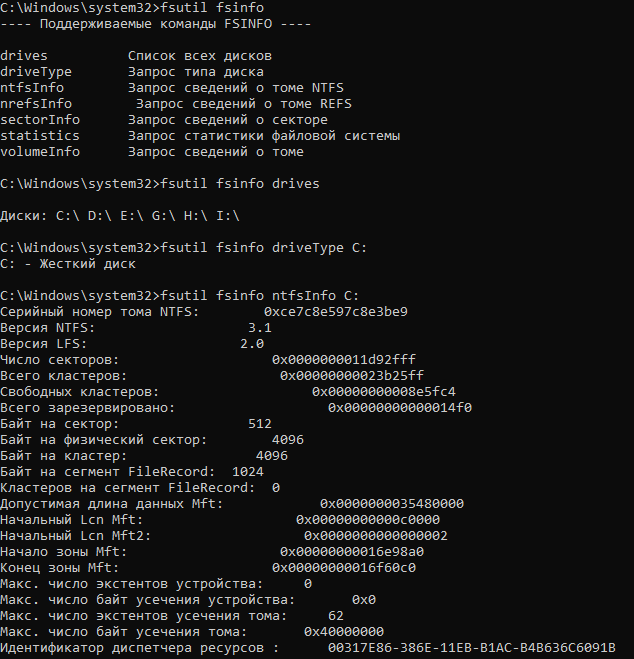
Список свойств ярлыков Hard link и Junction points в интерфейсе проводника совпадает со списком свойств для «обычных» файлов и каталогов. У ярлыка Junction point установлен атрибут «Только для чтения». В свойствах Symbolic link отображается дополнительная вкладка «Ярлык», в которой указан путь к каталогу.

#### Использование команды fsutil

Fsutil является служебной программой командной строки, которая используется для выполнения связанных задач файловых систем FAT и NTFS, таких как управление точками повторной обработки, управление разреженными файлами, отключение тома или расширение тома.

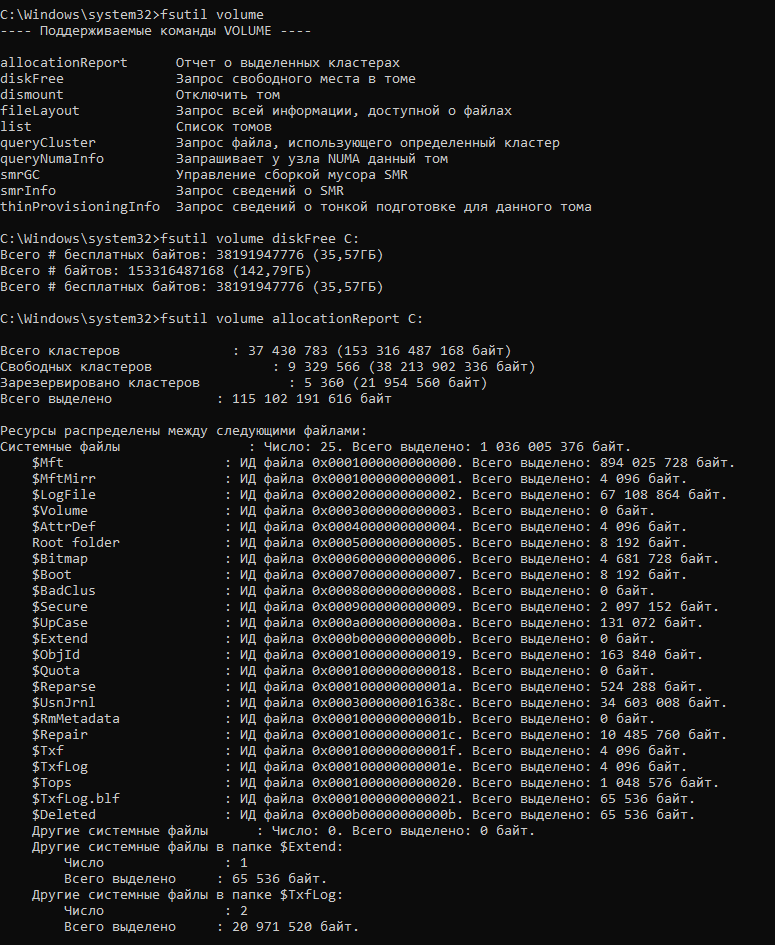
* Получение информации о файловой системе в целом– fsutil fsinfo:

Запросим название всех подключённых дисков, тип диска С: , информацию о томе С:.



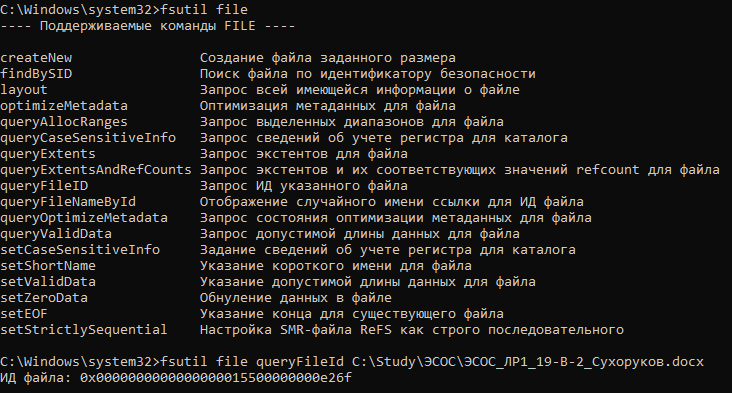
* Получение информации о томах – fsutil volume:

Отобразим информацию о свободном месте и о выделенных кластерах.



* Получение информации о файлах– fsutil file:

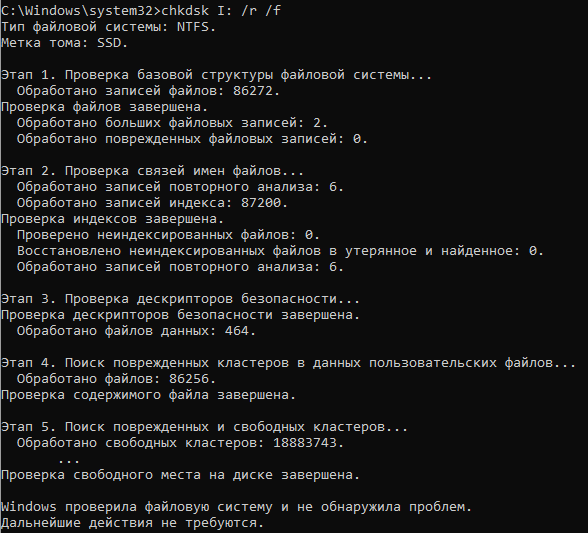
Отобразим ID файла ЭСОС\_ЛР1\_19-В-2\_Сухоруков.docx.



### Примеры использования chkdsk, diskpart, vssadmin

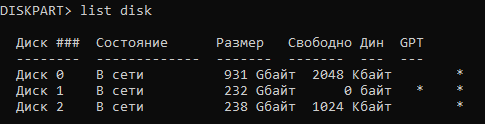
* **Сhkdsk** проверяет файловую систему и метаданные файловой системы тома на наличие логических и физических ошибок.

Поиск ошибок на диске и их исправление – chkdsk D: /r /f. Параметр “/f” указывает CHKDSK выполнить исправление найденных ошибок; “/r” указывает программе обнаружить повреждённые секторы на диске и восстановить информацию, которую возможно прочитать.



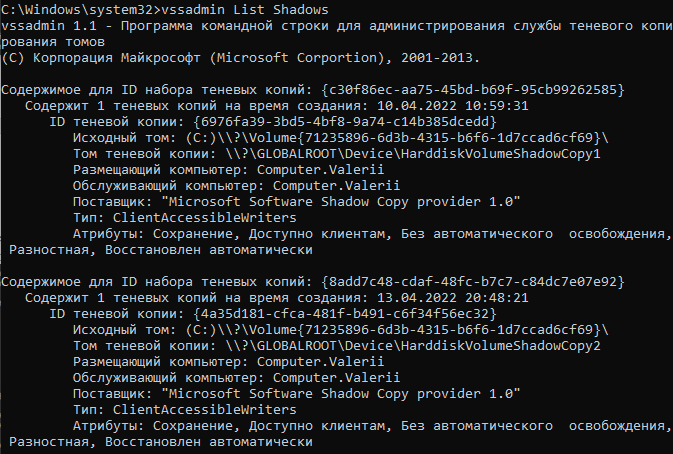
* **DiskPart** помогает управлять дисками компьютера (дисками, разделами, томами или виртуальными жесткими дисками).

Отображение подключенных дисков – diskpart list disk



* **Vssadmin** отображает резервные копии текущих теневых копий томов и всех установленных модулей записи и поставщиков теневого копирования.

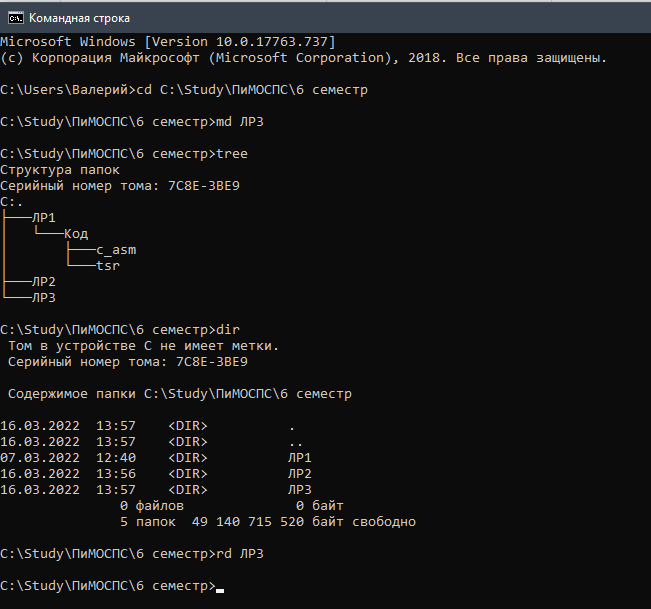
Вывести список существующих теневых копий томов - vssadmin List Shadows



## Командный язык и работа в режиме командной строки

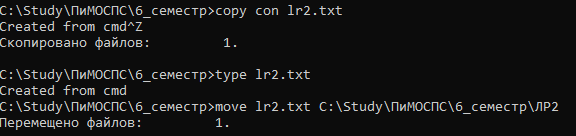
### Основные команды работы с каталогами

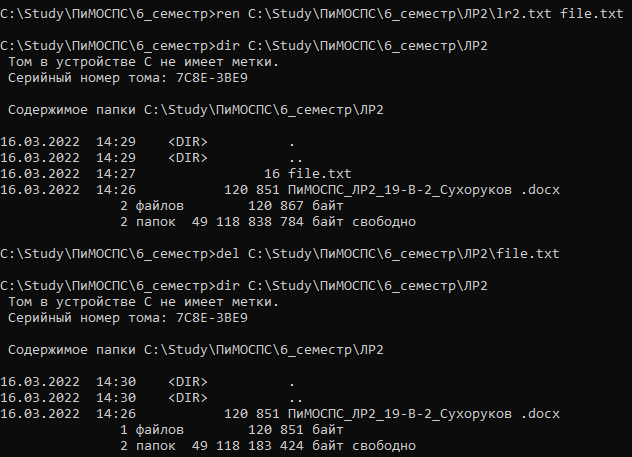
* MD — Создание каталога. Синтаксис: (MD имя каталога, есть возможность указать путь расположения).
* RD — Удаление каталога. Синтаксис: (RD имя каталога, есть возможность указать путь расположения). Примечание, RD без ключей позволяет удалить только пустой каталог.
* CD — Смена текущего каталога. Синтаксис: (CD имя каталога, есть возможность указать путь расположения). CD\ — Переход в корневой каталог. CD.. — Переход в родительский каталог.
* DIR — Просмотр каталога в виде списка. Синтаксис: (DIR имя каталога, есть возможность указать путь расположения).
* TREE — Вывод каталогов в графическом представлении. Синтаксис: (TREE имя каталога, есть возможность указать путь расположения).
* MOVE — Перемещение\Переименование каталога. Синтаксис: (MOVE что\_перемещаем куда\_перемещаем, есть возможность указать путь расположения).
* XCOPY — Копирование структур каталогов. Синтаксис: (XCOPY что\_копируем куда\_копируем, есть возможность указать путь расположения).



### Команды работы с файлами

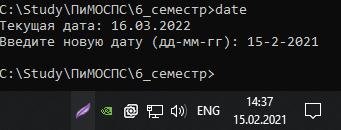
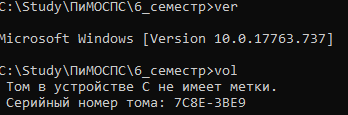
* COPY CON — Создание файла. Синтаксис: (COPY CON имя\_файла\_с\_расширением, есть возможность указать путь расположения).
* TYPE — Вывод содержимого файла на экран. Синтаксис: (TYPE имя\_файла\_с\_расширением, есть возможность указать путь расположения).
* DEL — Удаление файла. Синтаксис: (DEL имя\_файла\_с\_расширением, есть возможность указать путь расположения).
* COPY — Копирование файла\Обьединения файлов. Синтаксис(копирование): (COPY что\_копируем куда\_копируем, есть возможность указать путь расположения). Синтаксис(обьединение): (COPY имя\_файла + имя\_файла + имя\_файла… имя\_файла\_обьединения, есть возможность указать путь расположения).
* MOVE — Перемещение\Переименование файла. Синтаксис: (MOVE что\_перемещаем куда\_перемещаем, есть возможность указать путь расположения).
* REN — Переименование файлов. Синтаксис: (REN что\_переименовываем вочто\_переименовываем, есть возможность указать путь расположения).
* FC — Сравнение файлов. (FC имя\_файла имя\_файла имя\_файла…, есть возможность указать путь расположения).





Команды системного назначения

* CLS — Отчистка экрана.
* DATE — Просмотр и вывод на изменения текущей даты.
* VER — Вывод информации о текущей версии Операционной системы.
* VOL — Вывод информации о томе-логическом разделе диска.
* SYSTEMINFO — Вывод информации о конфигурациях системы.
* EXIT — Выход из командной строки.

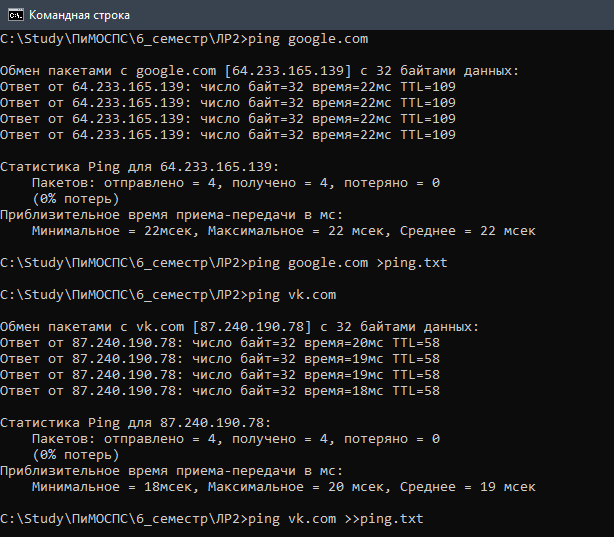


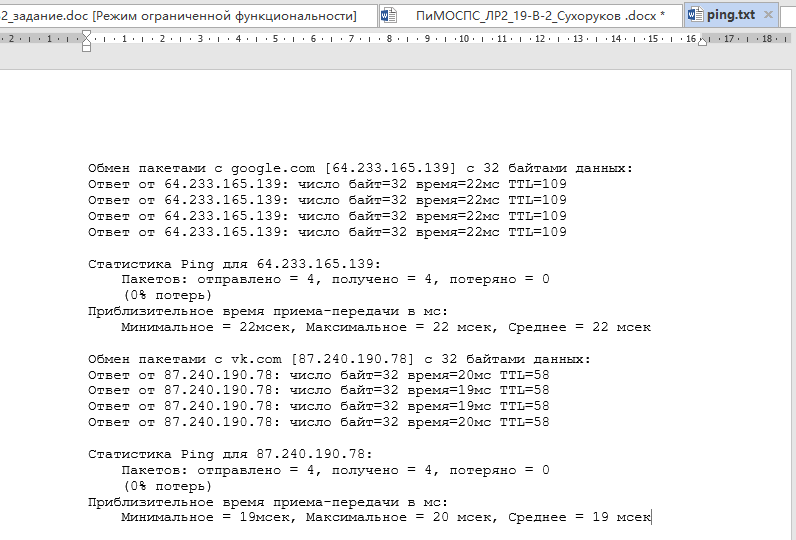
### Перенаправление ввода-вывода

С помощью переназначения устройств ввода/вывода одна программа может направить свой вывод на вход другой или перехватить вывод другой программы, используя его в качестве своих входных данных. Таким образом, имеется возможность передавать информацию от процесса к процессу при минимальных программных издержках. Практически это означает, что для программ, которые используют стандартные входные и выходные устройства, операционная система позволяет:

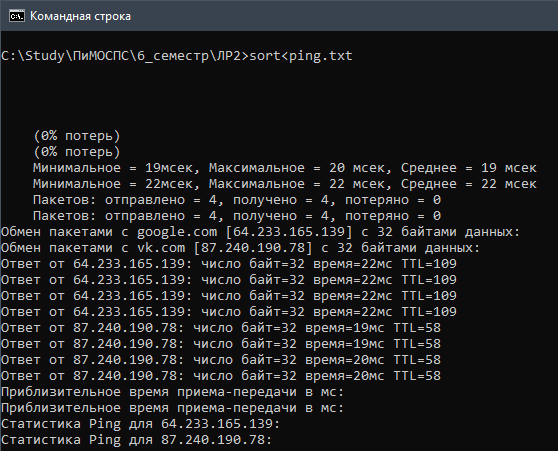
* выводить сообщения программ не на экран (стандартный выходной поток), а в файл или на принтер (перенаправление вывода);
* читать входные данные не с клавиатуры (стандартный входной поток), а из заранее подготовленного файла (перенаправление ввода);
* передавать сообщения, выводимые одной программой, в качестве входных данных для другой программы (конвейеризация или композиция команд).

Из командной строки эти возможности реализуются следующим образом. Для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой из командной строки, в текстовый файл, нужно использовать конструкцию **команда> имя\_файла.** Если при этом заданный для вывода файл уже существовал, то он перезаписывается (старое содержимое теряется), если не существовал создается. Можно также не создавать файл заново, а дописывать информацию, выводимую командой, в конец существующего файла. Для этого команда перенаправления вывода должна быть задана так: **команда>> имя\_файла**. С помощью символа **<**можно прочитать входные данные для заданной команды не с клавиатуры, а из определенного (заранее подготовленного) файла: **команда <имя\_файла.**

Команда **PING** позволяет выполнить отправку управляющего сообщения типа Echo Request (тип равен 8 и указывается в заголовке ICMP-сообщения) адресуемому узлу и интерпретировать полученный от него ответ в удобном для анализа виде.



Команда **SORT** используется для сортировки в алфавитном порядке строк текстового файла или стандартного вывода.



### Конвейеры команд и фильтры

Вывод первой команды является вводом для второй, вывод второй - вводом для третьей и т.д.

process1|process2|…process

**Фильтром** называется программа, принимающая на вход информационный поток, преобразующая его по определенному алгоритму и выводящая результат работы (команды find,sort,more)

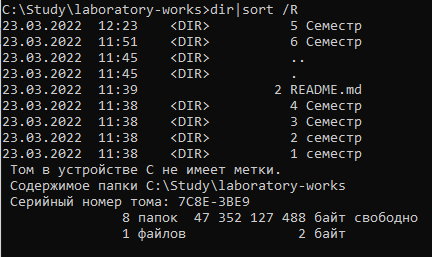
**Команда MORE** – постраничное отображение содержимого текстового файла (внешняя команда, файл …\Windows\System32\more.exe).

**Команда SORT** – сортировка строк текстового файла в алфавитном порядке.

* /R – сортировать в обратном порядке (от Z к A, от 9 к 0).

**Команда FIND** – поиск заданной последовательности символов (string) в одном или нескольких текстовых файлах (внешняя команда, файл …\Windows\System32\find.exe).

* /V – выводить только строки, не содержащие string;
* /C – вместо строк выводить их порядковые номера;
* /N – перед каждой выводимой строкой выводить и ее номер.



В данном примере обе команды запускаются одновременно, но команда sort приостанавливает работу до получения выходных данных команды dir. Команда sort использует выходные данные команды dir в качестве своих входных данных, а затем свои выходные данные отправляет в дескриптор 1 (STDOUT).

### Пример командного файла

@echo off

SET NAME="main"

SET -p Name="Enter file's name: "

ECHO TRANSLATION:

%TASM% -la -zi %NAME%

IF NOT EXIST %NAME%.obj GOTO error\_translation

ECHO LINKING:

%TLINK% -t %NAME%

IF NOT EXIST %NAME%.com GOTO error\_linking

ECHO FINISH

ECHO BUILDED: %NAME%.obj, %NAME%.lst, %NAME%.com, %NAME%.map

GOTO exit

:error\_translation

ECHO TRANSLATION FAULT

GOTO exit

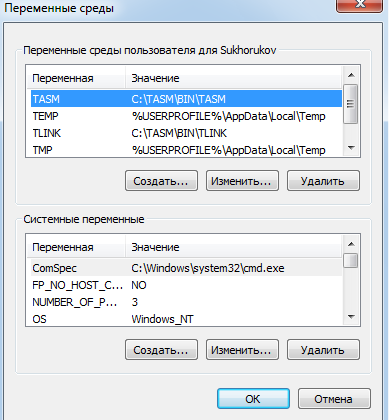
:error\_linking

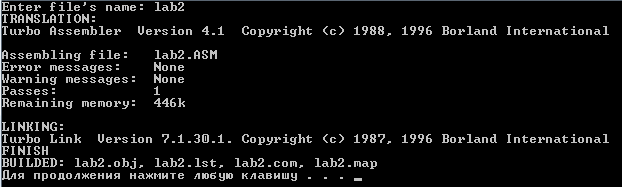
ECHO LINKING FAULT

:exit

PAUSE

Данный файл компилирует DOS программу из исходного файла, имя которого задаёт пользователь. Для ввода данных используется команда SET с ключом –p. При невозможности трансляции или компановки выводятся соответствующие сообщения, и программа переходит к соответствующей метке, проверка производится командой IF. Пути к программам заданы через переменные среды - %TASM% и %TLINK%.





Настройка переменных среды Результат работы командного файла

## Механизмы обмена данными между приложениями

### Clipboard

Clipboard называют универсальным буфером обмена, так как пользователь может записать в него самую различную информацию. Это общая область для хранения дескрипторов данных, через которую прикладные программы могут обмениваться форматированными данными.

Методика работы с Clipboard одинакова для всех приложений и обычно заключается в том, что пользователь выделяет нужную часть документа или изображения, а затем выбирает из меню "Edit" строки "Copy" или "Cut". В первом случае выделенный фрагмент копируется в Clipboard, во втором - также копируется, но после копирования фрагмент удаляется из документа. Clipboard может содержать данные одновременно в нескольких форматах.

Во-первых, приложение может записать в Clipboard данные в одном из форматов, предопределенных для Windows. Можно записать данные в текстовом формате, битовое изображение в формате, зависящем от устройства отображения (DDB), цветовую палитру, битовое изображение в формате, независящем от устройства отображения (DIB), в виде метафайла, а также в нескольких других форматах, созданных на базе текстового формата данных.

Во-вторых, приложение может использовать свой собственный, уникальный формат данных, зарегистрировав его в Windows при помощи специальной функции. Память, в которой хранятся данные, должна быть доступна для всех приложений. Кроме того, содержимое памяти не должно исчезать при завершении работы отдельных файлов.

### DDE – технология (Dynamic Data Exchange)

DDE – технология позволяет создать постоянно действующие каналы между несколькими одновременно работающими приложениями Windows. Эти каналы могут создаваться автоматически при запуске приложения или при необходимости, а также по явному запросу пользователя. После того как каналы созданы, они будут работать без вмешательства пользователя. Приложения, использующие технологию динамического обмена данных DDE, выступают как клиенты или серверы (или одновременно как клиенты и серверы). Начиная с версии 3.1 в составе Windows появилось расширение - управляющая библиотека динамического обмена данными DDEML, выполненная как обычная DLL-библиотека. Механизм Network DDE позволяет организовать каналы обмена данными между приложениями, которые запущены на разных рабочих станциях сети. Большая часть DDE- функций реализована в последних версиях OLE.

### OLE (Object Linking and Embedding)

Работа с Windows носит документно-ориентированный характер. Однако документы этой системы могут иметь сложную структуру, объединяющую тексты с графиками и различными объектами мультимедиа. Для создания и применения таких документов служит специальная технология, реализуемая системой Windows.

Механизм OLE (Object Linking and Embedding - Связывание и Встраивание Объектов) позволяет обеспечить интеграцию различных по своей сути объектов. При связывании объекта с документом создается ссылка на его файл (технология OLE1 – жесткая ссылка, технология OLE2 – относительная ссылка) OLE является набором средств, позволяющим легко подготавливать документы, включающие в себя данные, подготовленные в различных приложениях. В Windows 95 используется новая реализация этого метода, известная как OLE-2.

В отличие от операции встраивания, при выполнении операции связывания внутрь документа помещается не сам объект, а только ссылка на него (путь исходного файла, подготовленного приложением-сервером).

Использование операции связывания вместо операции вставки имеет два важных преимущества:

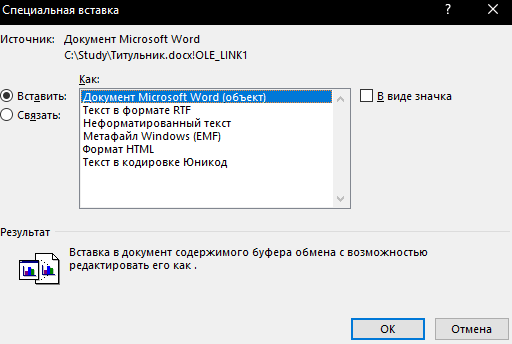
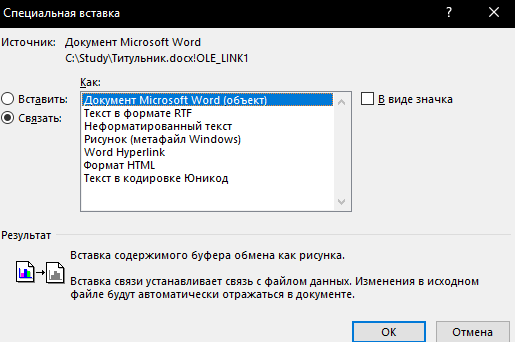
* Если объект, подготовленный приложением-сервером, используется в нескольких документах (приложениях-клиентах), то при изменении объекта, автоматически происходит обновление всех документов, которые содержат ссылку на этот объект.
* Так как в документе помещается только ссылка на связываемый объект, то размер документа получается меньше, чем при включении объекта в документ.

Однако, операция связывания имеет и свои недостатки:

* Так как связываемые объекты хранятся отдельно от документа, к которому они привязаны, то вместо одного файла документ будет состоять из множества файлов, расположенных в различных каталогах.
* При перемещении файла, содержащий связываемый объект в другой каталог, то необходимо изменить в документе ссылку на перемещенный объект.
* Активизация объекта по месту. Позволяет получить доступ к включенному объекту по месту, без переключения на другое окно или приложение. Есть возможность записывать, отображать, проигрывать и редактировать включенные объекты оставаясь в основном приложении (документе).
* Поддержка вложенных объектов. Позволяет взаимодействовать с объектами, вложенными в другие объекты
* Поддержка механизма Drag and Drop. Теперь можно перетаскивать объект из одного приложения в другое с помощью мыши. При этом происходит автоматическое встраивание объекта в документ другого приложения.

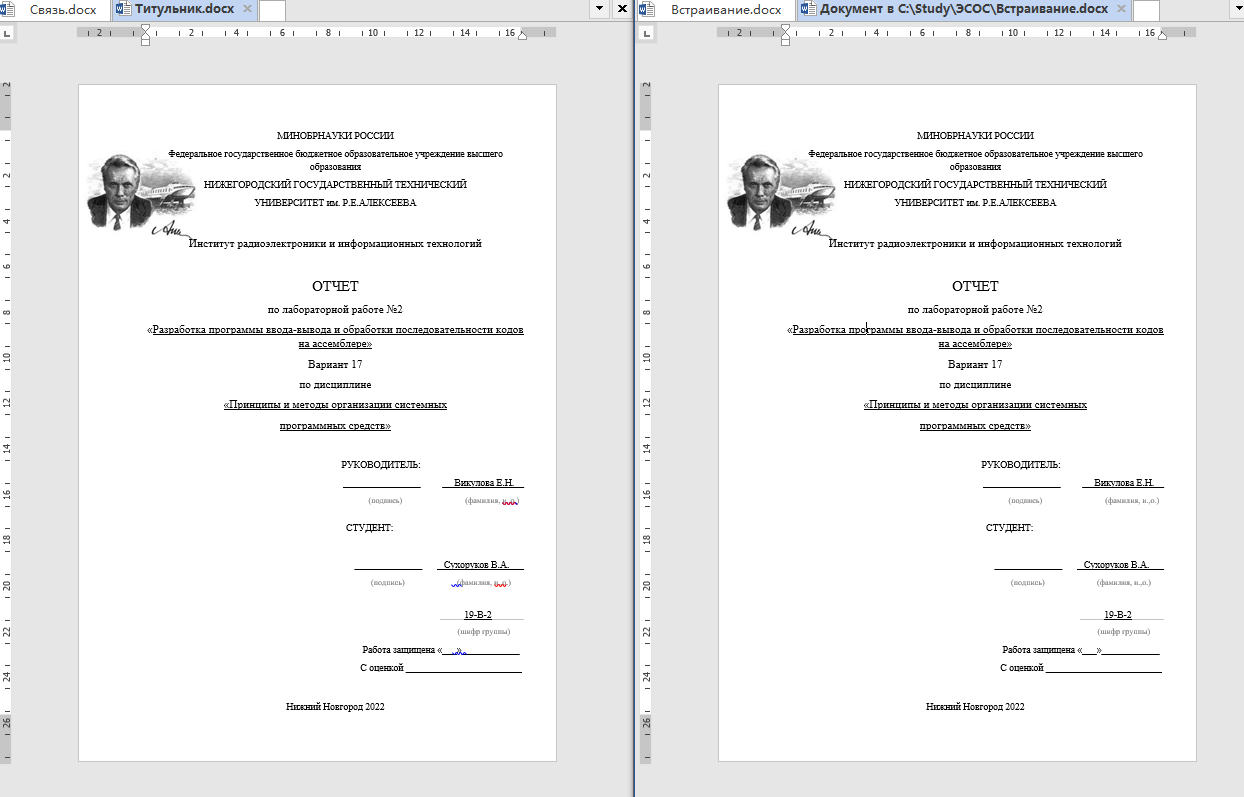
### Связывание и встраивание объекта в документ

Создадим два документа, в один из них встроим документ, содержащий титульный лист, другой свяжем с ней. Вставить или внедрить объект можно через меню «Специальная вставка»:

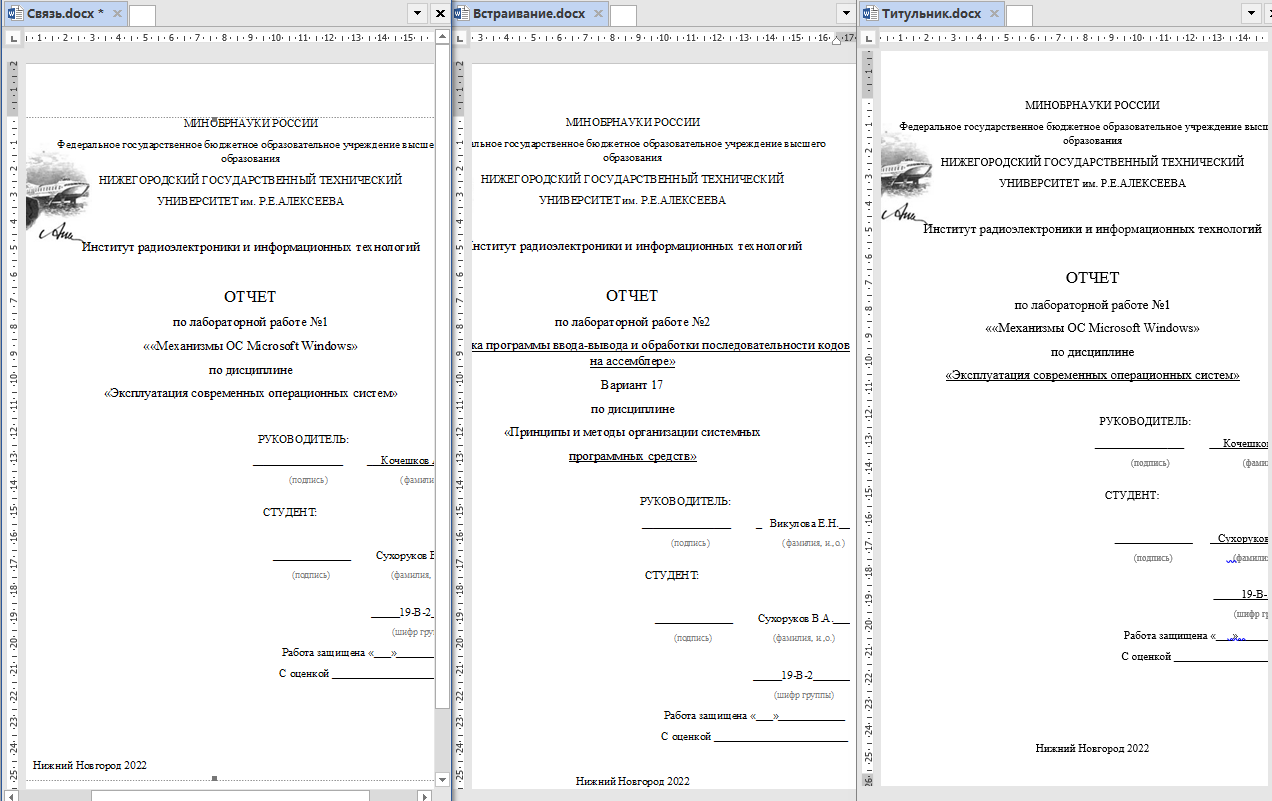


Свойства специальной вставки

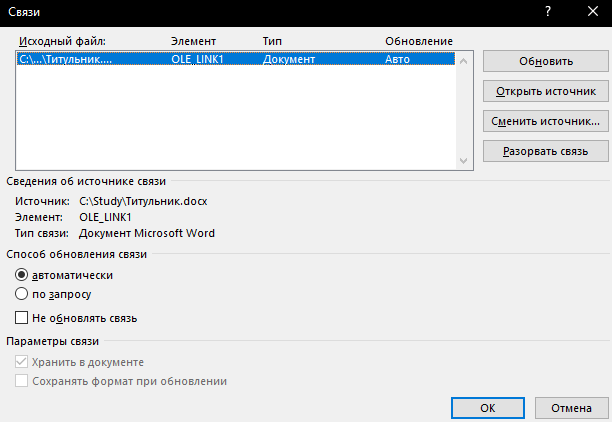
для файла «Связь.docx» для файла «Встраивание.docx»

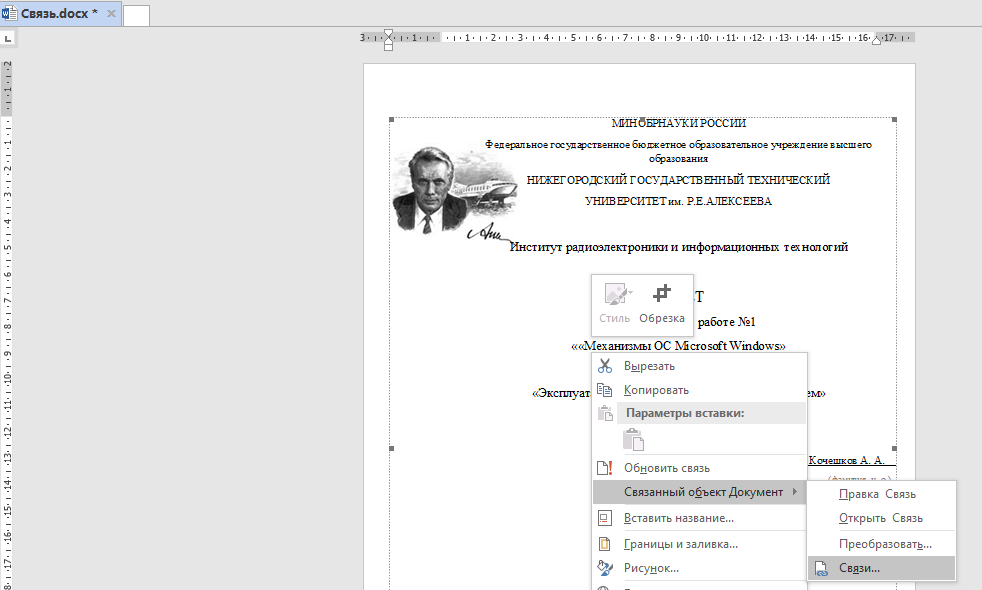


Изменим файл «Титульник.docx» и убедимся, что файл со связыванием изменится, со встраиванием – нет.

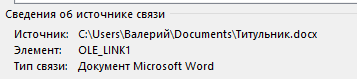


Через меню связи «Связи» отобразим связь на документ «Титульник.docx».

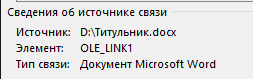




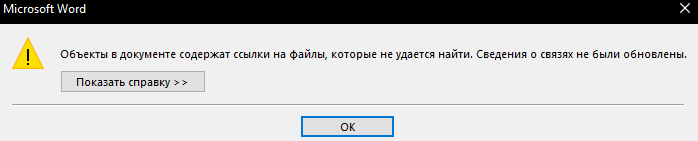
Переметим связанный файл в другой каталог на диске С:\ и обновим связь. Документ был найден в другом каталоге и связь обновилась.



Переместим файл на другой физический диск. Файл снова был найден.



Переместим файл на USB-флешнакопитель и обновим связь. На этот раз документ не был найден.



#### Вывод по сравнению встраивания и связывания

Встраивание файла позволяет редактировать встроенный объект внутри файла. Встроенный объект не нужно переносить вместе с основным документом, т.к. он является его частью. Связывание объекта и файла позволяет автоматически обновлять документы, которые связаны с файлом. Это удобно, если один объект связывается со множеством документов т.к. его изменение приведёт к изменению сразу всех документов. Минусом связывания является необходимость переносить его вместе с основным файлом.

# Вывод

В данной работе были изучены механизмы ОС Microsoft Windows: возможности разных редакций одной ОС, различия дисковой конфигурации, механизм и варианты загрузки.

Были изучены возможности и способ работы элементов графического интерфейса, рассмотрены основные системные каталоги и типы фалов.

В ходе работы были рассмотрены механизмы установки и функционирования приложений разного типа. При установке развитого Win32 приложения было запротоколировано изменение в файловой системе и реестре.

Были рассмотрены основные принципы файловой системы NTFS и утилиты командной строки, позволяющие работать с дисками, разделами, каталогами и файлами. В качестве практического задания был создан файл для компиляции Dos приложения.

Изучены механизмы обмена информаций между приложениями и выявлены их достоинства и недостатки.