МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

ОТЧЕТ

по лабораторной работе №1

по дисциплине

Эксплуатация современных операционных систем

Механизмы ОС Windows

РУКОВОДИТЕЛЬ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сапожников В.О.

19-В-1

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

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

Нижний Новгород 2022

**Цель работы:** получить базовые знания о механизмах Windows.

**План работы**

1. Подготовка к установке ОС Windows
2. Процесс загрузки Windows
3. Стартовое меню
4. Архитектура Windows. Состав и функции основных компонентов
5. Установка, выполнение и удаление приложений
6. Графический интерфейс
7. базовые свойства файловой системы NTFS
8. Командный язык и работа в режиме командной строки
9. Механизмы обмена данными между приложениями.
10. **Подготовка к установке ОС Windows**

**Выбор дистрибутива**

ОС Windows 7 имеет несколько редакций, каждая из которых имеет свои особенности.

*Windows 7 Starter (Начальная)*

Данная редакция предназначена для установки на маломощные машины, так же часто используется в качестве начальной ОС, устанавливаемой сборщиками или производителями. В данной редакции урезано довольно много функционала.

*Windows 7 Home Basic (Домашняя базовая)*

Предназначена для домашнего использования. В данной редакции отсутствуют многие дополнительные мультимедийные возможности и инструменты для профессиональной работы.

*Windows 7 Home Premium (Домашняя расширенная)*

Предназначена для домашнего использования. В отличии от Home Basic редакции имеет Windows Media Center, Windows DVD Maker для записи дисков, полная поддержка Windows Aero, другие полезные функции.

*Windows 7 Professional (Домашняя расширенная)*

Предназначена для использования на предприятиях малого и среднего бизнеса. В данной редакции имеется необходимых функционала для работы бизнес-приложений, администрирования системы, печать по сети, удаленное управление, шифрование файлов системы и т.д., которые реально полезны лишь в условиях промышленной эксплуатации. Данная и все последующие версии Windows 7 могут быть членами домена сети Windows.

*Windows 7 Enterprise (Корпоративная)*

Предназначена для корпоративных клиентов, крупных компаний. Распространяется только по корпоративной лицензии. Имеет множество функций, который используется в бизнес-процессах и не понадобятся при домашнем использовании.

*Windows 7 Ultimate (Максимальная)*

Данная редакция имеет практически все возможности всех остальных вышеперечисленных версий.

При выполнении данной лабораторной работы будет использована ОС **Windows 7 Professional.**

**Дисковая конфигурация**

При включении компьютера начинается процесс, который в итоге загрузит операционную системы из памяти. Ход данного процесса зависит от структуры разделов на жестком диске.

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

1. Структура данных на диске
2. Процесс загрузки ОС из памяти
3. Где начинается и заканчивается раздел.

*Master Boot Record*

MBR являет традиционной структурой управления дисками. Пусть она и является более старой, но всё еще широко распространена из-аз её совместимости с более ранними ОС.

Структура MBR:

* 446 байт – код загрузчика
* 64 байта – таблица основных разделов (Primary)
* 2 байта – сигнатура (подпись). Должна быть 55AАh.

Если сигнатура не равна 55AAh, значит, MBR поврежден.

После выполнения определенных операций, BIOS читает первый сектор накопителя, в котором и находится наша главная загрузочная запись. Выполняется код загрузчика, который просматривает таблицу разделов, и передает управление загрузчику операционной системы, который уже ее и загружает.

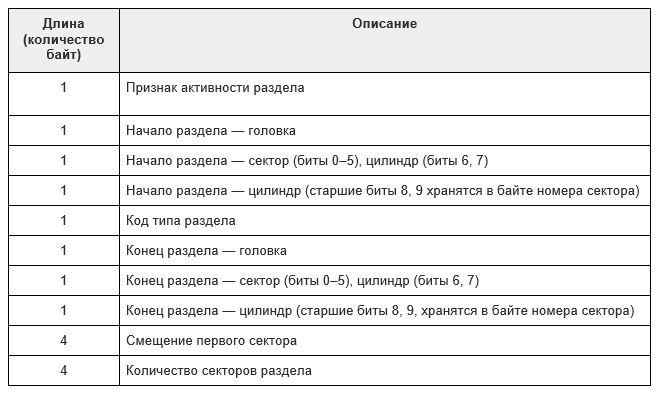
Таблица разделов:



Одна запись — это один раздел, из чего происходит первое ограничение MBR — максимальное число разделов на диске не может быть больше четырех. Но это утверждение правда лишь наполовину: нельзя создать больше четырех основных разделов.  Когда разрабатывалась структура MBR, это считалось достаточным. Однако, позднее был введён *дополнительный раздел*, структура которого (EBR) позволяет создавать внутри него неограниченное число *логических дисков (разделов)*

По правилам дополнительный раздел может быть только один. Таким образом, в максимальной конфигурации MBR на жёстком диске может быть сформировано три основных раздела и один дополнительный. При этом некоторые операционные системы способны загружаться только с основного раздела; а те, которые могут загружаться с дополнительного раздела, вынуждены держать на основном разделе свой загрузчик.

Структура 16-ой битной записи раздела:



Самый первый бит в записи обозначает признак активности раздела — т.е. признак того, что операционную систему следует загружать именно из этого раздела — и может иметь значения 80h (раздел активен) и 00h (раздел не активен).

Плюсы:

* Совместима с большинством ОС

Минусы:

* Допускает только четыре раздела, с возможностью создания дополнительных подразделов на одном из основных разделов
* Ограничивает размер раздела 2ТБ
* Информация о разделе хранится только в одном месте – в главной загрузочной записи. При её повреждении диск становится не читаемым.

*Таблица разделов GUID (GPT)*

GTP – более новый стандарт определения структуры разделов на диске. Для определения структуры используется глобальные уникальные идентификаторы (GUID).

Плюсы:

* Допускает неограниченное кол-во разделов
* Не ограничивает размер раздела
* Хранит копию раздела и загрузочных данных. 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.

Минусы:

* Может быть несовместима со старыми ОС

GPT использует UEFI, который не имеет процедуры хранения в загрузочном секторе первой стадии загрузчика с последующим вызовом второй стадии. 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.

В рамках лабораторной работы удобнее использовать структуру **MBR**.

**Установка и развёртывание. Активация и обновление**

*Варианты установки*

* Установка с использованием внешнего носителя – в BIOS выбирается приоритетный носитель, содержащий загрузочный образ ОС
* Обновление или установка новой ОС из текущей ОС – установка с помощью специальных программ предназначена для обновления ОС. Если платформа не меняется, то чаще всего можно выполнить обновление, при котором старые файлы системы переписываются новыми, а данные пользователя не изменяются.

Развертывание ОС позволяет создать образы ОС и развернуть их на целевых компьютерах. В ходе развертывания ОС возникает ряд последовательных задач, облегчающий процесс развёртывания системы и других пакетов ПО.

*Варианты развёртывания*

* **Развертывание, инициализированное PXE** – позволяет целевым компьютерам запрашивать развертывание по сети. При данном варианте образ ОС и образ загрузки в точку распространения с поддержкой PXE.  
  **Preboot eXecution Enviroment (PXE)** – среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей.
* **Образ ОС доступный в центре ПО** – ОС разворачивается и делается доступней в Центре программного обеспечения. Клиенты Configuration Manager могут инициировать установку ОС из Центра программного обеспечения.
* **Многоуровневое развертывание** – сохраняет пропускную способность сети. Развертывание образа ОС отправляется в точку распространения. Точка в свою очередь развертывает изображение, когда клиентские компьютеры посылают запрос на развертывание.
* **Развертывание мультимедиа для загрузки** – позволяет развертывать ОС после начала работы клиентской машины. При запуске клиентская машина извлекает последовательность задач, образ ОС и любой другой необходимый контент из сети.

Для выполнения работы использовалась не активированная версия ОС Windows 7 Professional. Пробного периода достаточно. Установка происходила при помощи внешнего носителя.

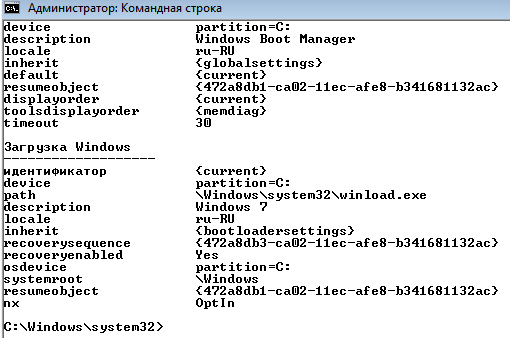
1. **Процесс загрузки Windows**

**Этапы загрузки ОС**

1. Начальный этап загрузки ОС начинается в **BIOS/UEFI**. На данном этапе происходит Power-On Self-Test (POST) и так же может выполняться быстрая проверка памяти (для UEFI). После прохождения POST начинается считывание данных с загрузочных устройств, пока не будет найдена запись с дальнейшими действиями.
2. Initial Startup process – запуск программ, находящихся на загрузочном устройстве. Передача управления загрузчику первого уровня: MBR для BIOS или MBR/GPT для UEFI.
   1. Not system boot start – начальный загрузчик, не связанный с ОС. Анализирует таблицу разделов и загружает раздел, помеченный активным, осуществляется переход на первый раздел активного диска. Передача управления загрузчику второго уровня Partition Boot Sector.
   2. В коде PBS записано имя файла загрузчика ОС, которому передается управление на данном этапе.
   3. Boot manager переходит к загрузке ядра ОС.
3. OS Loader – winload.exe – происходит распознавание устройств, образующих программный профиль.
4. Main Path Boot
   1. На основе выбранного программного профиля начинается загрузка ядра ОС.
   2. Инициализация ядра.
   3. Инициализация менеджера сеансов SMSS. Происходит загрузка среды системы win32 и выполняется работа с реестром: создание доп. файла подкачки, загрузка DLL, инициализация планирования пользовательского режима, создание временных файлов в Windows\Temp, запуск services.exe (системные и помеченные как авто запускаемые службы), lsass.exe – процесс, отвечающий за аутентификацию пользователя и применение политик безопасности.
   4. Win logon Init. Запуск winlogon.exe – процесс, дающий заставку и диалог входа пользователя. происходит инициализация сетевых драйверов, инициализация LSA, запуск служб, отложенных до работы с пользователем. При успешном запуске запись данной конфигурации системы, как последней успешной конфигурации запуска.
   5. Explorer Init. Происходит запуск explorer.exe, DWM, SCM кэширования файловой системы.
5. Этап PostBoot. Запуск служб с отволоженным запуском, запуск процессов автозагрузки. После выполнения всех основных действий освобождаются мощности процессора. Переход в режим “минимальной нагрузки”, где большая часть вычислительных мощностей процессора выделяется на нужды пользователя.

**Boot manager и BCD**

В Windows Vista / 7 / 8 / 10 диспетчер загрузки называется Bootmgr, который читает из системного хранилища Boot Configuration Data (BCD). Доступ к данному хранилищу можно получит при помощи утилиты bcdedit.exe

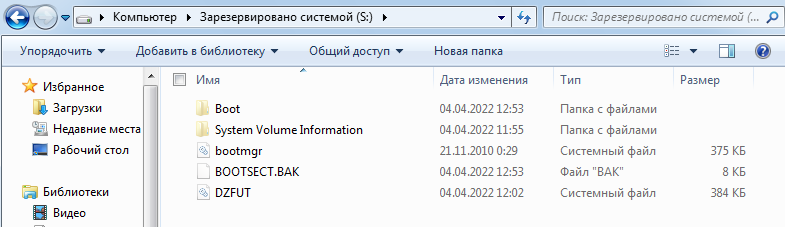


При стандартной инсталляции ОС Windows Vista / 7 / 8 / 10 в начале диска создается дополнительный раздел «Зарезервировано системой». Этот основной раздел имеет статус «Системный» и «Активный» и, следовательно, удалить его тоже нельзя. В Windows 7 размер такого раздела составляет 100 МБ, из которых занято около 30 МБ.

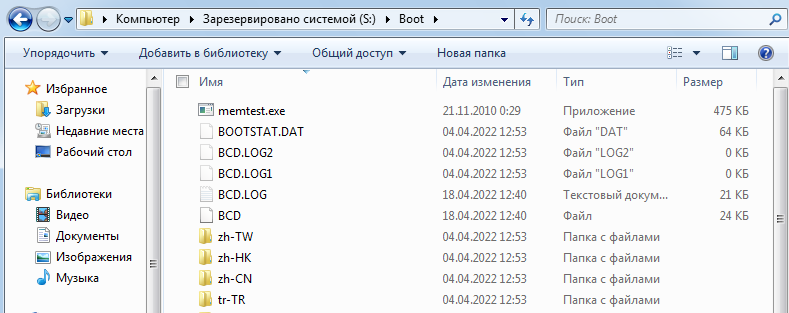


Чтобы просмотреть содержимое данного раздела необходимо назначить ему букву диска и включить отображение скрытых папок и файлов.

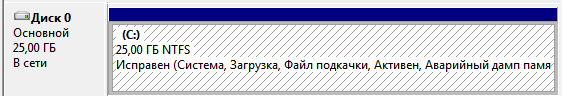
В корне раздела находится файл bootmgr, который является загрузчиком.



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

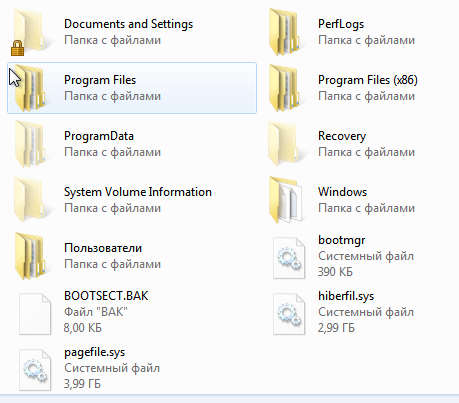


Так же возможна ситуация, когда системный и загрузочные каталоги “склеены” вместе, как и произошло при установке Windows 7 в виртуальной среде VMWare:



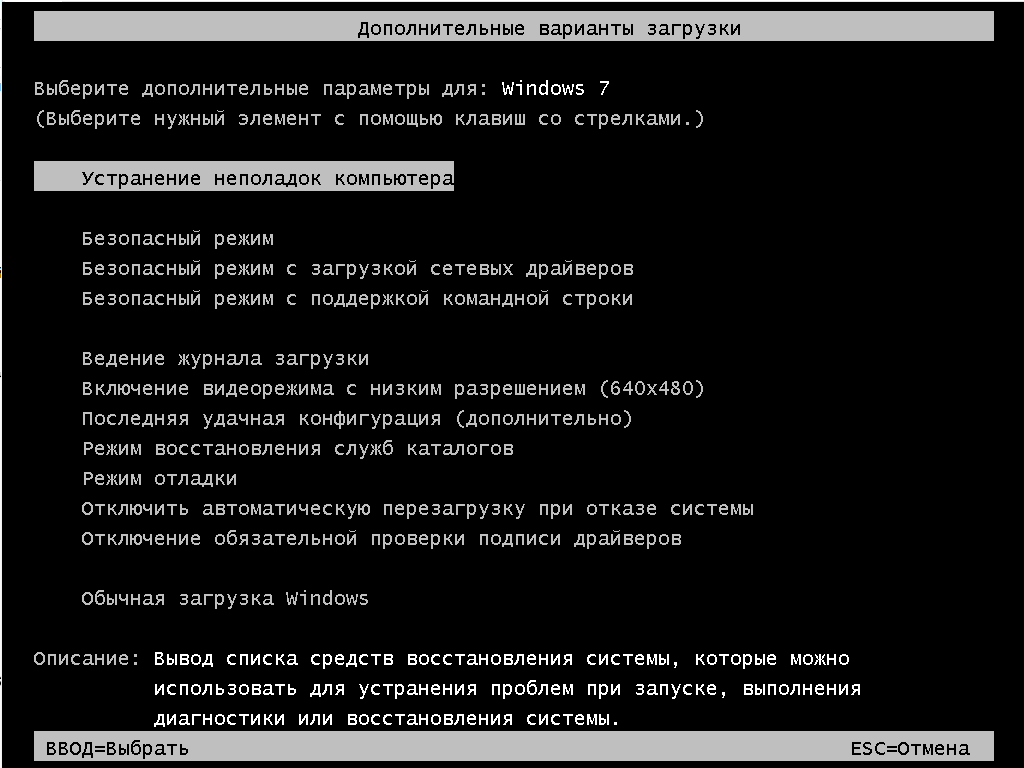
Имеет один раздел диска, который помечен и системным, и загрузочным, и активным.

В данном случае Bootmgr находится в корне данного диска вместе с остальными файлами системы:

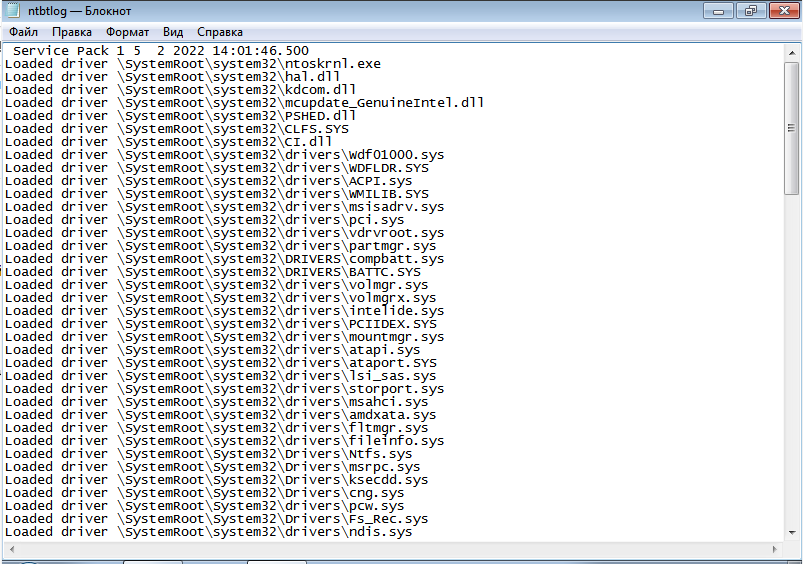


**Загрузка с протоколированием журнала загрузки драйверов**

Чтобы изменить режим загрузки ОС необходимо перейти к меню дополнительных вариантов загрузки до запуска ОС. На Windows 7 это можно сделать, зажав F8.



Выберем пункт «Ведение журнала загрузки». При загрузке в данном режиме будет создан файл C:\Windows\ntbtlog.txt с описание драйверов, загруженных во время запуска.



1. **Стартовое меню**

**Режимы запуска ОС**

С помощью boot manager можно произвести загрузку в разных режимах:

* **Безопасный режим** – запуск Windows только с основами драйверами и службами. Применяется при невозможности загрузки после установки нового устройства или драйвера. так же может быть использован для удаления вирусов.
* **Безопасный режим с загрузкой сетевых драйверов** – запуск Windows с основными драйверами и поддержкой сети.
* **Безопасный режим с поддержкой командной строки** – запуск Windows с основными драйверами, с последующим запуском командной строки, вместо обычного интерфейса Windows.
* **Ведение журнал** – создает ntbtlog.txt, в котором перечислены все драйверы, загруженные во время запуска, включая последний файл, который был загружен перед сбоем.
* **Включение видеорежима с низким разрешением (640\*480)** – запуск Windows в режиме низкого разрешения экрана (640x480). В данном режиме работает стандартный VGA драйвер. Используется для установки/сброса разрешения экрана
* **Последняя удачная конфигурация (дополнительно)** – запуск Windows с использованием параметров последней успешной загрузки. Данный режим используется если безопасный режим не помогает. Возвращает реестр к последнему исправному состоянию. Не всегда помогает, т.к. некоторые изменения могут отразиться и на прошлых настройках.
* **Режим восстановления служб каталогов** – (только для контроллеров домена Active Directory) запуск контролёра домена Windows, на котором исполняется Active Directory, для восстановления службы каталогов.
* **Режим откладки** – включение отладчика ядра Windows. Отладочная информация может быть послана по последовательному кабелю на другую машину с запущенным отладчиком.
* **Отключить автоматическую перезагрузку при отказе системы** – запрет автоматической перезагрузки Windows после сбоя. Позволяет избежать цикла, когда сбой в работе системы вызывает перезапуск и систему продолжают перезагружаться снова и снова.
* **Отключить обязательную проверку подписи драйверов** – разрешает загрузку драйверов, содержащих неверную подпись.
* **Обычная загрузка Widows** – запуск Windows с обычными параметрами

**Сравнение возможностей ОС при загрузке в обычном, безопасном режимах и режиме командной строки.**

Для сравнения возможности ОС необходимо узнать какие драйвера буду загружены в разных режимах и какие службы, процессы будут активны. Состояние загрузки драйверов можно наблюдать в файле ntbtlog. Процессы и службы в графическом режиме можно отобразить при помощи диспетчера задача, в режиме командной строки при помощи команды tasklist для процессов и sc для служб.

*Драйверы*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Драйвер** | **Описание драйвера** | **Обычный 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) – управление 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 | **+** | **-** | **-** |

Минимальный набор драйверов для запуска ОС (Безопасный режим и режим командной строки):

* Драйвер PCI шины
* Драйвера контроллеров SATA для подключения жестких дисков
* Драйвер дисководов компакт дисков (CD, DVD)
* Драйвер контроллеров USB
* Видеодрайвер VGA

При загрузке в обычном режиме так же запускаются драйверы, связанные с сетевыми функциями.

*Службы*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Служба** | **Описание службы** | **Обчный 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 ОС. В безопасно режиме можно просмотреть события, которые привели к сбою системы.
* Поддержка самонастраивающихся устройств – позволяет использовать устройства, подключённые к машине, которые не требуют установки драйверов.
* Удалённый вызов процедур (RPC) – обеспечивает правильную работу COM программ
* Службы криптографии – служба проверки подписи файлов и приложений
* Защитник Windows – служба защиты и ОС
* Сервисы Windows для настройки ОС

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

* Поддержка сетевых адаптеров и устроив, 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 удалённого доступа 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.

*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*

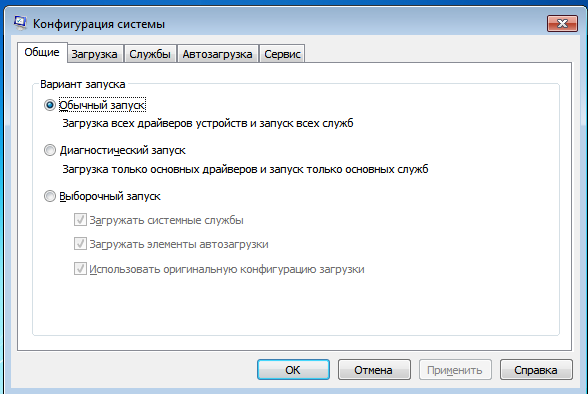
Безопасный режим запускает машину с минимальным необходимым набором дайверов и служб. Системные инструменты ограничены необходимым набором функций, сторонние программы отключены.

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 в режиме консоли обеспечивает большее быстродействие путём отключения графических интерфейсов, но требует четкого знания команд.

**Утилита msconfig**

Msconfig – утилита для управления авто запускаемыми программами и загрузкой Windows. Поставляется вместе с ОС начиная с Windows 98. Она находится в каталоге установки Windows. Её можно запустить из диалогового окна «Выполнить» или через командную строку.

****

*Возможности программы*

На вкладке «Общие» есть возможно выбирать вариант загрузки: обычный, диагностический или выборочный.

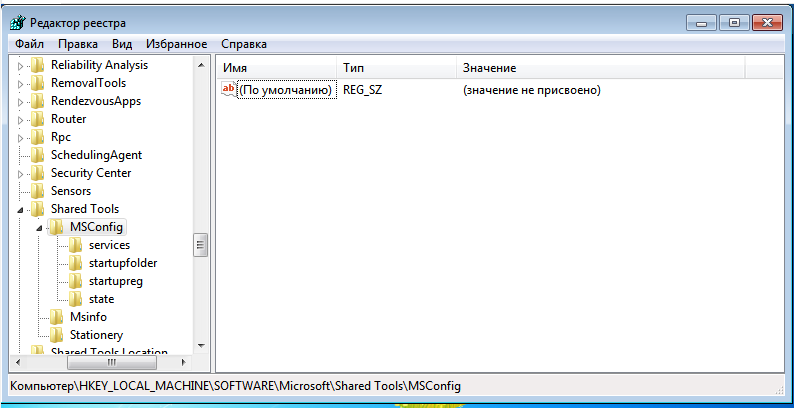
В разделе «Загрузка» можно настроить дополнительные параметры запуска ОС. Например, количество процессоров (ядер), количество памяти, включение/отключение журнала загрузки, наличие графического интерфейса.

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.

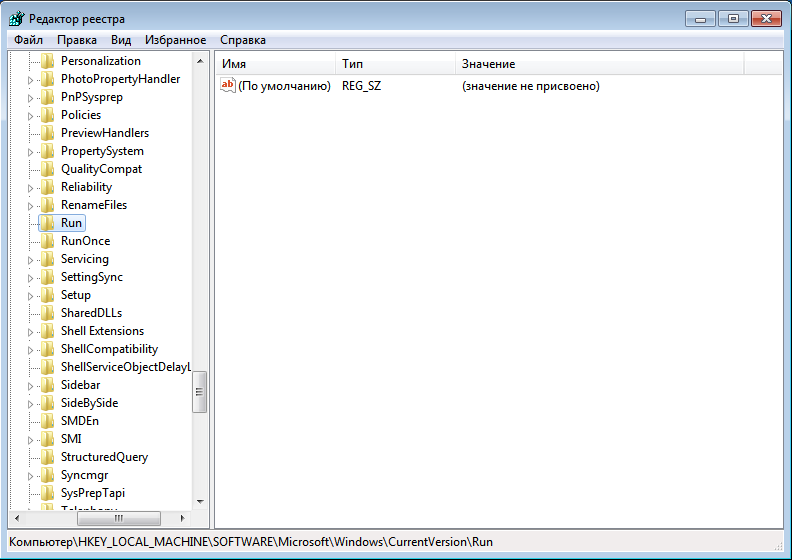
Вкладка «Автозагрузка» отвечает за загрузку приложений (программ) а также определённых служебных утилит, загружаемых не через службы.

*Пути в реестре*

Настройки данной программы храниться 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.

1. **Архитектура Windows. Состав и функции основных компонентов**

**Основные типы файлов**

|  |  |
| --- | --- |
| **.COM** | Простой тип исполняемого файла DOS систем, при котором данные, код и стек находятся в одном 16-битном сегменте. Размер данного файла не может превышать 65280 байт. Com формат является устаревшим, но может найти применение в узких сферах, где могут быть необходимы маленькие программы без настройки, например системы реального времени.  Пример файла: **command.com** |
| **Исполняемые файлы типа MZ с расширением .EXE** | Стандартный формат 16-битных исполняемых файлов с расширением .EXE для DOS. Формат был разработан как замена устаревшему формату .COM. Исполняемые файлы MZ включают метаданные, могут иметь размер больше 64 Кбайт и использовать несколько сегментов памяти различного типа (кода, данных и стека) |
| **Файлы типа PE** | **Portable Executable** (**PE**, «переносимый исполняемый») — формат исполняемых файлов, объектного кода и динамических библиотек (DLL), используемый в 32- и 64-разрядных версиях операционной системы Microsoft Windows. В операционных системах семейства Windows NT формат PE используется для EXE, DLL, SYS (драйверов устройств) и других типов исполняемых файлов.  Пример файла: **powershell.exe** |
| **.SYS** | Расширение драйверов и файла подкачки  Пример файла: **pagefile.sys** |
| **.DLL** | Динамическая библиотека, в которой программы могут хранить функции, переменные и т. д.  Пример файла: **hal.dll** |
| **.BAT** | Текстовый файл в, содержащий последовательность команд, предназначенных для исполнения командным интерпретатором.  Пример файла: **autoexec.bat** |
| **.REG** | Файлы данных реестра. |
| **.INF** | Файл конфигурации в виде текста, разбитого на секции. Содержит информацию об устройствах и файлах для установки или обновления ПО.  Пример файла: **autorun.inf** |
| **.INI** | Файл конфигурации, который содержит данные для инициализации настроек программ. ini файлы так же разбиваются на секции. С введением реестра считаются устаревшим форматом.  Пример файла: **desktop.ini** |
| **.ADM** | Шаблон, используемый Microsoft Group Policies. Описывает, где в системном реестре хранятся настройки групповой политики. Определяет также вид пользовательского интерфейса, который видят администраторы в Group Policy Object Editor при создании или изменении объектов Group Policy (GPO). |
| **.MANIFEST** | Cодержит описание установочного пакета программы для операционной системы Microsoft Windows. Файл может быть сохранен в текстовом [XML](https://open-file.ru/types/xml) формате или собственном формате программы. Файлы манифеста часто используются разработчиками программного обеспечения с целью описания сторонних компонентов и их версий, которые должны присутствовать на компьютере для корректной установки и настройки программы. |

**Основные системные файлы и 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.

Большинство системных файлов операционной системы «Windows» хранятся в каталогах «C:\Windows», «C:\Program Files», «C:\ProgrammData».

**Примеры системных файлов и каталогов**

* **C:\ -** корневой каталог основного раздела ОС Windows. Корневой каталог – каталог, не являющийся подкаталогом ни одного другого каталога (Находится на самом высоком уровне иерархии).  В Windows каждый из логических дисков имеет свой корневой каталог (C:\, D:\ и т. д).
* **bootmgr** – менеджер загрузки, используемый в ОС Windows Vista / 7 / 8 / 10
* **С:\Program Files –** рекомендуемое место, где установленные программы должны хранить исполняемые файлы, данные и другие файлы. Если обратить внимание, то при инсталляции приложений по умолчанию в пути установки указывается именно этот каталог.
* **С:\ProgrammData –** содержит данные и настроки общие для всех пользователей компьютера и доступны каждому из них.
* **С:\pagefile.sys –** файл, используемый системой для хранения частей программ и файлов данных, не помещающихся в оперативной памяти
* **C:\hiberfil.sys –** файл, используемый для “Спящего режима”.
* **C:\Boot\BCD -** системное хранилище Boot Configuration Data, к которому обращается **bootmgr**.
* **C:\Windows –** каталог предназначенный для хранения файлов ОС. В нем находятся прочие важные подкаталоги: Temp, assembly, System32, SysWOW64 и тд.
* **C:\Windows\System32**
  + **winload.exe –** системный загрузчик ОС
  + **services**.**exe –** исполняемый файл, отвечающий за запуск системных и помеченные как авто запускаемые служб при загрузке ОС
  + **winlogon.exe –** исполняемый файл, отвечающий за вход в систему

1. **Установка и удаление приложений.**

**DOS – приложения**

Для каждого приложения DOS ОС создает отдельную Virtual DOS Machine (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.

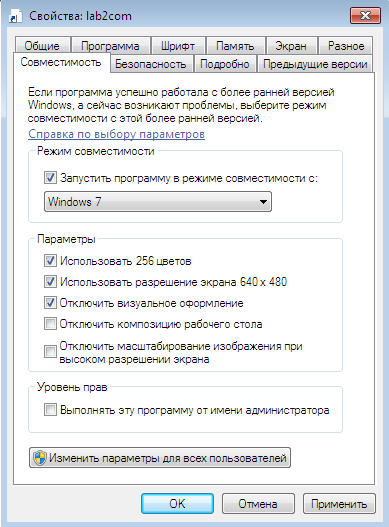
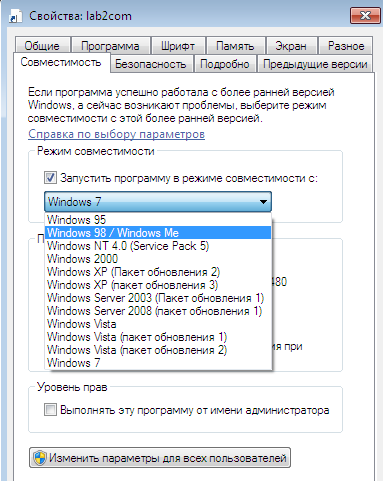
В каталоге **С:\Windows** находиться **\_default.pif** файл, который является стандартным ярлыком для DOS программ. Если для программы не создан отдельный ярлык, то она запускается с параметрами, указанными в этом файле.

Для изменений .pif файлов используется PIFEditor, который устанавливается вместе с ОС. Для вызова редактора необходимо обратится к свойствам данного файла. Сделать это можно, как и у всех файлов при помощи ПКМ →свойства или сочетания клавиш Alt + Enter. Поскольку \_**default.pif** файл является системным и не может быть отредактирован, рассмотрим свойства **.pif** на примере **lab2com.pif.** Данный файл является “настройкой” DOS программы, написанной в рамках второй лабораторной работы по дисциплине «Принципы и методы организации системных программных средств» первого семестра третьего курса.

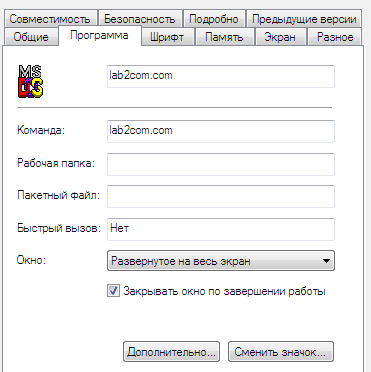
Меню свойств у PIF файлов обширнее остальных.

В меню совместимости мы можем указать с какой версией ОС мы хотим иметь совместимость, разрешение экрана, использование 256 цветов, визуального оформления.

В окне Программа можно указать команду, которая будет выполнена при

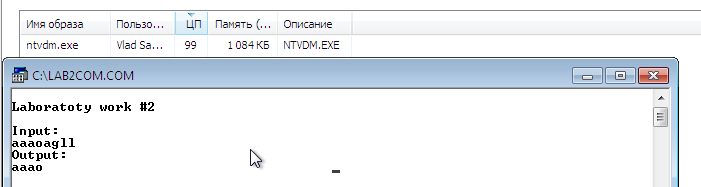


В окне Программа можно указать команду, которая будет выполнена при запуске данного **.pfi** файла. Так же в свойствах можно указать будет ли программа выполняться в окном или полноэкранном режиме, задать настройки памяти и тд.

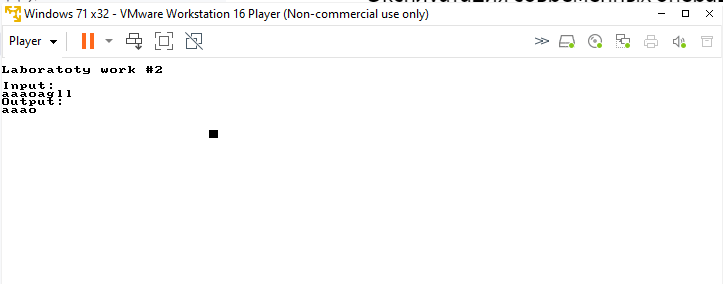


Поскольку программа **lab2com.com** является маленькой и выполняет простые операции по пересылке слов, в которых число сброшенных битов больше половины, то самым явным изменением в её работе будет запуск в полноэкранном режиме.

Запуск **lab2com.com** с настройками по умолчанию: \_**default.pif.** Программа будет выполняться в окно режиме. Чтобы, убедится в том, что запущена **VDM, запустим** диспетчер задач. Процесс **ntydm.exe** является процессом виртуальной машины.



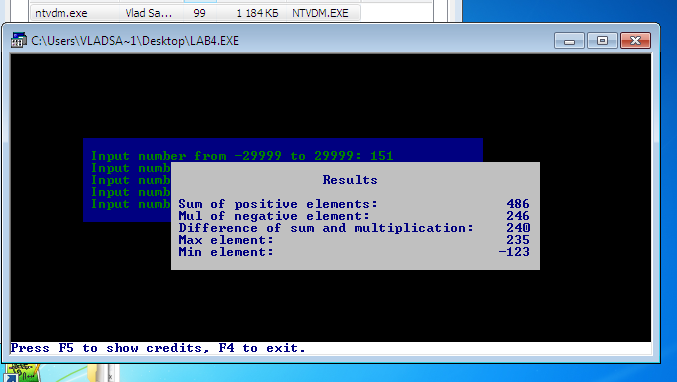
Запуск **lab2com.com** с пользовательскими настройками: **lab2com.pif**. Программа выполняется в полноэкранном режиме



**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.

****

**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 инсталлятора**

* InstallShiled
* Windows Installer Service (\*.msi)
* InstallShiled с MSI
* Inno Setup
* WISE Installer

**Установка и удаление развитого windows приложения**

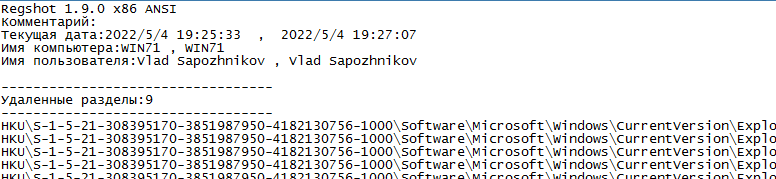
Установим **Google Chrome** для 32битной Windows и определим, какие изменения вносятся в файловую системы и реестр. Для проверки состояния реестра используем **Regshot**, для отслеживания состояния файловой системы **FolderChanhesView**.

**Regshot** делает “снимки” состояния реестра и умеет их сравнивать.

**FolderChanhesView** отслеживает изменения файловой системы в фоновом режиме.

После установки Google Chrome в реестре произошли следующие изменения:

Изменения в реестре



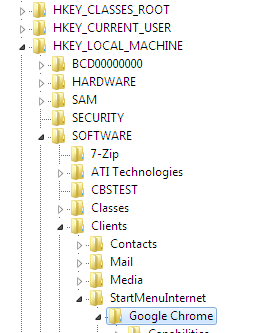
* Было удалено 9 разделов
* Добавлено 355 новых разделов
* Удалено 743 параметра
* Добавлено 638 параметра

Было добавлено множество ключей в разделе \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes, который содержит параметры по умолчанию, применяемые ко всем пользователям

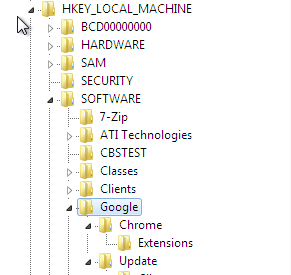
\HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\CLSID {CLSID} и \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\AppID {AppID\_GUID}, которые глобальными и групповыми идентификаторы, используемые обработчиком событий COM.

Добавлены ключи вида: \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\GoogleUpdate – отвечающие за параметры проверки и установки обновлений

В раздел, отвечающий за регистрацию интернет клиента \HKEY\_LOCAL\_MACHINE\SOFTWARE\Clients\StartMenuInterface добавлен подраздел GoogleChrome – позволяет определить GoogleChrome как интернет-браузер.

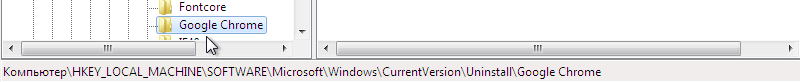


В разделе \HKEY\_LOCAL\_MACHINE\SOFTWARE появился подраздел Google, где хранится служебная информация, например: версия клиента и установленные расширения

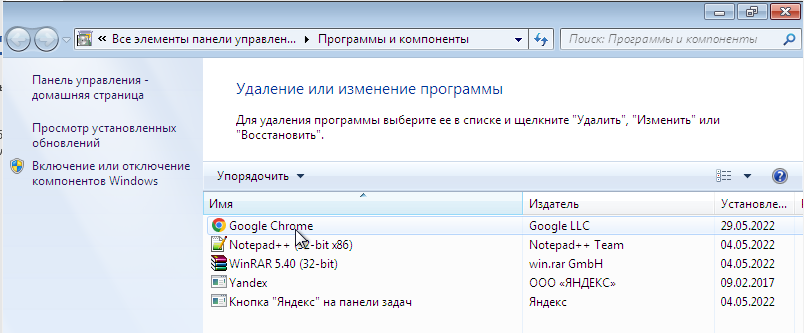


В раздел \HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\

CurrentVersion\Uninstall\Google Chrome – информация для удаления.

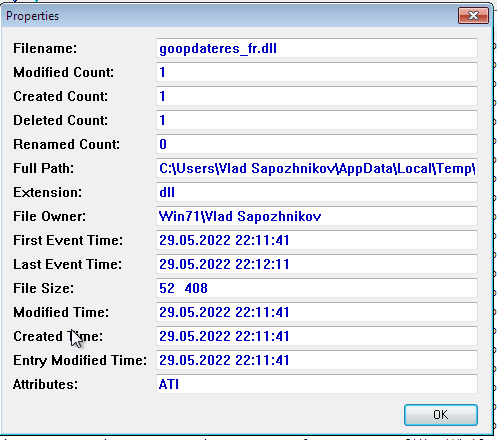


Благодаря данному подразделу google Chrome теперь отображается в «Удаление программ»

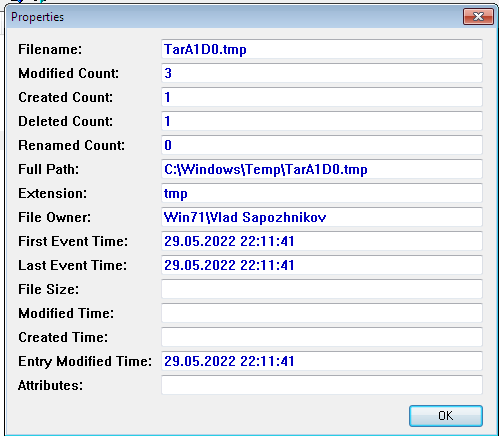


Изменения в файловой системе

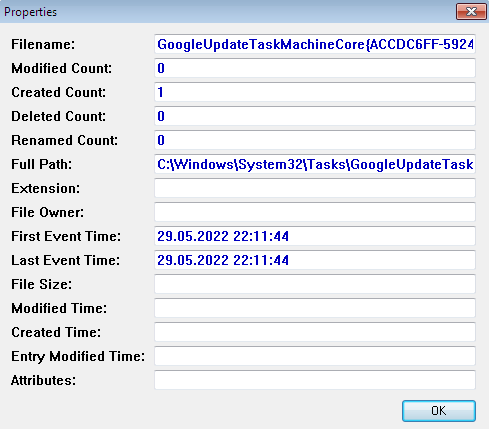
В каталоге **C:\Users\<имя\_пользователя>\AppData\Local\Temp** было добавлено множество файлов динамических библиотек .dll. Пример данного файла:



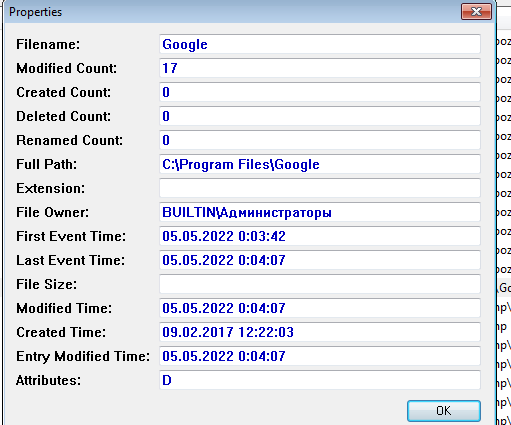
В каталоге **C:\Windows\Temp** добавлены временные файлы с расширением .tmp:



В каталоге **C:\Windows\System32\Tasks** появились две новые задачи: GoogleUpdateTaskMachineUA и GoogleUpdateTaskMachineCore, отвечающих за проверку и установку обновлений:



В файловой системе появился новый каталог **C:\ProgramFiles\Google,** который так же несет в себе несколько подкаталогов, множество **.dll,** сжатых **.pak** и временных **.temp** файлов.

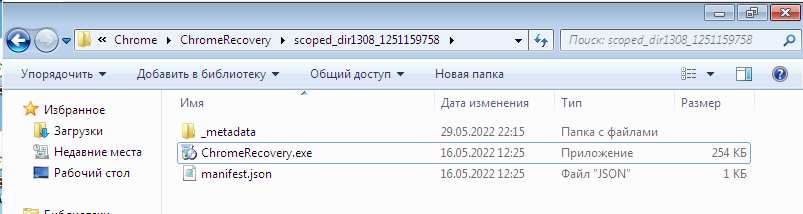


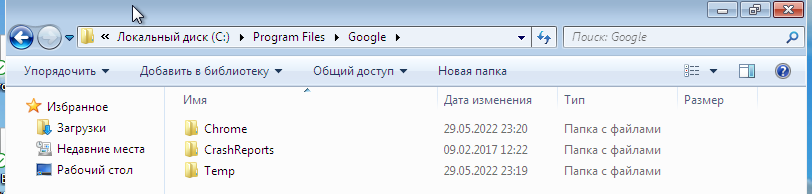
Удаление программы

Теперь выполним удаление **Google Chrome** без удаления данных о пользователе и сравним состояния реестра и файловой системы.

Изменения в файловой системе

В файловой системе остался каталог **C:\ProgramFiles\Google,** который теперь содержит утилиту для восстановления ChromeRecovery



И пустые подкаталоги временных файлов и отчетов об ошибках 

Из каталогов **C:\Users\<имя\_пользователя>\AppData\Local\Temp** и **C:\Windows\Temp** были удалены временные файлы и файлы динамических библиотек .dll.

Из каталоге **C:\Windows\System32\Tasks** были удалены две задачи: GoogleUpdateTaskMachineUA и GoogleUpdateTaskMachineCore, отвечающие за проверку и установку обновлений.

Изменения в реестре

Было удалено множество ключей в разделе \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes, который содержит параметры по умолчанию, применяемые ко всем пользователям

В разделах \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\CLSID {CLSID} и \HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\AppID {AppID} были удалены идентификаторы, связанные с Google Chrome

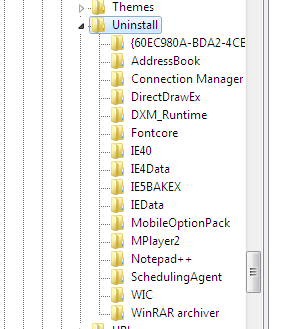
Удалены записи вида HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\GoogleUpdate – отвечающие за параметры проверки и установки обновлений

Подраздел GoogleChrome был удален из \HKEY\_LOCAL\_MACHINE\SOFTWARE\Clients\StartMenuInterface. Теперь Google Chrome не определяется как интернет-браузер (что верно, ведь такого приложения в системе больше нет)

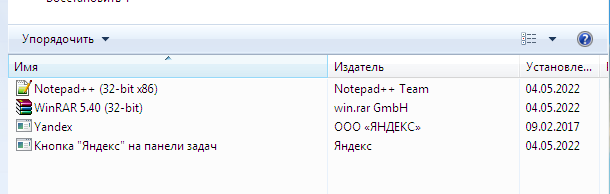


В разделе \HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\

CurrentVersion\Uninstall\ был удален подраздел связанный с Google chrome

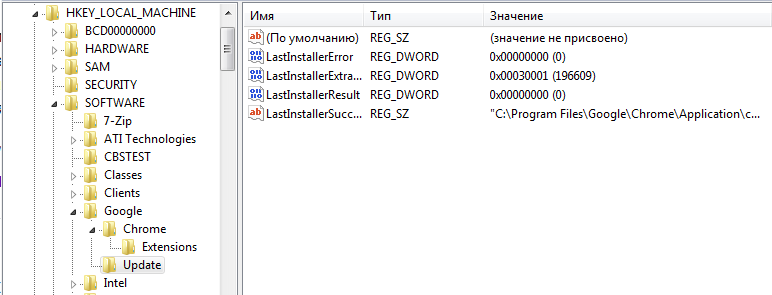


Благодаря чему Google Chrome был удален из списка «Удаление программ»



Однако не записи о GoogleChrome были удалены.

Например, в разделе реестра: HKLM\SOFTWARE остался раздел Google, в котором хранятся данные об расширениях и обновлениях.



1. **Графический интерфейс Windows.**

Графический интерфейс в Windows реализуется с помощью функций модуля Graphical Device Interface (GDI). предоставляет пользователю возможность удобной работы в ОС. Графическая подсистема обеспечивает поддержку графики для устройств ввода/вывода.

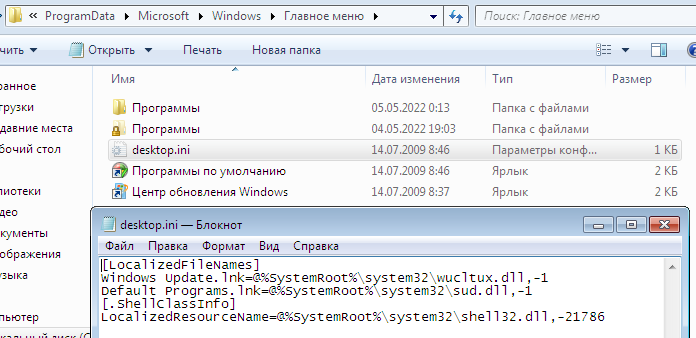
Рабочий стол поддерживает механизм Object Link and Embedding (OLE), имеется возможность размещения на столе как самих объектов (программ, рисунков, текстовых файлов), так и ярлыков объектов. Windows позволяет просматривать рабочий стол, а также файлы и папки с применением различных стилей.

**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.

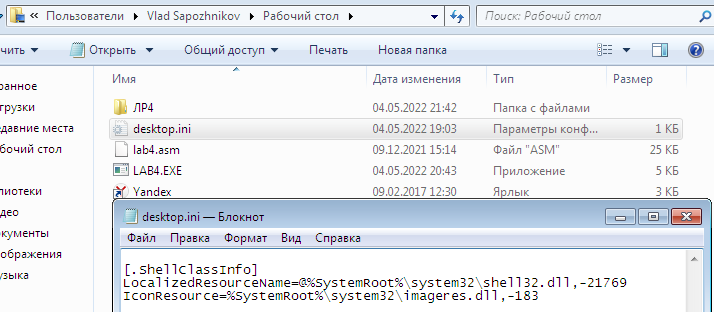
Каталог главного меню - C:\ProgramData\Microsoft\Windows\Start Menu. В подкаталоге «Programs» отображается список установленных программ. В файле desktop.ini находятся пути, на которое ссылаются ярлыки.



**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.

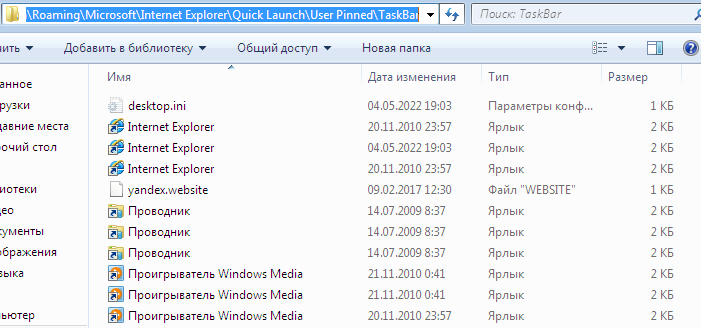


**Панель задач**

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.



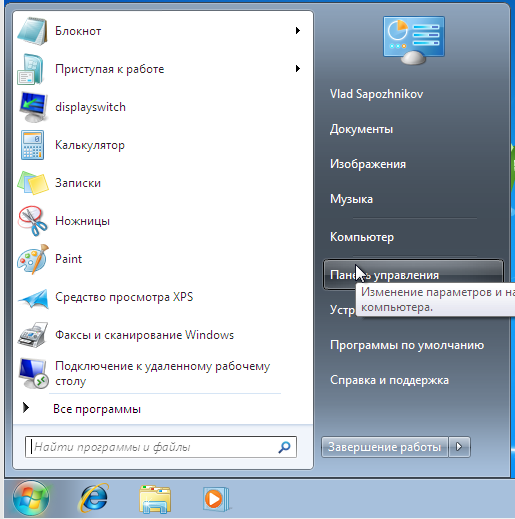
Программы, настройки времени, сети и языка находятся в каталоге C:\Windows и подкаталоге \System32.

**Меню пуск**

Меню «Пуск» является основным связующим звеном с программами, папками и параметрами компьютера. Оно называется *меню,* потому что содержит список для выбора. И как подразумевает слово «Пуск», это меню часто является местом для открытия или запуска элементов.

Меню «Пуск» используется для следующих задач:

* Запуск программ
* Открытие наиболее часто используемых папок
* Поиск файлов, папок и программ
* Настройка параметров компьютера
* Получение справки по работе с ОС Windows
* Выключение компьютера
* Завершения сеанса пользователя в Windows или переключение к учетной записи другого пользователя

****

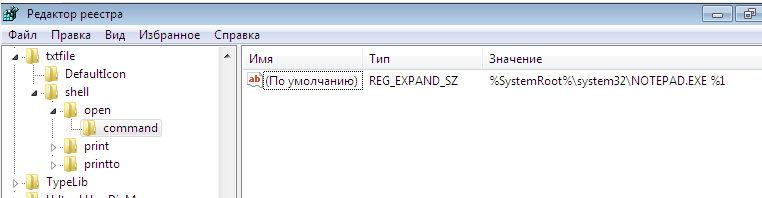
**Контекстное меню**

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.

1. **Базовые свойства файловой системы NTFS.**

Файловая система NTFS - Улучшенная файловая система, обеспечивающая уровень быстродействия и безопасности, а также дополнительные возможности, недоступные ни в одной версии файловой системы FAT. В отличие от FAT, NTFS поддерживает разделы большего объема, б*о*льшие файлы и количество файлов, хранящихся в разделе. Кроме того, в NTFS используются также кластеры меньшего размера, чем в FAT 32, что позволяет более эффективно использовать дисковое пространство. Например, в разделе NTFS объемом 30 GiB используются кластеры размером 4 KiB, тогда как в разделе такого же объема, отформатированном в FAT 32, исполь­зуются кластеры размером 16 KiB. Использование кластеров меньшего размера позволяет снизить потери дискового пространства.

**Размеры кластеров NTFS**

|  |  |  |
| --- | --- | --- |
| **Размер кластера** | **Количество секторов в каждом** | **Размер кластера** |
| От 16 до 512 MB | 1 | 0,5 KB |
| От 512 MB до 1 GB | 2 | 1 KB |
| От 1 до 2 GB | 4 | 2 KB |
| От 2 GB до 2 TB\* | 8 | 4 KB |

В NTFS используется специальная файловая структура, которая называется главной файло­вой таблицей (Master File Table — MFT), и файлы метаданных. В сущности, MFT представляет собой реляционную базу данных, состоящую из строк и столбцов, в которых содержатся записи и атрибуты файлов. Эта таблица содержит записи практически всех файлов, расположенных в разделе NTFS. Файловая система NTFS создает записи файлов и каталогов для каждого файла или каталога, созданного в разделе NTFS. Эти записи хранятся в MFT, причем каждая из них занимает 1 KB. Записи файлов содержат данные о местоположении записи в MFT, а также ат­рибуты файлов и другую информацию, относящуюся к этим файлам.

Файловая система NTFS была предназначена для управления кластерами с помощью 64-раз­рядных чисел, представляющих собой астрономические величины, но в существующих версиях используются только 32-разрядные числа. Использование 32-разрядных чисел позволяет обес­печить адресацию до 4 294 967 295 кластеров, каждый из которых обычно занимает 4 KiB.

В общей сложности NTFS резервирует 32 сектора, 16 из которых занимает созданный по умолчанию загрузочный сектор раздела, а следующие 16 — его резервная копия. Загрузочный сектор раздела, созданный по умолчанию, размещается в начале раздела (в логическом секто­ре 0), тогда как его резервная копия записывается либо в логическом центре (если раздел был отформатирован с помощью NT 3.51 или более ранней версии), либо в конце тома раздела (если последний был отформатирован с помощью операционной системы NT 4.0 или более поздней версии).

Как уже отмечалось, 32-разрядная нумерация секторов в таблицах разделов на дисках MBR не позволяет создавать основные диски NTFS емкостью более 2 TB. Тем не менее ис­пользование нескольких составных или чередующихся накопителей дает возможность созда­вать разделы NTFS более высокой емкости и, следовательно, динамические диски большего размера. Поскольку управление динамическими разделами осуществляется с помощью скры­той базы данных, то на них совершенно не влияет ограничение в 2,2 Тбайт, которое налагается таблицами разделов главной загрузочной записи.

**Характеристики NTFS**

* Размер файлов не превышает 16 ТиБ (без 64 КиБ) или ограничен размером раздела меньшей емкости. NTFS поддерживает до 4 294 967 295 файлов в разделе.
* Обычно NTFS не используется на сменных носителях, так как не позволяет сразу же передать данные на жесткий диск. Кроме того, отключение носителей, отформатиро­ванных в NTFS, без использования приложения Safe Removal может привести к потере данных. Для сменных носителей, которые могут быть неожиданно отключены, лучше использовать файловые системы FAT 12, FAT 16 или FAT 32.
* NTFS поддерживает регистрацию транзакций и функции восстановления данных. В случае отказа в работе NTFS восстанавливает после перезагрузки целостность фай­ловой системы, используя для этого системный журнал и данные контрольных точек.
* NTFS динамически перераспределяет кластеры, содержащиеся в сбойных секторах, и отмечает дефектные кластеры как поврежденные, что препятствует их дальнейшему использованию.
* NTFS имеет встроенные средства защиты, которые дают возможность устанавли­вать разрешения для каждого файла или каталога.
* NTFS имеет встроенную файловую систему кодирования (EFS). Эта система проводит динамическое кодирование и декодирование в процессе работы с зашифрованными файлами или папками, не позволяя другим пользователям обращаться к этим файлам.
* NTFS предоставляет возможность указывать дисковые квоты. Существует возмож­ность отслеживать и контролировать использование различными пользователями дискового пространства в разделах NTFS.
* NTFS имеет функцию встроенного динамического сжатия, которая позволяет сжи­мать и разархивировать файлы по мере их использования.
* Домены, которые входят в состав [Active Directory](ms-its:C:\WINDOWS\Help\misc.chm::/ntfs.htmHELP=glossary.hlp%20TOPIC=gls_activedirectory) и позволяют настраивать средства безопасности, не усложняя задачи администрирования. Для работы [контроллеров домена](ms-its:C:\WINDOWS\Help\misc.chm::/ntfs.htmHELP=glossary.hlp%20TOPIC=gls_domain_controller) необходимо использовать файловую систему NTFS.
* Разреженные файлы. Это файлы очень больших размеров, создаваемые в приложениях таким образом, что для их хранения требуется немного места на диске. В связи с этим NTFS выделяет дисковое пространство только для тех частей файла, в которые производится запись данных
* Служба внешнего хранилища, позволяющая расширить дисковое пространство путем организации удобного доступа к съемным носителям, таким, как магнитные ленты.

**Архитектура NTFS**

Несмотря на существующие различия в структуре раздела файловых систем FAT и NTFS, они имеют подобные элементы, например загрузочную область. Раздел NTFS состоит из *главной таблицы файлов (master file table — MFT).* Однако MFT — это не то же самое, что FAT. Вместо использования таблицы со ссылками на кластеры, MFT содержит больше информации о файлах и каталогах в разделе. В некоторых случаях MFT может даже содержать файлы и каталоги.

При организации раздела NTFS система создает 10 системных файлов NTFS:

* $mft (Master File Table (MFT)) – Содержит запись для каждого файла в разделе NTFS в его атрибуте Data.
* *$mftmirr* (Master File Table2 (MFT2)) - Зеркальная копия MFT, используемая для восстановления.
* $badclus (Файл поврежденных секторов) - Содержит все поврежденные секторы раздела.
* *$bitmap* (Карта распределения кластеров) - Содержит карту всего раздела, указывающую на занятые кластеры
* *$boot* (Загрузочный файл)- Содержит загрузочную информацию (если раздел загрузочный)
* $attrdef (Таблица определения атрибутов) - Содержит определение всех системных и пользовательских атрибутов раздела.
* $logfile (Файл журнала)- Представляет собой файл журнала транзакций, используемый для восстановления
* $quota (Таблица квот) - Представляет собой таблицу квот пользователей на данном разделе (используется только в NTFS 5).
* $upcase (Таблица символов) - Используется для преобразования символов верхнего и нижнего регистров в символы верхнего регистра Unicode.
* $volume (Раздел) - Содержит информацию о разделе, например имя раздела и версию.
* $extend (Файл расширения NTFS) - Используется для хранения дополнительных расширений, таких, как квоты, идентификаторы объектов и параметры точек монтирования

Первая запись в MFT называется *дескриптором* (*descriptor*) и содержит информацию о расположении самой MFT. Загрузочный сектор в разделе NTFS содержит ссылку на распо­ложение записи дескриптора.

Вторая запись в MFT — это копия дескриптора.

Третья запись — это запись файла журнала. Все операции (транзакции) в NTFS записы­ваются в специальный файл журнала, что позволяет восстановить данные после сбоя. Осталь­ная часть MFT состоит из записей для файлов и каталогов, которые хранятся в разделе. В файле NTFS хранятся атрибуты, определенные пользователем и системой. Атрибуты в раз­деле NTFS — это не простые флаги из раздела FAT. Вся информация о файле, т.е. атрибуты, в NTFS сохраняется вместе с файлом и является частью самого файла. Каталоги в NTFS со стоят в основном из индексов файлов в этом каталоге и не содержат такой информации о файле, как размер, дата, время и др.

Таким образом, MFT — это не просто список кластеров, это основная структура хранения данных в разделе. Если файл или каталог относительно небольшой (около 1 500 байт), его за­пись может храниться в MFT. Для больших массивов данных в MFT помещается указатель на файл или каталог, а сами данные располагаются в других кластерах в разделе. Эти кластеры называются *экстентами (extents).* Все записи в MFT, включая дескрипторы и файл журнала, могут использовать экстенты для хранения дополнительных атрибутов. Атрибуты файла, ко­торые являются частью записи MFT, называются *резидентными (resident),* а атрибуты, рас­положенные в экстентах, — *нерезидентными (nonresident).*

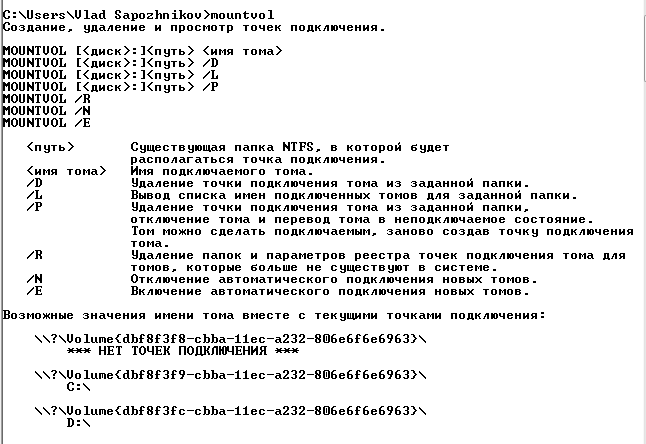
**Монтирование тома к точке соединения NTFS.**

**Создание Junction Point**

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:

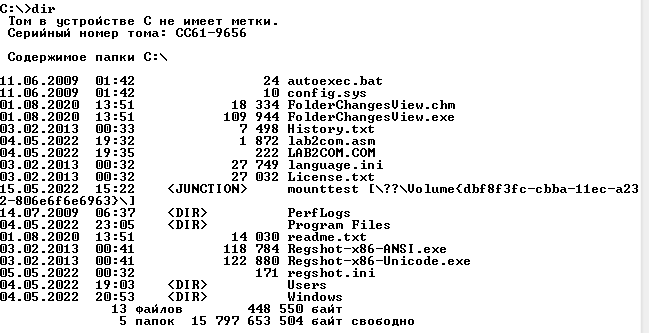


* mountvol [<диск>:]<путь> <имя тома> - подключить том с указным именем в качестве папки по указанному пути



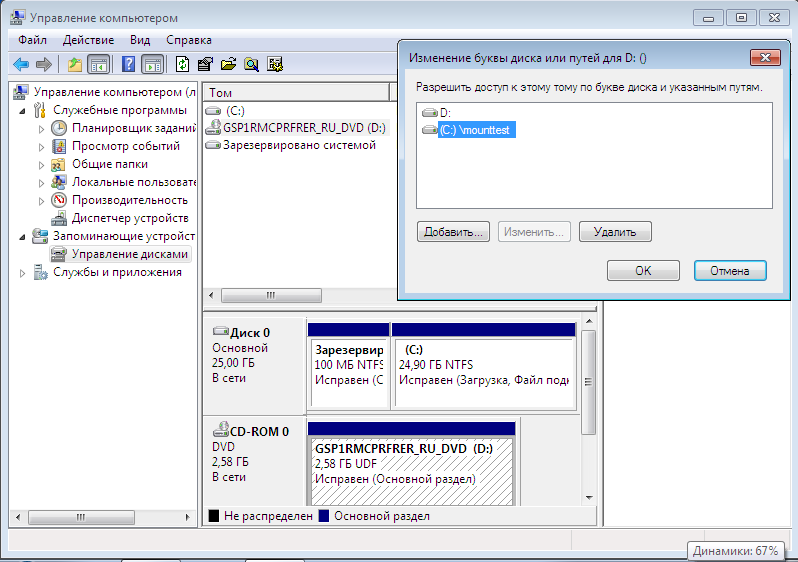
* mountvol [<диск>:]<путь> /l - отобразить список подключенных томов для папки по указанному пути





Как можно заметить была создана точка соединения (Junction Point) NTFS, находящаяся на диске C:\ и указывающая на диск D:\

Так же это можно увидеть и в «Управлении дисками»



**Создание Hard link и Sym Link.**

* **Hard Links** (жесткие ссылки) – могут указывать только на локальный файл, но не на папку. Такой файл – это ссылка на другой файла на этом же диске без фактического дублирования самого файла. У него отображается такой же размер и свойства, как у целевого файла (но реальное место на диске он не занимает);
* **Symbolic Links** (мягкая ссылка, симлинк) – могут указывать на локальный файл, папку и сетевой каталог на удаленном компьютере (UNC), поддерживаются относительные пути.

Использование команды fsutil hardlink create позволяет создать несколько логических файлов в различных местах. При этом все они будут ссылаться на один и тот же физический файл. Использование постоянных ссылок позволяет прозрачно для пользователей перенаправить обращение к файлу в одном месте к данным, фактически хранящимся в другом месте.

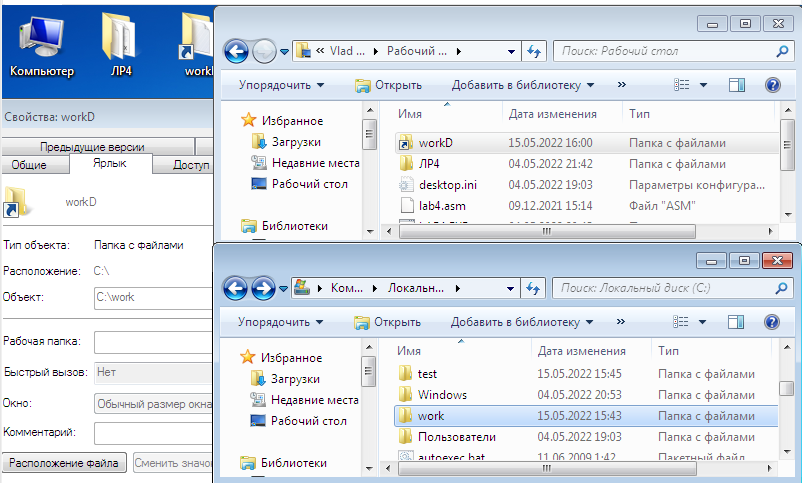
При удалении файла он не удаляется физически до тех пор, пока все жёсткие ссылки, указывающие на него, не будут удалены. Порядок их удаления значения не имеет. При удалении жёсткой ссылки в корзину количество ссылок у файла сохраняется.

При редактировании по одной ссылок изменяется и сам файл, на который указывает ссылка. При перемещении файла ссылка на него не теряется, т.к. связь идет через индексный дескриптор.

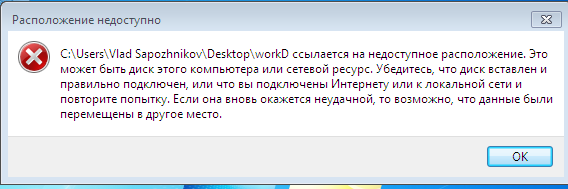


Создать Sym Link можно при помощи команды mklink с флагом /D





При перемещении файла/каталога, на который указывает SymLink сама ссылка не меняется и указывает на недоступное расположение:

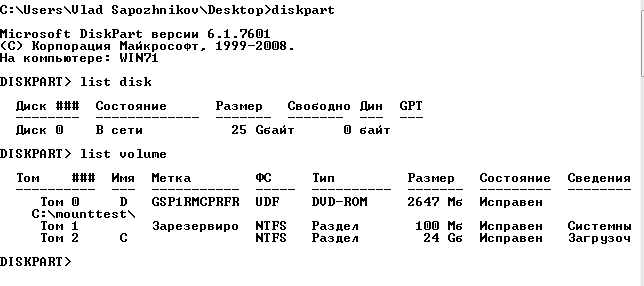


При удалении SymLink, файл на который указывает ссылка не удаляется.

**DiskPart**

Программа DiskPart.exe это работающий в текстовом режиме командный интерпретатор, который позволяет управлять объектами (дисками, разделами или томами) с помощью сценариев или команд, вводимых с командной строки. Перед использованием команд DiskPart.exe для диска, раздела или тома, необходимо сначала составить список объектов, а затем выбрать объект для работы. После этого все введенные команды DiskPart.exe выполняются для данного объекта.

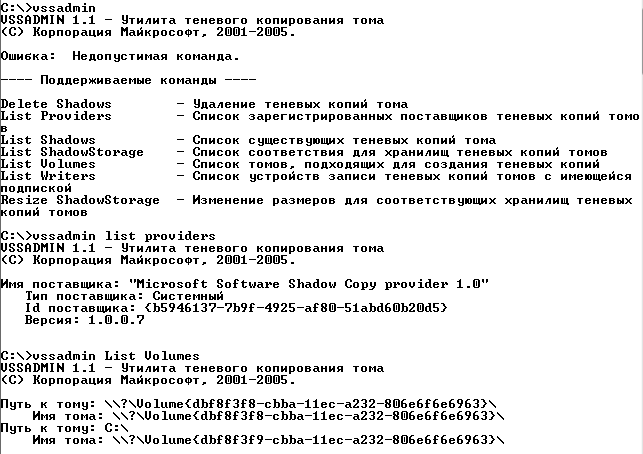
Составить список всех доступных объектов и определить номер объекта или букву диска можно с помощью команд list disk, list volume и list partition. Команды list disk и list volume позволяют вывести все диски и тома компьютера. Команда list partition, однако, выводит разделы только выбранного диска. При использовании команды list рядом с выбранным объектом отображается звездочка (\*). Объект выбирается по номеру или букве диска.



**VssAdmin**

Утилита VSSADMIN.EXE предназначена для администрирования в командной строке службы теневого копирования томов.

Теневое копирование тома (Volume Shadow Copy) – технология, используемая в операционных системах Windows и позволяющая копировать системные и заблокированные файлы, с которыми в данный момент времени ведется работа. Теневое копирование реализовано с использованием специальной службы VSS (Теневое копирование) и системных драйверов для получения снимков томов (Volume Snapshot). Основным назначением теневого копирования является создание системных точек восстановления, архивных образов системы, и архивирования пользовательских данных.

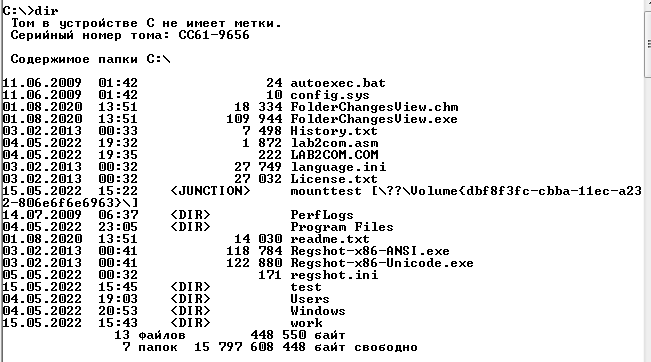


1. **Командный язык и работа в режиме командной строки.**

По своим функциям команды можно разделить на несколько групп: управление файлами и каталогами, управление устройствами, управление процессами, информационные команды ввода-вывода, команды программирования командных файлов, другие.

**Команды работы с каталогами и файлами**

|  |  |
| --- | --- |
| **ATTRIB** | Отображение и изменение атрибутов файлов. |
| **DEL** | Удаление одного или нескольких файлов. |
| **COPY** | Копирование одного или нескольких файлов в другое место. |
| **MKDIR** | Создание папки. |
| **CHDIR** | Вывод имени либо смена текущей папки. |
| **RMDIR** | Удаление папки. |
| **DIR** | Вывод списка файлов и подпапок из указанной папки. |
| **TYPE** | Вывод на экран содержимого текстовых файлов. |



**Команды управления работой с устройствами и процессами**

|  |  |
| --- | --- |
| **BREAK** | Включение/выключение режима обработки комбинации клавиш CTRL+C. |
| **CHKDSK** | Проверка диска и вывод статистики. |
| **CHKNTFS** | Отображение или изменение выполнения проверки диска во время загрузки. |
| **CONVERT** | Преобразование дисковых томов FAT в NTFS. Нельзя выполнить преобразование текущего активного диска. |
| **DISKCOMP** | Сравнение содержимого двух гибких дисков. |
| **FOR** | Запуск указанной команды для каждого из файлов в наборе. |
| **MORE** | Последовательный вывод данных по частям размером в один экран. |
| **VERIFY** | Установка режима проверки правильности записи файлов на диск. |



**Информационные команды и команды реконфигурации**

|  |  |
| --- | --- |
| **MEM** | Вывод сведений о полной и свободной системной памяти. |
| **KEYB** | Настройка клавиатуры на национальный алфавит |
| **MODE** | Отображение статуса и режима работы посимвольных устройств. Команда выполняет множество функций, для примера рассмотрим ее применение для поддержки кодовых страниц. Конфигурирование системных устройств. |
| **SET** | Установка значения глобальной переменной в окружении DOS и отображение окружения. |
| **ECHO** | Вывод сообщений и переключение режима отображения команд на экране. |
| **PATH** | Установка и отображение маршрутов поиска исполняемых файлов |
| **VER** | Установка и отображение соответственно даты. |
| **DATE** | Установка и отображение соответственно даты. |

****

**Команды, используемые для программирования пакетных файлов**

|  |  |
| --- | --- |
| **PAUSE** | Приостановка выполнения пакетного файла и вывод сообщения. |
| **CALL** | Вызов одного пакетного файла из другого. |
| **ENDLOCAL** | Конец локальных изменений среды для пакетного файла. |
| **GOTO** | Передача управления в отмеченную строку пакетного файла. |
| **IF** | Оператор условного выполнения команд в пакетном файле. |
| **REM** | Помещение комментариев в пакетные файлы и файл CONFIG.SYS. |
| **SETLOCAL** | Начало локальных изменений среды для пакетного файла. |
| **SHIFT** | Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла. |

**Пример командного файла**

Данная программы выводит информацию о системе (информация о системе получаем путём обращения к реестру): версия ОС, разрядность ОС, № Сборки, Разрядность среды из которой был запущен данный файл.

@echo off

SetLocal EnableExtensions

call :GetSystemVersion "OSVer" "Core" "Build" "EnvironCore"

echo Версия ОС: %OSVer%

echo Рязрядность ОС: %Core%

echo Сборка: %Build%

echo Разрядность среды запуска Batch: %EnvironCore%

pause

Exit /B

:GetSystemVersion [OSVersion] [OSCore] [OSBuild] [EnvironmentCore]

:: Определить версию ОС

:: %1-исх.Переменная для хранения названия ОС

:: %2-исх.Переменная для хранения разрядности ОС

:: %3-исх.Переменная для хранения версии сборки ОС

:: %4-исх.Переменная, идентифицирующая семейство ОС (9x, NT, Vista)

:: %5-исх.Переменная, идентифицирующая разрядность среды, из-под которой запущен скрипт

Set xOS=x64& If "%PROCESSOR\_ARCHITECTURE%"=="x86" If Not Defined PROCESSOR\_ARCHITEW6432 Set xOS=x32

set %~4=x32& if "%xOS%"=="x64" echo "%PROGRAMFILES%" |>nul find "x86" || set %~4=x64

set "%~2=%xOS%"

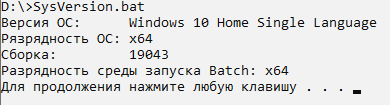
set \_key=HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion

For /f "tokens=2\*" %%a In ('Reg.exe query "%\_key%" /v "CurrentBuildNumber"^|Find "CurrentBuildNumber"') do set "%~3=%%~b"

For /f "tokens=2\*" %%a In ('Reg.exe query "%\_key%" /v "CurrentVersion"^|Find "CurrentVersion"') do set "\_ver=%%~b"

For /f "tokens=2\*" %%a In ('Reg.exe query "%\_key%" /v "ProductName"^|Find "ProductName"') do set "%~1=%%~b"

Exit /B

****

1. **Механизм 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).

**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.

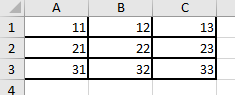
В отличие от операции встраивания, при выполнении операции связывания внутрь документа помещается не сам объект, а только ссылка на него (путь исходного файла, подготовленного приложением-сервером).

Использование операции связывания вместо операции вставки имеет два важных преимущества:

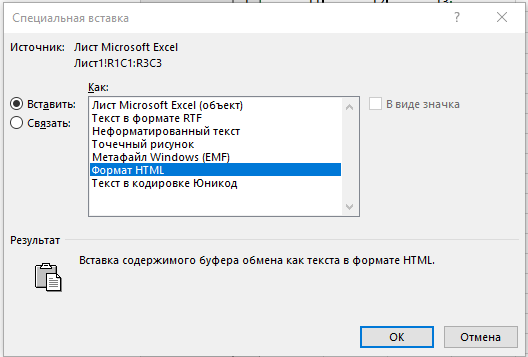
* Если объект, подготовленный приложением-сервером, используется в нескольких документах (приложениях-клиентах), то при изменении объекта, автоматически происходит обновление всех документов, которые содержат ссылку на этот объект.
* Так как в документе помещается только ссылка на связываемый объект,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

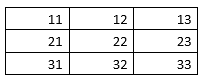
**Связывание и встраивание объектов**

Создадим два файла: текстовый документ Microsoft Word и документ электронных таблиц Microsoft Excel. Внесем данные в Excel документ:



Теперь скопируем данные из Excel и вставим в Word документ. Сделать это можно через меню «Специальная вставка – Формат HTML» или помощи сочетания клавиш Ctrl + V.

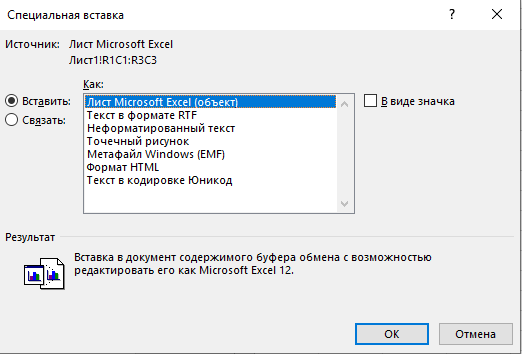


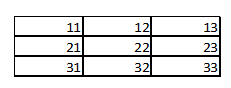


Таким образом мы провели «встраивание» данных из Excel в Word и имеем возможность редактирования их по правилам таблиц Word.

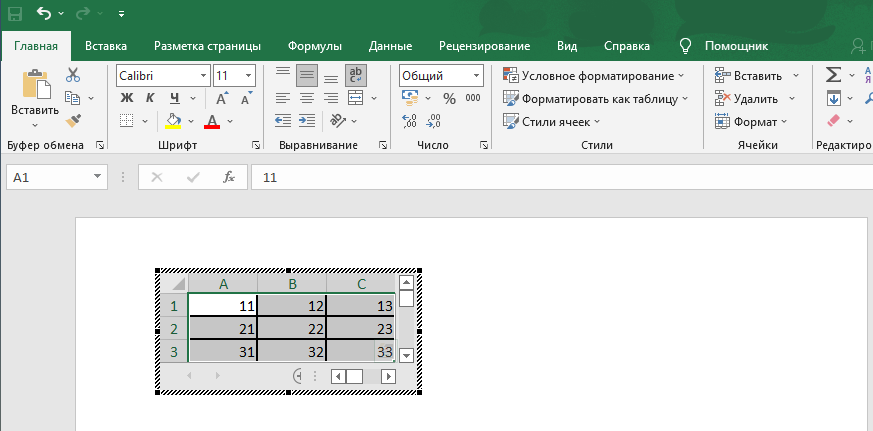
Изменение в Excel никак не влияют на вставленные в Word данные.

Выполним ещё одну вставку, но на этот раз выберем – Вставить как Лист Microsoft Excel (объект)

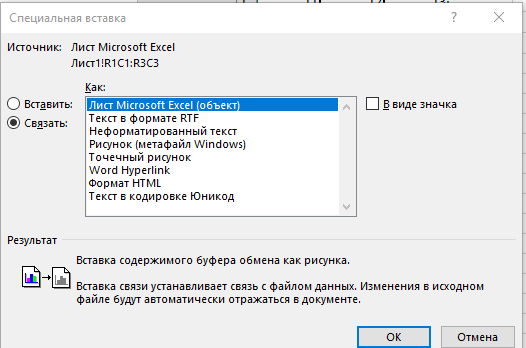


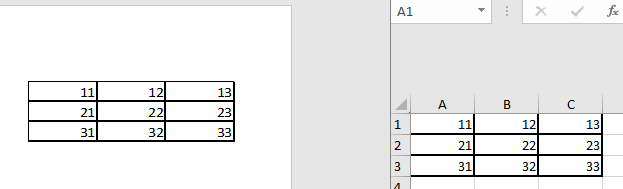


В данном случае данные были вставлены, как объект Excel и мы можем редактировать их как Excel таблицу прямо внутри Word документа

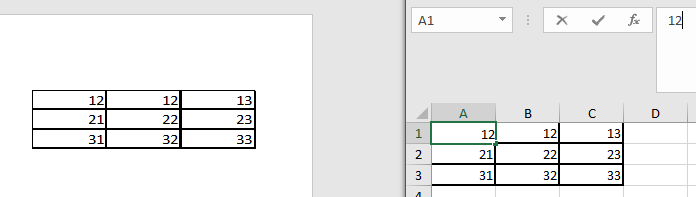


Выполним связывание объектов. Для этого вместо вставки выберем Связывание как Лист Microsoft Excel





В данном случае вставленный объект связан со своим исходным файлом. Все изменения в исходном файле так же отображаются и в файле вставки. Из файла вставки редактировать объект не получится, будет перенаправление в исходный документ.



**Вывод:** 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.