|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Создатели (+что и когда создали, год присуждения премии) | | | | | |
|  | Джон фон Нейман | Венгро-американский математик, сделавший важный вклад в **квантовую физику, квантовую логику, функциональный анализ, теорию множеств, информатику, экономику** и другие отрасли науки. Наиболее известен как **праотец современной архитектуры компьютеров**. Сформулировал 4 принципа устройства современных вычислителей (п. двоичного кодирования, п. адресуемости памяти, п. однородности, п. программного управления). **Создатель теории игр**. | | | |
| \* | Фред Брукс | 1999 – за **исторически значимый вклад** в компьютерную архитектуру, операционные системы, и проектирование программного обеспечения. Всего **9 наград**. | | | Американский учёный в области теории вычислительных систем. **Управлял  разработкой OS/360 в IBM** и написал **книгу** «Мифический человеко-месяц или Как создаются программные системы», наблюдая разработкой. |
| \* | Эдсгер Дейкстра | 1972 – за **Алгол** и публикации | Нидерландский ученый, отстаивавший необходимость **математического подхода к программирован**ию, активно участвовал в разработке языка программирования **Алгол** и написал первый **компилятор Алгол-60**. Проповедовал отказ от использования инструкции GOTO. Также ему принадлежит идея применения «семафоров» для синхронизации процессов в многозадачных системах и **алгоритм нахождения кратчайшего пути на ориентированном графе с неотрицательными весами рёбер**. | | |
| \* | (Дуглас) Даг Энгельбарт | 1997 – за **предвидение будущего развития интерактивных вычислений** и изобретение ключевых технологий, помогающих это предвидение реализовать | | Американский инженер. Один из первых исследователей **человеко-машинного интерфейса** и изобретатель компьютерного манипулятора — **мыши**. Кроме того, Энгельбарт создал: гиперссылки; первую систему обмена текстовыми сообщениями; универсальный пользовательский интерфейс; многооконную систему; архитектуру клиент-сервер | |
| \* | Кен Томпсон | 1983 – за разработку **общей теории операционных систем** и в частности за создание **Unix** | | Создатель **Си** и **Unix**. Так же создал язык **B** (предшественник Си), версию **редактора** **с регулярными выражениями** **QED** для CTSS и **ed** - стандартный редактор Unix | |
| \* | Дэнис Ритчи | Участвовал в создании [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [**BCPL**](https://ru.wikipedia.org/wiki/BCPL), [**B**](https://ru.wikipedia.org/wiki/B_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [**C**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), расширения **ALTRAN** для языка программирования [FORTRAN](https://ru.wikipedia.org/wiki/FORTRAN), участию в разработке [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [**Multics**](https://ru.wikipedia.org/wiki/Multics) и [**Unix**](https://ru.wikipedia.org/wiki/Unix). Соавтор книги **«Язык программирования Си»** | |
|  | Кен Олсен | Основатель компьютерной компании **DEC**, создатель дешевых миникомпьютеров **PDP-1** (Personal Digital Processor) | | | |
|  | Гарри Килдалл | Разработчик о.с. **CP/M** (Control Program for Microcomputers, 1975). CP/M состояла из базовой системы ввода-вывода (BIOS), дисковой ОС (BDOS), командная строка (CCP).  **Код был аппаратно независимым**, поэтому легко портировался | | | |
|  | Тим Паттерсон | Разработчик о.с. **QDOS** (Qick&Dirty Operating System, 1980),  предшественницы MS-DOS | | | |
|  | Джеф Раскин | Специалист по **компьютерным интерфейсам**, автор статей по юзабилити и книги «The Human Interface», сотрудник №31 фирмы Apple Computer, наиболее известен как инициатор проекта **Макинтош** в конце 70-x. Он придумал простой и дружественный **интерфейс Макинтошей** и **однокнопочную мышь**. В 1982 оставил Apple и основал **Information Appliance**. Его первым продуктом стал **SwyftCard**, карта расширения для компьютера Apple II, содержавшая программный пакет SwyftWare. В начале XXI века Раскин начал проект The Human Environment (THE), разработку компьютерного интерфейса, основанную на его тридцатилетних работах и исследованиях в этой области | | | |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Стив Джобс | Американский инженер и предприниматель, сооснователь и директор американской корпорации Apple Inc. Осуществил **переход на Intel** (процессоры). Закрыл убыточные проекты **Apple**. При нем были созданы iPod, iPhone, iTunes Store и MacBookAir и тд. |
|  | Дейв Кэтлер | Создатель **Windows NT** (линейка операционных систем) |
|  | Ричард Столлман | основатель **движения свободного ПО**, проекта GNU, Фонда свободных программ и Лиги за **свободу программирования**. Автор концепции «копилефта», призванной защищать идеалы движения; эту концепцию он с помощью юристов позже воплотил в лицензии GNU General Public License (GNU GPL) для ПО. Создатель текстового редактора **GNU Emacs**, коллекции компиляторов GNU **GCC** и отладчика GNU (**GDB**). |
|  | Эндрю Таненбаум | Профессор Амстердамского свободного университета. Создатель **Minix** (свободная [Unix-подобная](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://en.wikipedia.org/wiki/Unix-like%26amp;sa%3DD%26amp;ust%3D1572899821154000&sa=D&ust=1572899821297000&usg=AFQjCNG8wqdDbZd_f-BlK3SuV_2mUfVwPw) [о.с.](https://www.google.com/url?q=https://www.google.com/url?q%3Dhttp://ru.wikipedia.org/wiki/%2525D0%25259E%2525D0%2525BF%2525D0%2525B5%2525D1%252580%2525D0%2525B0%2525D1%252586%2525D0%2525B8%2525D0%2525BE%2525D0%2525BD%2525D0%2525BD%2525D0%2525B0%2525D1%25258F_%2525D1%252581%2525D0%2525B8%2525D1%252581%2525D1%252582%2525D0%2525B5%2525D0%2525BC%2525D0%2525B0%26amp;sa%3DD%26amp;ust%3D1572899821154000&sa=D&ust=1572899821297000&usg=AFQjCNFIiJzseQAPme3Th86qzHmYx_K0pQ) для студенческих лабораторий) и **RFID-вируса**. Главный разработчик «Amsterdam Compiler Kit» (межплатформенный набор компиляторов и инструментов). **Книги**: "**Компьютерные сети**", "**Операционные системы**: разработка и реализация", "Современные операционные системы", "**Архитектура компьютера**", "Распределённые системы. Принципы и парадигмы" |
|  | Линус Торвальдс | создатель **GNU/Linux** (с 1991) |
|  | Марк Руссинович | американский программист и писатель, один из ведущих специалистов в области архитектуры и дизайна операционных систем, а в частности — внутреннего устройства Windows. В **2006** году вошел в **Top 5 хакеров** планеты. В **1996** году вместе с Брайсом Когсвеллом (разработчик ПО) организовал предприятие **Winternals Software LP**. Они писали утилиты для администрирования и диагностики MS Windows (реже для Linux), которые они распространяли через сайт *sysinternals.com*. Утилиты демонстрировали недостатки Windows, чем заинтересовалась корпорация **Microsoft**. Они **купили Winternals Software**, а Марка взяли в свои ряды в подразделение платформ и служб. |

|  |  |
| --- | --- |
| Принципы устройства современных вычислителей (+примеры нарушений, где они есть) | |
| двоичное кодирование | Информация кодируется с помощью **одного набора кодов**, состоящих из 0 и 1. Нарушение: машинка «Сéтунь», сделанная в СССР, с троичным кодированием. |
| адресуемость памяти (RAM) | Каждая ячейка памяти имеет свой **адрес**, по которому из нее можно достать информацию. |
| программное управление (хранимая в памяти программа) | В процессоре есть регистры, позволяющие общаться с памятью; один из таких регистров – IP (instruction pointer), содержащий адрес текущей ячейки.  Процессор **последовательно выполняет инструкции**, а IP указывает на следующую команду. |
| однородность памяти | И программы (инструкции), и данные хранятся **в одной памяти**, ячейки которой имеют **фиксированный размер**. Нарушение: 1) Кэш L1 имеет разделение памяти на инструкции и на данные; 2) Гарвардская архитектура. |
| наличие пространства ввода/вывода | Некоторые адреса оперативной памяти, известные процессору, на самом деле в нее (RAM) не ведут, а ведут в **контролеры внешних устройств**. |
| аппаратная поддержка стека (LIFO) | (Last In First Out) наличие в процессоре **регистра SP** (stack pointer), указывающего на вершину стека. |
| механизм прерываний  (внешних и внутренних) | Прерывания – сигналы, по которым **управление передается обработчику прерываний** (распознание прерывания, передача управления обработчику, корректное возвращение к прерванной программе). |
| уровни привилегий, привилегированные команды (защищённый режим) | Процессоры должны поддерживать **минимум 2 режима**: обычный и привилегированный. Ядро ОС работает на привилегированном. В архитектуре x86 существует 4 уровня (0, 1, 2, 3), но на практике используется только два, а 2 промежуточных – для драйверов.  В зависимости от уровня в контексте данного потока (процессора) может быть выполнена данная инструкция или нет. Если данную команду нельзя выполнить в данном режиме, то попытка выполнения приведет к прерыванию GR (General Protection). Также некоторые команды могут быть выполнены, но с ограничениями.  Нарушение: RISC и MISC (виды архитектур). |

|  |  |
| --- | --- |
| Типы прерываний (+примеры, происхождение, обработка и применение) | |
| Внутренние (исключения) | Происходят синхронно выполнению программы при появлении ошибки **в ходе исполнения некоторой инструкции программы**. Исключения возникают непосредственно в ходе выполнения тактов команды («внутри» выполнения).  Пример: деление на ноль, ошибка защиты памяти, обращение по несуществующему адресу, выполнение привилегированной инструкции в пользовательском режиме. |
| Внешние (аппаратные) | Данные прерывания являются асинхронными по отношению к потоку инструкций прерываемой программы, т.е. они возникают **между выполнением двух соседних инструкций** и после их обработки система продолжает выполнение процесса уже со следующей инструкции. Внешние прерывания могут возникать в результате **действий пользователя или оператора за терминалом**, а также в результате поступления **сигналов аппаратных устройств** (сигналов завершения операций ввода-вывода, вырабатываемых контроллерами внешних устройств, таких как принтер или накопитель; сигналов от датчиков управляемых компьютером технических объектов). Пример: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши.  События, исходящие от внешних аппаратных устройств, которые могут произойти в любой момент. Факт возникновения такого прерывания трактуется как **запрос на прерывание** – устройства сообщают, что они требуют внимания со стороны ОС. |
| Немаскируемые | Прерывания с **наивысшим п**риоритетом. Они **обрабатываются всегда**, независимо от запретов других прерываний. Их невозможно замаскировать с помощью PIC (Programmable Interrupt Controller – программируемый контроллер прерываний). Данные прерывания **идут сразу в процессор**, минуя PIC.  Генерируются при ошибках RAM и невосстановимых ошибках «железа».  Пример: сбой в микросхеме памяти, синий экран. |
| Программные | Программы могут сами **вызывать прерывания с заданным номером**. Программное прерывание возникает при выполнении особой команды процессора, выполнение которой имитирует прерывание, т.е. переход на новую последовательность инструкций. Программные прерывания происходят **предсказуемо** (в точке программы, где написана данная команда), поэтому они являются асинхронными.  Программные прерывания в прямом смысле прерываниями не являются – это специфический способ вызова процедур (не по адресу, а по номеру в таблице). |

Маскируемые — прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний (в [x86](https://ru.wikipedia.org/wiki/X86)-процессорах — сбросом флага IF в [регистре флагов](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80_%D1%84%D0%BB%D0%B0%D0%B3%D0%BE%D0%B2));

Маскируемые – можно заблокировать.

Аппаратные – происходят вне зависимости от разработчика.

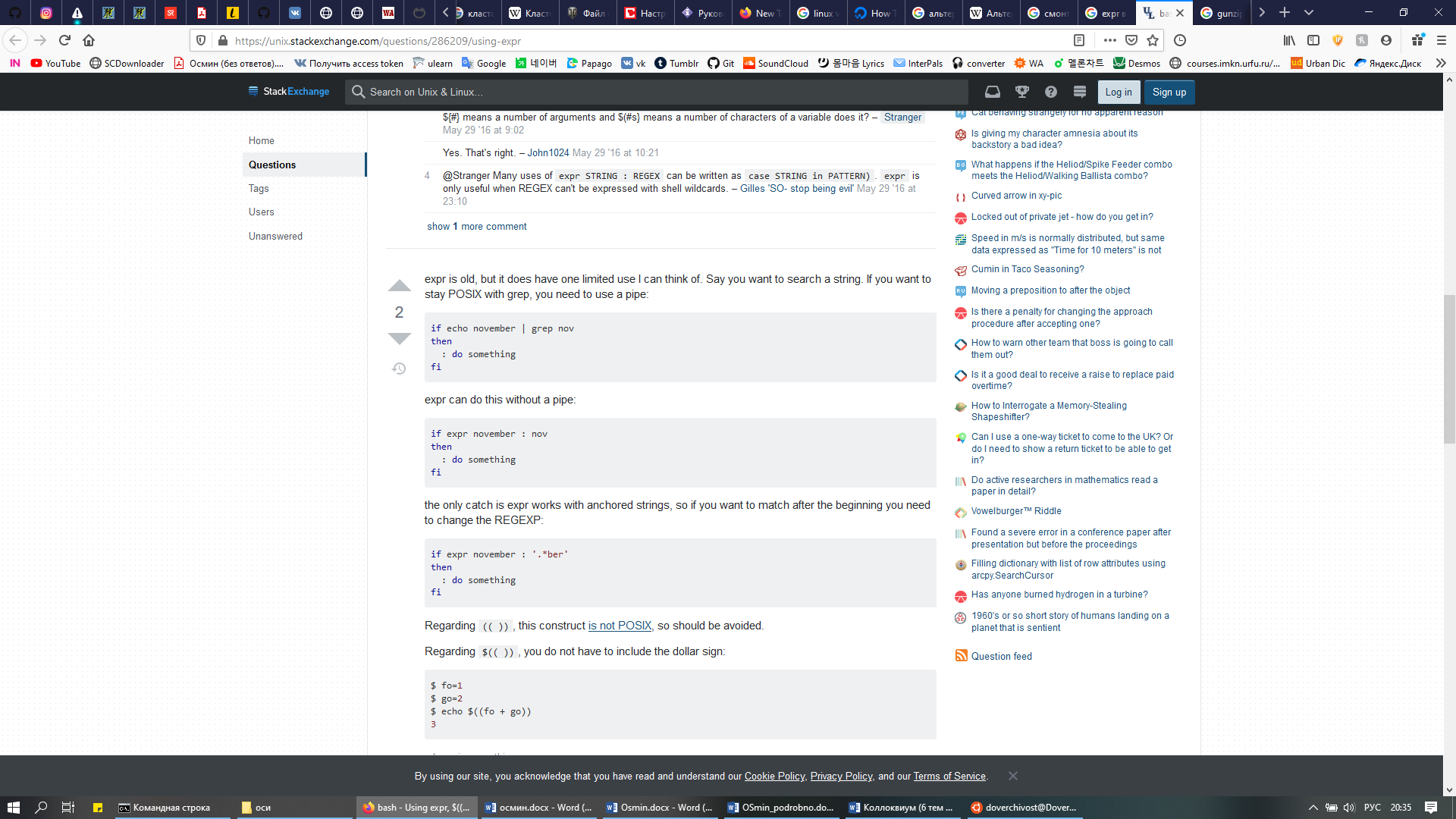
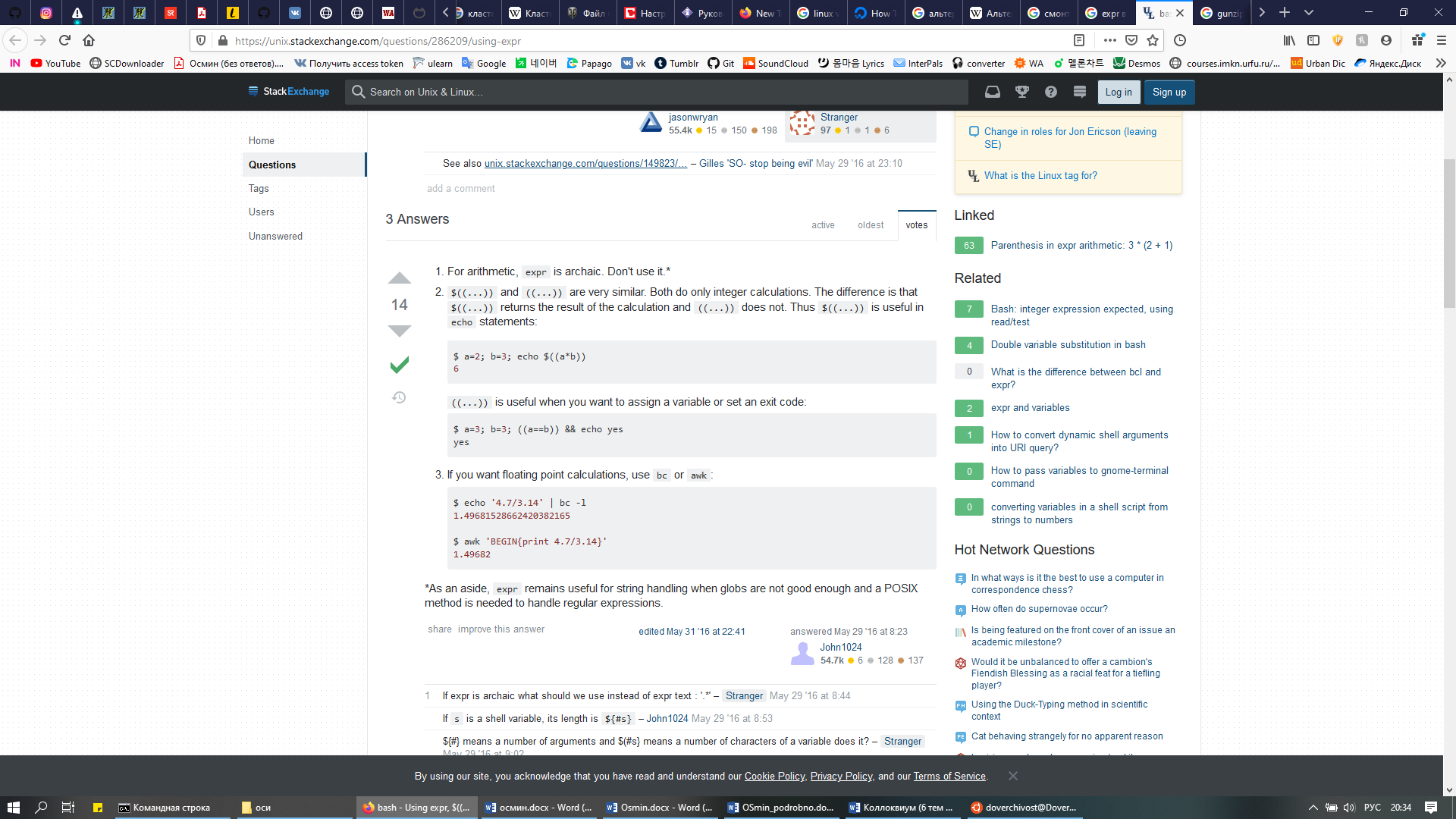
Программные – вызов системной функции в коде программы, будто обычную функцию (int 21h).

От немаскируемых прерываний зависит жизнь компьютера.

|  |  |
| --- | --- |
| Утилиты и команды CMD (+что делают, что возвращают, опции, расширения) | |
| date | Вывод и изменение даты.  /t – просто вывод, без запроса новой даты. |
| time | Вывод и изменение времени.  /t – просто вывод, без запроса нового времени. |
| at | Запуск программ и команд в указанное время по определённым дням.  Указывается время запуска и команда Window NT или имя пакетного файла.  /delete – отмена запланированной задачи. Без кода отменяются все задачи.  /yes – отмена запроса на подтверждение при отмене всех запланированных задач.  /interactive – разрешение взаимодействия задачи с пользователем.  /every – по указанным дням недели или месяца.  /next – запуск в следующий указанный день. |
|  |  |
| cmd | Запуск новой копии интерпретатора команд Windows.  /c – выполнение последующей команды (строки) с последующим завершением.  /k – выполнение без последующего завершения.  /s – изменение поведения после /c и /k.  /q – отключение режима вывода команд на экран (echo).  /d – отключение выполнения команд AutoRun из реестра.  /a – вывод результатов выполнения команд в формате ANSI.  /u – вывод результатов выполнения команд в формате Unicode. |
| start | Запускает указанную программу или команду в отдельном окне.  Можно указать заголовок нового окна (вместо “C:\Windows\system32\cmd.exe”).  /min – в свернутом окне  /max – в развернутом (на весь экран) окне.  Можно выбирать приоритет. |
| pause | Приостановка выполнения пакетного файла и вывод сообщения:  Для продолжения нажмите любую клавишу…  Press any key to continue . . . |
| echo | Вывод сообщений и переключение режима отображения команд на экране  echo – отображение текущего значения режима вывода команд.  echo [on | off] – вкл/выкл вывода команд на экран  echo [message] – вывод сообщения.  @echo off |
| cls | Очищает содержимое экрана. |
|  |  |
| type | Вывод содержимого одного или нескольких текстовых файлов. |
| more | Последовательный вывод данных по частям размеров в один экран.  MORE [/E [/C] [/P] [/S] [/Tn] [+n]] < [диск:][путь]имя\_файла  имя\_команды | MORE [/E [/C] [/P] [/S] [/Tn] [+n]]  MORE /E [/C] [/P] [/S] [/Tn] [+n] [файлы]  /e – разрешение использования дополнительных возможностей.  /c – очистка экрана перед выводом каждой страницы  /s – сжатие нескольких пустых строк в одну.  +n – начало выода первого файла с n-ой строки. |
| find | Поиск текстовой строки в одном или нескольких файлах.  /v – вывод всех строк, НЕ содержащих заданную строку.  /c – вывод только общего числа строк, содержащих данную строку.  /n – вывод номеров отображаемых строк.  /i – без учета регистра. |
| sort | Сортировка в алфавитном порядке строк текстового файла или стандартного ввода.  /r – в обратном порядке  /o [путь] – определяет имя файла, в котором сохраняется результат (не в консоль). |
|  |  |
| dir | Вывод списка файлов и подкаталогов в указанном каталоге.  /a (DRHASILO-) – отображение файлов с указанными атрибутами (D каталоги; R только чтение; H скрытые; A готовые для архивирования; S системные файлы; …).  /b – вывод только имен файлов.  /o – сортировка (N по имени, S – по размеру, E – по расширению, и тд).  /s – отображение файлов из указанного каталога и всех его подкатлогов. |
| cd | Выводит имя или изменяет текущий каталог.  cd /d | chdir /d – одновременная смена текущего диска и каталога.  cd .. | chdir .. – переход в родительский каталог. |
| md | Создание каталога.  Команда mkdir при необходимости создает промежуточные каталоги (mkdir\a\b\c\d). |
| rd | Удаление каталога.  /s – удаление дерева каталогов (т.е. всех файлов и подкаталогов).  /q – отключение запроса подтверждения при использовании /s. |
| del | Удаление одного или нескольких файлов.  /p – запрос подтверждения перед удалением каждого файла.  /f – принудительное удаление файлов, доступных только для чтения.  /s – удаление указанных файлов из всех подкаталогов.  /q – отключение запроса на подтверждение при удалении файлов.  /a – отбор файлов для удаления по атрибутам. |
| ren | Переименование одного или нескольких файлов (нельзя указать другой диск или путь). |
| move | Перемещение файлов и переименование файлов и папок. |
| copy | Копирование одного или нескольких файлов в другое место. |
| xcopy | Копирует файлы и деревья каталогов.  /a – с установленным атрибутом архивации (атрибут не меняется).  /m - с установленным атрибутом архивации (после копирования атрибут снимается).  /d:м-д-г – файлы, измененный не ранее указанной даты.  /s – копирует только непустые каталоги с подкаталогами. /e – включая пустые. |
| attrib | Отображает или изменяет атрибуты файлов. + устанавливает атрибут, - снимает его.  R Атрибут "Файл только для чтения".  A Атрибут "Файл архива".  S Атрибут "Системный файл".  H Атрибут "Скрытый файл".  O Атрибут "Автономный".  I Атрибут "Файл с неиндексированным содержимым".  X Атрибут "Файл без очистки".  V Атрибут "Целостность".  P Атрибут "Закрепленный".  U Атрибут "Открепленный".  B BLOB-атрибут SMR. |
| fc | Сравнение двух файлов или двух наборов файлов и вывод различий между ними.  /a – вывод только первой и последней строки для каждой группы различий.  /c – сравнение без учета регистра.  /w – пропуск пробелов и символов табуляции при сравнении. |
|  |  |
| if | Оператор условного выполнения команд в пакетном файле.  IF [NOT] ERRORLEVEL число команда  IF [NOT] строка1==строка2 команда  IF [NOT] EXIST имя\_файла команда  NOT - Указывает, что Windows должна выполнить эту команду, только если условие является ложным.  ERRORLEVEL число - Условие является истинным, если код возврата последней выполненной программы не меньше указанного числа.  строка1==строка2 - Условие является истинным, если указанные строки совпадают.  EXIST имя\_файла - если файл с указанным именем существует.  Команда – Задает команду, выполняемую при истинности условия. За этой командой может следовать ключевое слово ELSE. В случае, если указанное условие ложно, будет выполнена команда, находящаяся после слова ELSE. |
| for | Выполнение указанной команды для каждого файла набора.  FOR %переменная IN (набор) DO команда [параметры].  В пакетных файлах используется %%I вместо %I. Регистр важен.  /D – для всех подходящих имен каталогов, а не имен файлов.  /R – для каталога и всех подкаталогов.  /L – набор раскрывается в последовательность чисел (начало,шаг,конец)  /F – открытие, чтение обработка файлов (разбор каждой строки). |
| set | Вывод, задание и удаление переменных среды cmd.exe.  SET – вывод текущих переменных среды  SET P – вывод переменных, имена которых начинаются на «Р».  Не допускает использование «=» в имени переменной.  /A – числовое выражение, значение которого вычисляется  /P – позволяет установить значение переменной для входной строки, введенной пользователем. Показывает указанное приглашение promptString перед чтением  введенной строки. Приглашение promptString может быть пустым. |
| setx | Создание или изменение переменных среды в пользовательской или системной среде. Переменные можно задавать с помощью аргументов, параметров реестра или ввода из файла.  Примечания:  1) SETX записывает переменные в главную среду в реестре.  2) На локальном компьютере переменные, созданные или измененные данным средством, будут доступны в будущем окне командной строки, но не в текущем окне CMD.exe.  3) На удаленном компьютере переменные, созданные или измененные данным средством, будут доступны со следующего сеанса входа.  4) Допустимые типы параметров реестра: REG\_DWORD, REG\_EXPAND\_SZ, REG\_SZ, REG\_MULTI\_SZ.  5) Поддерживаемые кусты: HKEY\_LOCAL\_MACHINE (HKLM), HKEY\_CURRENT\_USER (HKCU).  6) Разделители определяются с учетом регистра.7) Значения типа REG\_DWORD извлекаются из реестра в десятичном формате. |
| setlocal | Начало локализации изменений среды в пакетном файле. Изменения среды, внесённые после этой команды, являются локальными относительно текущего пакетного файла. Для восстановления их прежних значений необходимо выполнить команду ENDLOCAL. При достижении конца пакетного файла автоматически выполняется команда ENDLOCAL для всех действующих команд SETLOCAL в этом пакетном файле.  ENABLEEXTENSIONS / DISABLEEXTENSIONS – расширенная обработка команд.  ENABLEDELAYEDEXPANSION / DISABLEDELAYEDEXPANSION – отложенное расширение переменных среды. Если отложенное расширение переменных среды включено, то для замены текущего значения переменной среды во время выполнения может использоваться символ восклицательного знака "!". |
| endlocal | Завершение локализации изменений среды в пакетном файле. Изменения среды, внесенные после выполнения этой команды, не являются локальными относительно текущего пакетного файла; их прежние значения не будут восстановлены по завершении этого пакетного файла. |
| pushd | Сохраняет текущий каталог для использования командой POPD, а затем переходит в указанный каталог. Когда расширенная обработка команд включена, команда PUSHD допускает ввод сетевых путей в дополнение к обычным именам дисков и путям. Если указан сетевой путь, команда создает временное имя диска, указывающее на заданный сетевой ресурс, а затем производит смену текущего диска и каталога, используя вновь определённое имя диска. |
| popd | Переход в каталог, сохраненный PUSHD.  Когда расширенная обработка команд включена, POPD удаляет временные имена дисков, созданные командой PUSHD для сетевых ресурсов, при удалении соответствующего диска из стека каталогов. |
| shift | Изменение содержимого (сдвиг) подставляемых параметров пакетного файла.  /n – число от 0 до 9, задающее начало сдвига  (/2: %3 заменяется на %2, %4 на %3 и т.д., а %0 и %1 остаются без изменений.) |
|  |  |
| goto | Передача управления содержащей метку строке пакетного файла.  Метка должна находится в отдельной строке и начинаться с двоеточия (:М)  :EOF – переходит на конец текущего пакетного файла. |
| call | Вызов одного пакетного файла из другого. Допускается вызов меток в качестве адресата.  CALL [диск:][путь]имя\_файла [параметры]  CALL :метка аргументы |
| exit | Завершает программу CMD.exe (интерпретатор команд) или текущий пакетный файл.  /b – завершить текущий пакетный файл. /b exitCode – указывает номер для ERRORLEVEL. |
|  |  |
| format | Форматирование дичка для работы с Windows.  FORMAT том [/FS:ф\_с] [/V:метка] [/Q] [/L[:состояние]] [/A:размер] [/C] [/I:сост.] [/X] [/P:проходы] [/S:сост.]  /FS:фс – указывает тип файловой системы (FAT, FAT32, exFAT, NTFS, UDF, ReFS).  /Q – быстрое форматирование.  /P:count – обнуление каждого сектора тома. После обнуления том будет перезаписан count раз; при этом при каждой перезаписи будет использоваться новое случайное число. |
| chkdsk | Проверка диска и вывод отчета.  CHKDSK [том[[путь]имя\_файла]]] [/F] [/V] [/R] [/X] [/I] [/C] [/L[:размер]] [/B] [/scan] [/spotfix]  Тип файловой системы. Занятость памяти на диске. |
| chkntfs | Выводит или изменяет параметры проверки диска во время загрузки. |
|  |  |
| cacls | ПРИМЕЧАНИЕ: CACLS считается устаревшей, рекомендуется использовать ICACLS.  Просмотр и изменение списком управления доступом (ACL) к файлам.  /t – изменение ACL для указанных файлов в текущем каталоге и вложенных каталогов.  /d имя – запрет на доступ пользователя. |
| icacls | Команда iCACLS позволяет отображать или изменять списки управления доступом (**A**ccess **C**ontrol **L**ists (ACLs) ) к файлам и папкам файловой системы. Утилита iCACLS.EXE является дальнейшим усовершенствованием утилиты управления доступом CACLS.EXE.  Управление доступом к объектам файловой системы NTFS реализуется с использованием специальных записей в таблице MFT (Master File Table). Каждому файлу или папке файловой системы NTFS соответствует запись в таблице MFT, содержащая специальный дескриптор безопасности **SD** (Security Descriptor). Каждый дескриптор безопасности содержит два списка контроля доступа: **- System Access-Control List (SACL)** - системный список управления доступом . **- Discretionary Access-Control List (DACL)** - список управления избирательным доступом.  **SACL** управляется системой и используется для обеспечения аудита попыток доступа к объектам файловой системы, определяя условия при которых генерируется события безопасности. В операционных системах Windows Vista и более поздних, SACL используется еще и для реализации механизма защиты системы с использованием уровней целостности ( Integrity Level, IL).  **DACL** - это собственно и есть список управления доступом ACL в обычном понимании. Именно **DACL** формирует правила, определяющие, кому разрешить доступ к объекту, а кому – запретить. Каждый список контроля доступа (ACL) представляет собой набор элементов (записей) контроля доступа - **Access Control Entries**, или **ACE**) . Записи **ACE** бывают двух типов (разрешающий и запрещающий доступ), и содержит три поля:   * **SID** пользователя или группы, к которому применяется данное правило * **Вид доступа**, на которое распространяется данное правило * **Тип ACE** - разрешающий или запрещающий. |
|  |  |
| tasklist | Отображает список процессов, которые сейчас выполняются на локальном или удаленном компьютере. |
| taskkill | Завершает процесс по его ID (PID) или имени образа.  /PID <процесс> - завершение по идентификатору процесса.  /IM <образ> - имя образа, которое необходимо завершить.  /F – принудительное завершнеие. |
|  |  |
| where | Отображает расположение файлов, совпадающих с шаблоном поиска. По умолчанию поиск выполняется в текущем каталоге и в каталогах, указанных в переменной среды PATH.  /r – рекурсивный поиск, начиная с указанного каталога.  /q – возврат только кода выхода без отображения списка найденных файлов.  /f – отображение найденных файлов в кавычках.  /t – отображение размеров, даты и времени изменения всех найденных файлов. |

ftype – Просмотр и изменение типов файлов, сопоставленных с расширением имен файлов

|  |  |
| --- | --- |
| Утилиты GNU/Linux и команды bash (+что делают, что возвращают, основные опции) | |
| uname | Выводит информацию о системе. |
| date | Дата и время. |
| time | Время выполнения команды. |
| ps | Информация о процессах. |
| top | Список наиболее активных процессов в реальном времени (только то, что помещается на экран). |
| dmesg | Проверка или управление буфером кольца ядра.  The kernel keeps its logs in a ring buffer. The main reason for this is so that the logs from the system startup get saved until the syslog daemon gets a chance to start up and collect them. Otherwise there would be no record of any logs prior to the startup of the syslog daemon. The contents of that ring buffer can be seen at any time using the dmesg command, and its contents are also saved to /var/log/dmesg just as the syslog daemon is starting up. |
|  |  |
| bash | Интерпретатор. |
| read | Считывает строку и разделяет на поля. |
| echo | Вывод строки. |
| clear | Очистка терминала. |
| history | Содержит историю вызова команд bash. |
| bg | Возобновление исполнения остановленной задачи в фоновом режиме. |
| fg | Перевод фоновой задачи в обычный режим. |
|  |  |
| cat | Печать содержимого файла. |
| more | Построчный вывод содержимого файла. |
| head | Начальные строки файла (default: 10) |
| tail | Конечные строки файла (default: 10) |
| less | Просмотр содержимого текстового файла с возможностью вернуться к предыдущим страницам. |
| find | Найти файл с именем «имя» в директории иерархично. Можно по шаблону. |
| wc | Подсчет строк, слов и байтов. |
| cut | Разрезать строку на подстроки. |
| grep | Поиск фрагмента текста в файлах, удовлетворяющих шаблону. |
| sort | Сортирует строки файла в алфавитном порядке. Если задать несколько файлов, то соединит их и, отсортировав, выдаст единым выводом. |
| awk | Утилита для обработки строк. |
|  |  |
| ls | Список файлов текущей директории (папки с фоном, исполняемые - зеленые). |
| cd | Сменить директорию. |
| mkdir | Создать директорию. |
| rmdir | Удалить директорию. |
| mknod | Создание специальный файлов устройств (блок- или байт ориентированные, небуферизированное байт-ориентированное устройство, именованный канал). |
| rm | Удалить файл. |
| mv | Переместить файл. |
| cp | Копировать файлы. |
| chmod | Изменить права пользователя на папку/файл. |
| chown | Сменить владельца папки/файла. |
|  |  |
| diff | Утилита сравнения файлов, выводящая разницу между ними. |
| touch | Изменение времени последнего доступа к файлу. Если такого файла нет – создает файл с таким именем. |
| du | Выводит информацию о размере файлов по директориям начиная с текущей. |
| df | Вывод информации о свободном и используемом месте на диске. |
| stat | Подробная информация о заданном файле или каталоге. |
|  |  |
| if | Условный переход |
| for | Цикл |
| while | Цикл «пока». |
| case | Оператор выбора. |
| function | Объявление функции. |
| set | Изменение значения внутренних переменных. |
| env | Переменные окружения. Вывод и создание. |
| export | Экспорт переменных и функций текущего процесса в дочерний процесс. |
| declare | Создание переменных с атрибутами. |
|  |  |
| sleep | Приостанавливает исполнение сценария на заданное количество секунд, ничего не делая (sleep 10s; sleep 10m; sleep 1h). |
| exit | Завершение выполнения. |
|  |  |
| tar | Архиватор. Несколько файлов в один диск. |
| gzip | Архивирование и разархивирование файлов. |
| hd | Все виды дампов (ascii, hex, oct, dec) (default: 16) |
| dd | "data duplicator", создает копию файла (или stdin/stdout), выполняя по пути некоторые преобразования. Может использоваться для создания образов дисков, считывая данные прямо с устройств. |
| gunzip | Разархивировать |
| zcat | cat для архивированных файлов. |
| zless | less -..-. |
|  |  |
| [ -e|f|d|x|s|z ] (test) | Логические условия. |
| expr | Вычисление значения выражения. |
| bc | Калькулятор. |
|  |  |
| fdisk | Семейство утилит для работы с разделами жесткого диска. |
| cfdisk | Улучшенный fdisk. |
| mkfs | Создание файловой системы Linux на устройстве. |
| fsck | Проверка и устранение ошибок файловой системы. |
| mount | Смонтировать файловую систему. Монтирование файловой системы — системный процесс, подготавливающий раздел диска к использованию операционной системой. |
| umount | Размонтировать файловую систему |
|  |  |
| shutdown | Позволяет автоматически завершить процессы в указанное время, оповестить пользователь и только потом выключить систему. |
| poweroff | Выключить. |
| reboot | Остановить и перезапустить. |
| su | Вход в систему как суперпользователь. |
| sudo | Исполнить заданную команду от имени суперпользователя. |
| passwd | Устанавливает или изменяет пароль пользователя. |
|  |  |
| who | Список работающих в данный момент на машине пользователей. |
| w | Список активных пользователей и что они сейчас делают. |
| whoami | Имя текущего пользователя. |
| id | Выводит идентификатор пользователя (реальный ($UID) и эффективный ($EUID)) и список групп, в которых он состоит ($GROUPS) |
| which | Поиск бинарных файлов в PATH. Возвращает полный путь к файлу. |
| lsof | Вывод информации о том, какие файлы используются процессами. |
|  |  |
| source | Запуск скрипта внутри текущей сессии. |
| . | Выполнить команду из файла в текущей оболочке. |
|  |  |
| : |  |
| yes | Непрерывный вывод указанной строки на экран. Если строка не указана – «у». |
| true | Успешно ничего не делай. (после - $? = 0) |
| false | Неудачно ничего не делай. (после - $? = 1) |
|  |  |
| xargs | Команда передачи аргументов указанной команде, может рассматриваться как мощная замена обратным одиночным кавычкам. Зачастую, когда команды, заключенные в обратные одиночные кавычки, завершаются с ошибкой too many arguments (слишком много аргументов), использование xargs позволяет обойти это ограничение. |
| curl | Программа для передачи данных по различным протоколам. |
| ssh | Протокол удаленного управления компьютером. В основном используется для удаленного управления серверами через терминал. |
| wget | Свободная неинтерактивная консольная программа для загрузки файлов по сети |



|  |  |
| --- | --- |
| Переменные окружения Windows (+что содержат, как и кем используются) | |
| windir | Каталог, в котором **установлена Windows**. |
| TEMP | **Временные каталоги**, по умолчанию используемые приложениями, которые доступны пользователям, выполнившим вход в систему. |
| ProgramFiles | Путь к каталогу ProgramFiles, где хранятся **установленные программы**. |
| USERPROFILE | Каталог **текущего пользователя**. |
| Path | Путь поиска **исполняемых файлов** (разделены ;). |
| PATHEXT | Список **расширений**, которые рассматриваются ОС как **исполняемые**. |
| CD | Путь к **текущей папке**. |
| DATE | **Текущая дата** (>date /t). |
| TIME | **Текущее время** (>time /t). |
| RANDOM | **Случайное десятичное число** от 0 до 32767. |
| ERRORLEVEL | **Код ошибки** последней использовавшейся операции (≠0 -> есть ошибка). |
| %windir%\SoftwareDistribution\Download | SoftwareDistribution – содержит файлы обновлений системы. Download – в данном каталоге ОС сохраняет **апдейты, скачанные с сервера Microsoft.** |

|  |  |  |
| --- | --- | --- |
| Переменные окружения GNU/Linux (+что содержат, как и кем используются): | | |
| PATH | | Путь поиска **исполняемых файлов** (разделены :). |
| RANDOM | | Внутренняя функция Bash, возвращающая **псевдослучайные десятичные целые числа** от 0 до 32767. |
| PWD | | Путь к **текущему каталогу**. |
| HOME | | **Домашний** каталог **текущего пользователя**. |
| LOGNAME | | Имя пользователя, которое использовалось для входа в систему. **Имя текущего залогиненного пользователя**. |
| USER | | **Имя текущего пользователя**. |
| UID | | Числовой идентификатор текущего пользователя (**ID пользователя**). |
| HOSTNAME | | **Имя данного ПК** (сетевое имя текущего хоста). |
| HOSTTYPE | | **Архитектура машины**. |
| LANG | | Текущие настройки **языка и локализации**, в том числе кодировка символов. |
| IFS | | **Разделитель полей** (по умолчанию пробел). |
| PS1 | приглашения оболочки | Определяет вид **приглашения интерпретатора** (Bash), которая выдается, когда оболочка **ждет ввода команды** ($ - обычный пользователь; # - суперпользователь). |
| PS2 | Вторичное приглашение появляется, когда оболочка ожидает **ввода еще каких-то данных, необходимых для продолжения работы**. При нажатии ENTER, синтаксически не завершив ввод команды (По умолчанию: >). |
| PS3 | Приглашение, используемое **командой select**. |
| PS4 | Выводится **при отладке скриптов в начале каждой строки** перед исполняющейся командой (По умолчанию: +). |
| \_ | | **Аргументы последней команды** (не обязательно выполненной(т.е. несуществующей) и даже не обязательно команды). Код возврата команды, функции или скрипта. |
| ? | | **Код ошибки** последней выполненной команды.  The underscore variable is set at shell startup and contains the absolute file name of the shell or script being executed as passed in the argument list. Subsequently, it expands to the last argument to the previous command, after expansion. It is also set to the full pathname of each command executed and placed in the environment exported to that command. When checking mail, this parameter holds the name of the mail file. |
| $ | | Содержит **идентификатор текущего процесса**.  Играет важную роль при создании уникальных файлов. Каждый новый расчет, выполняемый компьютером, инициирует один или несколько процессов, автоматически получающих номера по порядку. Поэтому, используя номер процесса в качестве имени файла, можно быть уверенным, что каждый новый файл будет иметь новое имя. |

|  |  |  |  |
| --- | --- | --- | --- |
| Утилиты DOS/Win (+что делают, что возвращают, опции и где взять) | | | |
| regedit | **Единый редактор реестра**. Хранилище системы, содержащее как настройки и параметры ОС, так и программ, установленных в ней. | Windows | |
| reg | **Консольный редактор реестра**. | WindowsSystem32 | |
|  | | | |
| runas | **Запуск** приложения **от имени** другого пользователя. | консоль | |
| shutdown | **Завершение работы**. | WindowsSystem32 | |
|  | | | |
| winmerge | **Сравнение и синхронизация** (слияние) **файлов и директорий** (свободное ПО. Распространяется на условиях лицензии GNU GPL) |  | |
|  | | | |
| diskpart | Консольная утилита для **управления разделами жестких дисков**. | WindowsSystem32 | |
| diskmgmt.msc | **Управление дисками** (создание и форматирование разделов и томов, назначение букв дисков). | WindowsSystem32 | |
| compmgmt.msc | (*Computer Management*) **Управление компьютеров**. Инструмент, предназначенный для управления локальными и удаленными компьютерами. | WindowsSystem32 | |
| gpedit | **Редактор групповой политики** это отдельная MMC оснастка, которая по сути представляет собой графическую надстройку для удобного управления параметрами Windows в реестре. |  | |
|  | | | |
| procmon | (*process monitor*) **Мониторинг** и слежение за работой всей **файловой системы**, а также предоставляет подробные описания событий в реальном времени | sysimternals | |
| diskmon | **Монитор действий над дисками**. Сохраняет и показывает всю активность жесткого диска. | sysimternals | |
|  | | | |
| autoruns | **Управление автозагрузками** (программ, сервисов, модулей, драйверов и других компонентов системы). | sysimternals | |
| net | **Пользование сетью**. Управление сетевой конфигурацией в командной строке. | консоль | |
| mstsc | (*Microsoft Terminal Services Client*) Утилита для **управления подключениями к удаленному рабочему столу**.  Создание подключений к серверам терминалов или другим удаленным компьютерам, редактирование существующего файла конфигурации подключения к удаленному рабочему столу (.rdp) и перенос подключений Windows XP (подключения, созданные с помощью диспетчера клиентских подключений) в новые файлы .rdp. | WindowsSystem32 | |
| msconfig | Утилита для управления **автозапускаемыми программами и загрузкой** Windows. | |  |
| fsutil | **Управление объектами файловой системы в командной строке** Windows. Позволяет оптимизировать параметры надежности и быстродействия программно-аппаратных систем хранения и обработки данных. (управление точками повторной обработки, управление разреженными файлами, отключение тома или расширение тома) | |  |
| dir /r | **Список альтернативный файловых потоков** в NTFS.  Альтернативные потоки данных (англ. Alternate Data Streams, ADS) — метаданные, связанные с объектом файловой системы NTFS. В файловой системе NTFS файл, кроме основных данных, может также быть связан с одним или несколькими дополнительными потоками данных. При этом дополнительный поток может быть произвольного размера, в том числе может превышать размер основного файла. | |  |
| chcp | Команда CHCP используется для просмотра или **изменения текущей кодовой страницы** в окне командной строки Windows. Кодовая страница (Code Page или сокращенно CP) определяет соответствие между двоичным кодом и соответствующим ему символом, отображаемом на экране. | |  |
| whoami | Команда WHOAMI – получение информации о текущем пользователе Windows. Команда WHOAMI используется для получения сведений об имени пользователя и группе, а также о соответствующих идентификаторах безопасности (SID), привилегиях, идентификаторах входа (ID) текущего пользователя (токене доступа) на локальном компьютере. | |  |
| gpedit | (выше) | |  |
| mklink | Создание **символической** (символьной) **ссылки** на файл или каталог файловой системы Windows. | |  |
| contig | Предназначена для **оптимизации отдельных файлов**, и имеет две функции: во-первых, собирает фрагменты файла, разбросанные по всему разделу диска в один непрерывный файл или, по крайней мере, минимизирует количество фрагментов, и, во-вторых, ее можно использовать для создания но­вых непрерывных файлов. | |  |
| defrag | Используется для **определения степени фрагментации и объединения фрагментированных файлов** на локальных томах в целях повышения производительности системы. | |  |
| mountvol | Позволяет **создавать, удалять и просматривать точки подключения томов** (точки монтирования) в командной строке. Точки монтирования доступны при использовании NTFS. В среде операционных систем семейства Windows, существует два вида точек монтирования: точка монтирования каталога (англ. junction point) и точка монтирования тома (англ. volume mount point). Создание точек монтирования первого типа осуществляется через консольную команду mklink /J, создание точек монтирования второго типа — через команду mountvol. | |  |
| sysdm.cpl | System Applet for the Control Panel. Он используется для загрузки системных свойств в Windows. | |  |
| netplwiz | это «скрытый» инструмент, используемый для **управления учетными записями пользователей**. | |  |
| wsreset | Встроенный инструмент в командную строку, который **очищает и сбрасывает кэш**. | |  |
| gpupdate | **Выполнение обновления групповых политик для пользователя и компьютера**.  Команда gpupdate обновляет как локальные параметры групповой политики, так и параметры на основе Active Directory, включая параметры безопасности на компьютере, где исполняется команда. Команду gpupdate можно использовать на локальном компьютере для немедленного обновления политики. | |  |
| gpresult | Представляет собой консольное приложение, предназначенное для анализа настроек и диагностики групповых политик, которые применяются к компьютеру и/или пользователю в домене Active Directory. В частности, GPResult позволяет получить данные результирующего набора политик (Resultant Set of Policy, RSOP), список примененных доменных политик (GPO), их настройки и детальную информацию об ошибках их обработки. | |  |
| gpedit | (выше) | |  |
| pushd | Команда PUSHD используется совместно с командой POPD для **упрощения переходов между каталогами файловой системы**. Используется для запоминания текущего каталога в виртуальном стеке каталогов и переходу в указанный параметром командной строки. Многократное выполнение команды позволяет запоминать пути к рабочим каталогам файловой системы и осуществлять возврат при выполнении команд POPD. | |  |
| slmgr | Команда slmgr используется для **управления лицензированием программного обеспечения** Windows на локальном или удаленном компьютере. Представляет собой сценарий VBS и при выполнении использует параметры, хранящиеся в файле %SystemRoot\system32\slmgr\slmgr.ini | |  |

***Системный реестр*** — иерархически построенная база данных параметров и настроек в большинстве операционных систем Microsoft Windows.

***Реестр Windows*** (системный реестр) – иерархически построенная база данных параметров и настроек. Реестр содержит информацию и настройки для аппаратного обеспечения, программного обеспечения, профилей пользователей, предустановки. Большинство изменений в Панели управления, ассоциации файлов, системные политики, список установленного ПО фиксируются в реестре.

***HKEY\_CURRENT\_USER*** содержит настройки текущего активного пользователя, вошедшего в систему. Здесь хранятся папки пользователя, цвета экрана и параметры панели управления. Выглядит как один из основных разделов, но на самом деле является ссылкой на один из профилей в HKEY\_USERS\.

***HKEY\_LOCAL\_MACHINE*** содержит параметры конфигурации, относящиеся к данному компьютеру для всех пользователей.

|  |
| --- |
| Ключи реестра (+кем и как используются) |

**HKLM\System\CurrentControlSet\Services**

Ключ Services содержит **список драйверов, файловых систем, сервисных программ** (программы обслуживания персонального компьютера и его программного обеспечения), работающих в режиме пользователя. Данные этого ключа **определяют загружаемые сервисы и задают порядок их загрузки**. Кроме того, эти данные определяют, каким образом эти данные могут вызывать друг друга.

**HKCU(LM)\Software\**

Вложенные ключи, описывающие конфигури**руемые параметры настройки установленного на компьютере программного обеспечения**, с которым пользователь (вошедший в систему в данный момент) имеет право работать.

**HKCU(LM)\Software\Microsoft\Windows\ CurrentVersion**

**Run**

Программы, которые **запускаются каждый раз при входе** пользователя в систему.

**RunOnce**

Программ, которые **запускаются только один раз при входе** пользователя в систему. После запуска ключи программ **автоматически удаляются** из данного раздела реестра.

**HKCU(LM)\Software\Microsoft\Windows NT\CurrentVersion\Winlogon**

**Настройки входа в систему.**

(**LM** – для всех пользователей; **CU** – для текущего)

|  |  |
| --- | --- |
| Структура каталогов в GNU/Linux (+их назначение) | |
| . | **Текущая** директория. |
| .. | **Родительская** директория. |
| ~ | **Домашняя** директория **текущего пользователя**. |
| / | **Корневой каталог**, который формирует базу файловой системы. Все файлы и каталоги логически содержатся внутри него, независимо от их физического местоположения. |
| /bin | Каталог, в котором хранятся **основные исполняемые программы**, являющиеся частью операционной системы Linux. |
| /boot | Содержит **конфигурационные файлы загрузчика GRUB, образцы ядра, файлы Initid**. Когда начинается начальная загрузка системы, сначала монтируется временный виртуальный диск, который содержит временную корневую файловую систему, с помощью которой осуществляется запуск Linux. Файл Initid – образ этой временной корневой системы, который хранится на загрузочном устройстве. |
| /dev | Директория, содержащая **файлы устройств**, подключенные к операционной системе (консоль, жесткий диск, дисковод, порт и т.д.). |
| /etc | Содержит основную часть **конфигурационных файлов** самой операционной системы (настройки сети, список пользователей, групп и т.д.) и различных программ (Apache, Samba и т. д.). |
| /home | Содержит **домашние каталоги всех пользователей**. |
| /mnt | Монтированные вручную **временные** файловые системы (флэшка, диск). |
| /media | Монтированные **автоматически различные устройства** USB-накопителей, CD-ROM и т.д. При вставке в привод CD-ROM диска он автоматически подключается во вложенный каталог этой директории. |
| /proc | Информация о текущих запущенных процессах.  Директория, к которой примонтирована **виртуальная файловая система procfs**. **Информация о «железе»**, которую ядро может сообщить пользователям, находится в "файлах" этого каталога (список загруженных модулей ядра, информация о процессоре компьютера и тд). |
| /root | **Домашняя директория суперпользователя**. Фактически ничем не отличается от директории обычного пользователя. По умолчанию расположена в корневом каталоге. |
| /sbin | Основные **системные программы для администрирования и настройки системы** (iptables, ifconfig и т.д.). |
| /tmp | Хранение **временных файлов**. Все пользователи имеют права чтения и записи в этом каталоге. |
| /usr | Все **установленные пакеты программ, документация, исходный код ядра и**  **система X Window**. Все пользователи кроме root – только чтение. Может быть смонтирована по сети и может быть общей для нескольких машин. |
| /usr/local | Директория, содержащая **локальные файлы**. Использования при локальной установке программного обеспечения (которое не управляется системным пакетом).  **Программное обеспечение**, используемое **локально в рамках данного хоста**;  **не переустанавливается при обновлении системного ПО** и может использоваться программами вне /usr. |
| /usr/share | Директория, содержащая **общие файлы**, как, **конфигурация по умолчанию** (файлы, изображения, документы и т.д.).  **Архитектурно независимые общие данные**. |
| /var | Это каталог для **часто меняющихся данных** (журналы операционной системы, системные log-файлы, cache-файлы и т. д.) |
| /var/log | В этой директории хранятся **все log-файлы**. |

/mnt и /media

/mnt – электронная подпись, виртуальный жесткий диск.

/media – внешние устройства (флешка)

Media – не монтируемые, а подключаемые. Неизвестно, знает ли ОС, что это за устройство.

Смонтированное устройство попадает в mnt. Доступ к данным устройств через media.

|  |
| --- |
| **Сочетания клавиш GNU/Linux (+что и когда делают)** |

**Консоль:**

**TAB** - автодополнение в консоли (команды / файлы)

**Shift+PgUp, Shift+PgDown** - прокрутка окна консоли вверх/вниз

**Ctrl+C** - прервать выполнение команды (убить процесс) (Cill)

**Ctrl+\** - Посылка сигнала завершения программе (обрабатывается программой)

**Ctrl+Z** - приостанавливает выполнение программы и переводит её в фоновый режим (восстановление командой fg) (Zz-top)

**Ctrl+S** - приостановить вывод программы на терминал (Suspend)

**Ctrl+Q** - возобновить вывод (Qontinue)

**Ctrl+D** - удалить символ под курсором (Delete) / посылка сигнала "конец файла" / выйти из консоли

**Ctrl+L** - очистка консоли (cLear)

**Ctrl+N** - подставить следующую команду (Next)

**Ctrl+P** - подставить предыдущую команду (Previous)

**Ctrl+R** - поиск по истории команд (Reverse-history-search)

**Ctrl+-** - уменьшение шрифта в консоли (+/- на NumPad не работает)

**Работа с текстом (консоль):**

**Ctrl+A** - курсор в начало строки (A - начало алфавита)

**Ctrl+E** - курсор в конец строки (End)

**Ctrl+B** - перейти на символ влево (Backward)

**Ctrl+F -** перейти на символ вправо (Forward)

**Alt+F / Alt+B** - перейти на слово вперёд (Forward) / назад (Backward)

**Ctrl+W** - удалить слово слева (Word)

**Ctrl+H** - удалить символ слева (Налево или naHui)

**Ctrl+U** - Удалить всё слева от курсора

**Ctrl+K** - удалить всё справа от курсора (Kхуям, если не ошибаюсь)

**Ctrl+V** - "экранирование" - следующий введённый символ вставляется как есть (можно ввести [ctrl+c] и [ctrl+v, затем ctrl+c] и увидеть эффект)

**Ctrl+R** - восстановить команду, как она выглядела в истории (т.е. отмена всех изменений и возвращение к последней команде истории, в том числе пустой)

**Система:**

**Ctrl+Alt+Del** - перезагрузка

**Ctrl+Alt+Fn** - переключение между виртуальными терминалами (F7 - графический интерфейс).

**Alt+Fn** - то же самое, но работает только из виртуальных терминалов.

**Alt+SysRq+H** - Help — выводит все доступные возможности SysRq, выделяя большой буквой клавишу действия;

**Alt+SysRq+B** - reBoot — выполняет немедленную перезагрузку системы аналогично кнопке RESET (без синхронизации и размонтирования файловых систем);

**Alt+SysRq+C** - Crashdump — выполняет директиву kexec для перезагрузки чтобы получить сведения об ошибках.

Вызывает крах системы. Если настроено, будет сохранён crashdump.

**Alt+SysRq+O** - powerOff — корректно выключает систему (если настроено и поддерживается, обычно это прямая команда отключения);

**Alt+SysRq+S** - Sync - ядро производит попытку синхронизации файловых систем (т.е. записи на диске приводятся в соответствие с записями в кэше). //что-то вроде “безопасного извлечения”

|  |  |  |
| --- | --- | --- |
| Расширения файлов (+что внутри, в каких ОС и как используются) | | |
| exe | расширение **исполнимых** файлов | DOS, Windows, Symbian OS, OS/2 |
| dll | (dynamic-link library — «динамически подключаемая библиотека») — понятие операционных систем Microsoft Windows и IBM OS/2, **Динамическая библиотека**, позволяет различным программам получать доступ к общедоступным системным функциям. | Windows  IBM OS/2 |
| scr | (screensaver) - Обычно используется для установк**и заставки на компьютер**. Но в общем случае это просто исполняемый файл (скрипт?), который может выполнять и другие задачи. Часто под видом заставки распространяют вирусы. |  |
| vxd | (virtual xxx driver) - **драйвер виртуального устройства** | Windows 9x |
| com | (command) - **Исполняемый файл** DOS, CP/M, аналог .exe в Windows.  Очень часто встречался в операционной системе MS-DOS, как исполняемый файл игровых приложений. Также win.com запускал операционную систему Windows из-под MS-DOS. | MS-DOS  DOS  CP/M |
| sys | **Системные файлы**, которые содержат код, необходимый для правильной **загрузки и настройки Windows** (например config.sys, pagefile.sys), также использовались в ОС MS-DOS. Обычно устанавливаются вместе с операционной системой Windows, а также некоторыми программами. Большинство файлов с расширением .sys используются в качестве **драйверов**. | MS-DOS  Windows |
| ocx | Файл объекта, используемый в формах ActiveX (сокр. от англ. OLE Custom eXtension), специальных программах, которые созданы для использования приложениями в операционной системе Microsoft Windows. На самом деле OCX реализованы как **модуль динамически подключаемой библиотеки** .DLL. (Вы можете представить DLL как подпрограмму, которую может использовать любое число приложений, каждая из которых – это контейнер для DLL или OCX/ActiveX). Файлы .OCX чаще всего пишут на языках C++ и Visual Basic, то так же могут быть созданы и используя другие языки программирования. .OCX чаще всего используется для создания динамических веб-сайтов и приложений. | Microsoft |
| so | Файл совместно используемой библиотеки в \*nix-подобных системах, **функционально похож на файл dll** в операционной системе Windows. | Unix |
| a | Статическая библиотека, которая используется программным обеспечением, написанным на языке C/C++. Библиотека содержит набор **скомпилированных подпрограмм или объе**ктов, подключаемых к исходной программе в виде объектных файлов. | Unix |
| lib | Библиотека информации, используемая определенной программой. Часто **содержит функции и постоянные, на которые ссылается программа**. Может содержать объекты, такие как текст, изображения или другие медиа-файлы. На нее обычно ссылаются изображения, она не должна открываться вручную. |  |
| o | Файл, содержащий **скомпилированный код C или C++.** |  |
| obj | Файл, содержащий **3D-объекты**.  Формат файлов описания геометрии, разработанный в Wavefront Technologies для их анимационного пакета Advanced Visualizer. Формат файла является открытым и был принят другими разработчиками приложений 3D графики. |  |
|  |  |  |
|  |  |  |
| Пакетные файлы | | |
| bat | пакетный файл. BAT представляет собой текстовый файл, содержащий последовательность команд для операционной системы компьютера. |  |
| cmd | это расширение пакетных файлов | Windows NT, OS/2 |
| inf | (information file) Файл **конфигурации установки** программ/драйверов |  |
| reg | (Registration entries) - Файл осуществляет **обновление** либо добавление **информации в реестр** Windows при запуске двойным кликом мыши. Такие файлы чаще всего создаются для изменения различных настроек Windows. | Windows |
| sh | Сценарии в Bash | Unix |
| Ссылки | | |
| lnk | Ярлык, используемый в качестве ссылки на какой-либо файл, чаще всего **программу с расширением .exe**. Содержит такие данные, как тип объекта, путь к нему, атрибуты запуска, рабочую папку, размер открываемого окна и т.п. |  |
| url | Содержит внутри себя ссылку на определенный **сайт** или страницу в сети Интернет. Также может содержать дополнительную информацию (иконку, пути, дату изменения и т.д.) |  |
| Установочный пакет | | |
| msc | Установочный пакет для инсталлятора MS Windows Installer (встроенный установщик Windows). Может быть использован как для **обновления Windows**, так и сторонними разработчиками для **установки программного обеспечения**. Содержит в себе сценарий установки и установочные файлы. | Windows |
| Сценарии (скрипты) | | |
| js | Файл, содержащий код на языке программирования **JavaScript**, используемый для выполнения инструкций JavaScript в **веб-страницах**. |  |
| vbs | (VBScript) - Скрипты Visual Basic представляют собой текстовый файл с расширением .vbs; содержат набор команд, в соответствии с определенным синтаксисом языка и могут быть изменены в любом текстовом редакторе. |  |
| Архивы | | |
| cab | (Cabinet) - Файл архива, используемый в основном **программами-инсталляторами** от Microsoft. Их зачастую можно встретить в дистрибутивах Windows, MS Office и других программ. | Microsoft |
| zip | Файл, сжатый или запакованный в архив при помощи сжатия **Zip**. |  |
| rar | (Roshal ARchiver) - Файловый архив, содержащий один или несколько файлов и/или каталогов, упакованных с использованием сжатия **RAR**. Формат обеспечивает более высокую степень сжатия, чем стандартное ZIP-сжатие. |  |
| tar | Расширение tar – это общепринятое расширение для стандартного формата архивных файлов в системе Unix. Программа tar не создаёт сжатых архивов, а использует для сжатия внешние утилиты, такие, как **gzip и bzip2**. | Unix |
| gz | Сжатый **файл** формата GZ создается при помощи утилиты **gzip** (сокращение от GNU Zip), использующей алгоритм DEFLATE. Используется как правило в UNIX-системах, где зачастую является стандартом для сжатия данных. | Unix |
| bz2 | Сжатый **файл**, созданный утилитой сжатия файлов **bzip2**, часто используемой в Unix-системах. | Unix |
| tgz | Архивный файл **TAR** операционной системы Unix, который дополнительно сжат с **GNU Zip** (GZIP). | Unix |
| lzma | Lempel-Ziv-Markov chain-Algorithm) - Архив **LZMA**. Алгоритм основан на схеме сжатия данных по словарю, и обеспечивает высокий коэффициент сжатия (обычно превышающий коэффициент, получаемый при сжатии с использованием bzip2), а также позволяет использовать словари различного размера (до 4 Гб). |  |
| 7z | Сжатый файловый архив, созданный с помощью популярной программы-архиватора **7-Zip** |  |
| xz | Архив(или один файл), сжатый при помощи алгоритма **LZMA2** |  |
| Образы дисков | | |
| iso | **Образ данных CD или DVD** диска, созданный в соответствии со стандартом ISO-9660. Стандарт описывает файловую систему для дисков CD-ROM. |  |
| dmg | (Disk Image) - Файл образа диска, создаваемый в операционной системе **Apple Mac OS X**. Монтируется как жесткий диск, если открыть его в MAC. | Mac OS X |
| vmdk | (Virtual Machine Disk) - Виртуальный диск, хранящий **контент жесткого диска виртуальной машины VMware**. |  |
| vhd | (Virtual Hard Disk) **виртуальный жёсткий диск**. Это резервный образ жёсткого диска персонального компьютера.  Возможность посекторно скопировать диск. |  |
| mds | Информационный файл, описывающий **форматирование диска CD или DVD**. Включает в себя заглавную информацию, которая определяет трек на диске и способ структуризации данных. Записывается программами создания образов дисков и распознается стандартными проигрывателями CD и DVD. |  |
| mdf | Файл с расширением MDF (Media Disc Image File) является **образом CD или DVD диска**, сделанный с помощью специальной программы. Нередко совместно с файлом MDF находится файл c расширением MDS. В файле MDF хранится непосредственно образ диска, в файле MDS - вспомогательная информация о диске, с которого снят образ, например информация о дорожках |  |
| Установочные пакеты | | |
| msi | Установочный пакет для **инсталлятора MS Windows Installer** (встроенный установщик Windows). Может быть использован как для обновления ОС, так и сторонними разработчиками для установки ПО. Содержит в себе сценарий установки и установочные файлы. | Windows |
| rpm | Изначально разработанный для **Red Hat Linux**. Не редко используется для установки программ на системах Linux. | Linux |
| deb | **Инсталляционный пакет** в операционной среде **Debian**. В ОС, построенных на основе Unix , установка программного обеспечения происходит двумя способами. Либо, путем компиляции приложения из исходных текстов программного кода, либо установкой бинарных пакетов. | Unix – FreeBSD, Linux, Debian… |

|  |  |  |
| --- | --- | --- |
| Кодировки символов (+размер символа) | | |
| ASCII | (American standard code for information interchange) Изначально была разработана для кодирования символов, коды которых помещались 7 бит (128 символов), старший бит использовался для контроля ошибок. Со временем, кодировку дополнили до 256 символов, коды первый 128 не изменились.  ASCII стала восприниматься как половина 8-битной кодировки, а «расширенной ASCII» называли ASCII с задействованным 8-м битом (например, КОИ-8). | 7 бит;  1 байт |
| CP866 (КОИ-8 Н1) | 8-битная кодировка, работающая с **кириллицей**, содержит символы **псевдографики**. Используется в **консоли CMD**, для записи в файловой система FAT. Установлена как ГОСТ в России. | 1 байт |
| WINDOWS-1251 | Стандартная кодировка **кириллицы в Windows**. Наиболее полная из кириллистических 8-битных. **Нет псевдографических символов**. В некоторых программах есть проблема с буквой «Я» (из-за проблем с восьмым битом) | 1 байт |
| KOI8-R (CP878) | Стандарт **кириллистической** кодировки в 1990 году. Совместима с ASCII. Позиции символов кириллицы соответствуют **фонетическим аналогам** в английском алфавите из нижней части таблицы. Отказ от алфавитного порядка **решал проблему срезаемого восьмого бита** – кириллистический заменялся на читабельный латинский. Это означает, что если в тексте, написанном в КОИ-8, для каждого символа убрать по одному биту слева, то получится относительно читаемый текст, подобный транслиту. Например, слова «Русский Текст» превратятся в «rUSSKIJ tEKST». | 1 байт |
| UTF-8 | (Unicode Transformation Format, 8-bit) Распространенный стандарт кодирования текста, позволяющий **компактно хранить и передавать** символы Юникода. Используется **переменное количество бит** (от 1 до 4). Обеспечена полная **обратная совместимость с 7-битной ASCII**. Доминирующая в веб-пространсве. | 1-4 байт |
| UTF-16 | Способ кодирования символов из Юникода с помощью двух последовательных **16-битных юнитов** (слов). Позволяет записать 1,112,064 символа. Каждый символ записывается одним или двумя юнитами (суррогатная пара). | 2-4 байт |
| UTF-32 | Один из способов кодирования символов Юникода, использующий для кодирования любого символа **ровно 32 бита**. Остальные кодировки, UTF-8 и UTF-16, используют для представления символов переменное число байтов.  Преимущество: символы Юникод **непосредственно индексируемы**. Получение n-ой кодовой позиции является операцией, занимающей одинаковое время. Недостаток: **неэффективное использование пространства**. | 4 байта |

|  |  |
| --- | --- |
| Размер в UTF-8 | Представленные классы символов |
| 1 байт | ASCII, в том числе латинский алфавит, простейшие знаки препинания и арабские цифры. |
| 2 байта | кириллица, расширенная латиница, арабский, армянский, греческий, еврейский и коптский алфавит; сирийское письмо, тана, нко; МФА; некоторые знаки препинания. |
| 3 байта | все другие современные формы письменности, в том числе грузинский алфавит, индийское, китайское, корейское и японское письмо; сложные знаки препинания; математические и другие специальные символы. |
| 4 байта | музыкальные символы, смайлы, редкие китайские иероглифы, вымершие формы письменности. |
| 5 и 6 байт | Изначально кодировка UTF-8 допускала использование до 6 байтов для кодирования одного символа, однако в ноябре 2003 года стандарт RFC 3629 запретил использование 5 и 6 байтов. Это было сделано для совместимости с UTF-16. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Файловые системы (+размеры файла, тома, число файлов, длина имён) | | | | | |
| название | **размер файла** | **размер тома** | **число файлов** | **длина имен** | **Где используются** |
| FAT12 | ограничен размером тома **32 Мб** | **32 Мб** | **4,068** | **8.3** или **255** символов | Дискеты |
| FAT16 | ограничен размером тома **2 Гб** | **2 / 4 Гб** | **216** 65,536 | Диски небольшого объема |
| FAT32 | **4 Гб** | **2 / 16 ТБб** | **228**  268,13,300 | Цифровые камеры, MP3 плееры (в т.ч. iPod) |
| (File Allocation Table) Самая примитивная файловая система, применяемая в флеш-накопителях, дискетах, картах памяти. Все FAT нежурналируемые. Пространство накопителя содержит: загрузочный сектор, 2 копии таблицы FAT и сами данные. Число обозначает разрядность дискового ядра (т.е. количеством бит, отведенных для хранения номера кластера). Изначально не поддерживала иерархическую систему. На самом деле на 32, а 28 бит. | | | | | |
| NTFS4 | **264-210 байт** | **232- 1 кластер**  размер кластера =  512 байт – 64 Кб | **232-1** | **255**  16-битовых **слов** в UTF-16 | Windows |
| NTFS5 |
| (New Technology File System) Как и в FAT есть таблица MFT, но: 1) таблица хранится где угодно (а не по заданному адресу). Таблица – данные; 2) она содержит не только ссылки, но и сами данные, если они вмещаются; 3) может себя дефрагментировать; 4) при изменении файлов все записывается в LogFile; 5) есть права доступа; 6) опция сжатия файлов; 7) форматирование тома под зашифрованную ФС; 5) у файла/объекта есть свои атрибуты. | | | | | |
| Ext2 | 16 Гб – **2 Тб** | 2 Тб – **32 Тб** | **1018** | **255 байт** | Флеш-карты и SSD |
| Ext3 | **2 Тб** | **32 Тб** | Указывается при создании |  |
| Ext4 | **16 тебибайт** (244 байт) | **1 ЭиБ** (ограничен 16 Тб) | **До 4 млрд**  Указывается при создании |  |
| (Extended File System) Первая файловая система, разработанная специально для ОС на ядре Linux.  EXT2 не журналируема (преимущество, если кол-о циклов записи ограничено). EXT3 – журналируема. | | | | | |
| XFS | **8 ЭиБ– 1 байт** (8\*260байт – 1 байт) | **16 эксбибайт** | **264** | **255 байт** | IRIX, CentOS 7 |
| Высокопроизводительная 64-битная журналируемая файловая система, созданная компанией Silicon Graphics. О тличается от других файловых систем тем, что она изначально была рассчитана для использования на дисках большого объёма (более 2 терабайт). Поддерживается в Linux.  В XFS уменьшена фрагментация за счет отложенного выделения места. | | | | | |
| HFS | **2 Гб** | **2 Тб** | **65,535** | **31** символ | Дискеты, жесткие диски, CD-ROM. |
| (Hierarchical File System) Разработана Apple Computer для компьютеров с установленной MacOS.  HFS+ - улучшенная версия, поддерживающая файлы большего размера (ограничения составляют 16ЭБ на максимальный размер файла и тома) и использующая символы Unicode (вместо Mac OS Roman) в именах файлов. | | | | | |
| NFS | (Network File System) протокол сетевого доступа к файловым системам. Используется на всех современных системах Linux для объединения (на логическом уровне) файловых систем отдельных компонентов в единое целое. Позволяет подключать (монтировать) удаленные файловые системы через сеть. | | | | |
| SMB | (Server Message Block) сетевой протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессорного взаимодействия. | | | | |
| CIFS | (Common Internet File System) сетевой протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессорного взаимодействия в Windows NT. | | | | |
| F2FS | **3,94 ТБ** | **16 ТБ** | от размера тома | **255 байт** | Flash, SSD, карты памяти, флэш-чипы. |
| F2FS (Flash-Friendly File System) разработана специально с учётом специфики флеш-памяти и учитывает такие особенности, как неизменное время доступа и ограниченный ресурс количества циклов перезаписи данных. Поддержка файловой системы F2FS включена в ядро Linux начиная с 3.8.  Особенности:  - Хранение структур данных организовано в форме журнала (лога), а при обновлении информации используется механизм копирования при записи (Copy-On-Write, COW) — при изменении данные не перезаписываются, а сохраняются в новом месте. Метод работы F2FS позволит существенно продлить жизнь флеш-накопителей, поскольку файловая система учитывает внутреннюю геометрию расположения чипов в носителе и работу контроллера. Для снижения износа флеш-накопителя данные по возможности распределяются равномерно, сводя к минимуму повторную запись в одни и те же блоки. С этой целью используется алгоритм последовательного заполнения накопителя, при котором новые данные всегда записываются только в области, следующие после ранее записанных данных, без оглядки на возможную фрагментацию. После достижения конца тома запись начинается с начала, занимая, по возможности, освобождённые блоки. Для исключения конфликтов с логикой контроллера накопителя, в F2FS учитывается специфика работы прослойки FTL (Flash Translation Layer), выполняющего на многих накопителях подобную задачу по равномерному заполнению.  - Для обеспечения целостности используется модель с фиксацией точек и возможностью отката изменений (roll-back) в случае возникновения проблем.  - Для адаптации F2FS к различным видам флеш-накопителей, отличающихся своими характеристиками в зависимости от внутренней геометрии и схемы управления, предусмотрен широкий спектр параметров для управления структурой распределения данных в разделе и предоставлена возможность выбора различных алгоритмов очистки и выделения блоков.  - Раздел F2FS формируется из сегментов размером 2 МБ, сегменты группируются в секции, которые в свою очередь объединяются в зоны.  - В процессе разработки F2FS учтены проблемы предыдущих специализированных файловых систем на основе структур в форме журнала и приложены все усилия для устранения известных недостатков, таких как большое потребление памяти и высокие накладные расходы при выполнении операций очистки.  - Файловая система F2FS защищена от «эффекта снежного кома», проявляющегося при использовании Wandering-деревьев: в ситуации, когда вместо перезаписи создаются новые элементы (меняется номер блока), для деревьев, в которых родительский узел ссылается на дочерние узлы, изменение узла приводит к перестройке всех вышележащих узлов.  - Для ускорения выполнения операций в процессе работы индексы c информацией о распределении данных хранятся в оперативной памяти.  - Для выполнения операций сборки мусора реализован специальный сборщик, выполняющийся в фоне в моменты простоя системы.  - В F2FS поддерживается как традиционная для UNIX схема разграничения доступа, так и такие расширенные механизмы, как xattr и POSIX ACL. | | | | | |
| HDFS | (Hadoop Distributed File System) — файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера. Все блоки в HDFS (кроме последнего блока файла) имеют одинаковый размер, и каждый блок может быть размещён на нескольких узлах, размер блока и коэффициент репликации (количество узлов, на которых должен быть размещён каждый блок) определяются в настройках на уровне файла. Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов. Файлы в HDFS могут быть записаны лишь однажды (модификация не поддерживается), а запись в файл в одно время может вести только один процесс. Организация файлов в пространстве имён — традиционная иерархическая: есть корневой каталог, поддерживается вложение каталогов, в одном каталоге могут располагаться и файлы, и другие каталоги.  Развёртывание экземпляра HDFS предусматривает наличие центрального узла имён (англ. name node), хранящего метаданные файловой системы и метаинформацию о распределении блоков, и серии узлов данных (англ. data node), непосредственно хранящих блоки файлов. Узел имён отвечает за обработку операций уровня файлов и каталогов — открытие и закрытие файлов, манипуляция с каталогами, узлы данных непосредственно отрабатывают операции по записи и чтению данных. Узел имён и узлы данных снабжаются веб-серверами, отображающими текущий статус узлов и позволяющими просматривать содержимое файловой системы. Административные функции доступны из интерфейса командной строки. | | | | |

Ограничение по размерам у сетевых протоколов зависят от того, какая реальная файловая система находиться под протоколами.

|  |  |  |
| --- | --- | --- |
| Скорость файловых операций (+сравнение и объяснение) | | |
| чтения | | Выше, чем записи, так как не требуется искать свободное место на диске; необходимо просто найти первый кластер. |
| записи | | Медленнее, чем чтение, так как тратится время на поиск свободного места (куда поместить кластер). |
| маленьких файлов | | Скорость высокая, так как они занимают мало места. |
| больших файлов | | Скорость ниже из-за фрагментации. |
| фрагментированных и не очень | | Чтение и запись на фрагментированном диске выполняется медленно, так как нужно собирать файл «по кусочкам». |
| первый и второй раз | | Второй раз быстрее, так как файлы кэшируются. |
| проверки (scandisk, chkdsk, fsck) | | **chkdsk** only checks the filesystems that are supported in windows, Fat32 and NTFS as the main ones.  **fsck** checks, Fat32, NTFS (Via ntfs-3g), ext2, ext3, ext4, reiser, xfs and a whole bunch of other ones.  **scandisk** checks and repairs file systems errors on a disk drive, while the system starts.  Scandisk медленнее, но более точный (устарела). С исправлением ошибок дольше. |
| полного форматирования | Скорость форматирования зависит от скорости чтения/записи и от скорости работы используемого алгоритма.  Т.е. в разных программах алгоритм форматирования может быть реализован по-разному.  При полном форматировании происходит не только запись загрузочного сектора и пустой таблицы файловой системы, но и выполняется запись нулей во все сектора диска. К тому же происходит проверка всех секторов диска, и сбойные сектора помечаются особым способом и в дальнейшем не используются для записи данных, поэтому после полного форматирования объем диска может несколько уменьшиться. Так как по сути нужно заполнить весь диск нулями, процесс полного форматирования происходит очень долго, особенно на жестких дисках (для терабайтного диска он может занять до 2-3 часов).  В случае с SSD система форматирует диски иначе — для быстрого форматирования используется команда TRIM: при ее использовании контроллер SSD затирает все данные на накопителе и пересоздает список секторов. То есть для SSD быстрое форматирование выполняет по сути то же самое, что для HDD — полное. | |
| поиска по имени | | Быстро, так как просматривается только дерево файлов (иногда применим бинарный поиск).  В NTFS быстрее, чем в FAT, так как там двоичный поиск. |
| поиска по фрагменту | | Долго, так как необходимо смотреть на наличие фрагмента в каждом файле.  В NTFS быстрее, так как каждому фрагменту соответствует запись. |

|  |  |  |
| --- | --- | --- |
| Скорость устройств (+сравнение и объяснение) | | |
| RAM | ~ 1-20 Гб /с | Напрямую связана с процессором, работает через регистры и кэш |
| HDD | ~ 80-160 Мб/с | Работает медленнее, чем SSD, из-за наличия механических деталей |
| SSD | ~ 200-500 Мб/с |  |
| Flash | ~ 10-50 Мб/с | К примеру, USB2.0 работает на 4 контактах. 2 на питание, 2 на информацию. У SSD через sata гораздо больше контактов. Пропускная способность разъема Flash ниже => ниже скорость |
| Сеть | 10 Мбит - 10 Гбит |  |

|  |  |  |
| --- | --- | --- |
| Специальные файлы Windows (+кем и как используются, где лежат, как создать) | | |
| Псевдоустройства | | |
| con | консоль |  |
| prn | принтер |  |
| aux | звук (системный динамик) |  |
| nul | путь вникуда |  |
| Корень загрузочного диска | | |
| ntldr | Начальный загрузчик систем на базе Windows NT (до XP и 2003 включительно). |  |
| boot.ini | Инициализация при загрузке (ntldr использует этот файл), содержит список ОС и их параметры загрузки. |  |
| Файл подкачки | | |
| pagefile.sys | Файл подкачки. В файл pagefile.sys при нехватке памяти отправляются обычные приложения, которые установленные из разных источников. А в swapfile.sys — встроенные приложения Windows 10 и приложения, установленные из Магазина. | корневая папка локального диска. |
| %windir%\System32\ | | |
| ntoskrnl.exe | Ядро Windows. | System32 |
| hal.dll | Одна из частей ядра - слой аппаратных абстракций, используется для абстрагирования основной части ядра от особенностей аппаратуры. | System32 |
| ntdll.dll | Главная системная библиотека пользовательского режима(базовый набор api), является связующим звеном между API и ядром. | System32 |
| Файлы реестра (%WinDir%\System32\config) | | |
| default | Настройки пользователя для экрана входа в систему. | куст HKCU\.Default |
| SAM | Информация подсистемы безопасности о пользователях и группах, в том числе зашифрованные пароли. | куст HKLM\SAM |
| system | Информация о установке системы, устройствах, драйверах, службах. | куст HKLM\SYSTEM |
| Поток | | |
| :Zone.Identifier | Поток, содержащий текстовые данные. раметр ZoneId с числом означает зону, откуда прибыл файл на компьютер. Допустимы следующие значения:  0 — локальный компьютер  1 — интранет  2 — доверенный источник  3 — интернет  4 — недоверенный источник  При значении 3 система выдаст предупреждение. Если ZoneId содержит значение 4, открытие файла блокируется. | Поток создаётся в момент сохранения файла на жёсткий диск у неподписанных программ |

Если сделать файл подкачки размера 0. Когда будет использовано много памяти, при открытии чего-нибудь, система открыть не даст и выдаст ошибку о нехватке памяти «Ошибка выделения памяти».

|  |  |  |
| --- | --- | --- |
| Специальные файлы GNU/Linux (+кем и как используются, где лежат, как создать) | | |
| zero | Источник **нулевых** байт. При чтении этого файла никогда не достигается его конец. Любые данные, записанные в /dev/zero, будут игнорированы, а сама запись завершается успешно — точно так же, как и при записи в /dev/null. | /dev/ |
| null | Данные, записываемые в /dev/null - **пустой** специальный файл, на самом деле просто **игнорируются**. При чтении из пустого специального файла всегда возвращается 0 байт. | /dev/ |
| [h|s]d[a|b|...] | **Жёсткие диски** EIDE/IDE известны как "hd" и именуются в алфавитном порядке, начиная с "a" (hda, hdb, hdc и т.д.); **SCSI/SATA диски** чаще всего называются "sd" и также именуются в алфавитном порядке, начиная с "a" (sda, sdb, sdc ... и т.д.). Разделы дисков нумеруются, начиная с номера 1 ( hda1, hdb1, hdb2, hdb3, hdc1, sda1, sda2). | /dev/ |
| random | Генерирует случайные числа с **высокой степенью случайности**, но слишком много генерировать нельзя, иначе будет тормозить. | /dev/ |
| urandom | Генерирует **псевдослучайные числа**. Стоит использовать этот генератор для общих целей. | /dev/ |
|  | | |
| passwd | **База данных о пользователях** в Unix системах является текстовым файлом /etc/passwd (называется файл паролей (password file)), который хранит все имена пользователей и сведения о них. | /etc/ |
| group | ASCII файл, который описывает **группы**, членами которых являются пользователи. | /etc/ |
| shadow | В данном файле также хранятся **хеши паролей** пользователей из файла /etc/passwd/. Доступен для чтения только root, что повышает защиту. | /etc/ |
| sudoers | **Список пользователей**, с указанием того, **что они могут выполнять** (настройки sudo). | /etc/ |
|  | | |
| fstab | содержит информацию обо **всех разделах жесткого диска и других носителях информации в компьютере.** | /etc/ |
| mtab | **Когда программа mount подключает файловую систему, она дописывает соответствующую строку в /etc/mtab**. Когда umount отключает файловую систему, из этого файла соответствующая строка удаляется. | /etc/ |
| crontab | Конфигурационный файл демона cron. **Список процессов с датой и временем их запуска.** | /etc/ |
|  | | |
| cpuinfo | Подробная информация о **ядрах и процессоре**. | /proc/ |
| swaps | Показывают все **разделы и все пространства подкачки**. | /proc/ |
| version | Предоставляет данные о **версии ядра**. | /proc/ |
| uptime | Информация про uptime системы (**время работы** системы с момента последней загрузки). | /proc/ |
| meminfo | Предоставляет данные об **использовании оперативной памяти**. | /proc/ |
| partitions | Подробная информация о **разделах, доступных в системе**. | /proc/ |

|  |  |
| --- | --- |
| **Службы, демоны и системные процессы** | |
| **Windows** | |
| **idle** | **Бездействие системы** – процесс, выполняемый процессором в пространстве ядра операционной системы в случае, если нет других процессов, которые процессор мог бы выполнять. |
| **Windows Time** | Устанавливает **дату и время** (W32Time синхронизирует дату и время для всех компьютеров, работающих в сети Microsoft Windows). |
| **Print Spooler** | Диспетчер очереди печати, **управляет очередями печати** в системе, а также **взаимодействует с драйверами принтеров** и **компонентами ввода-вывода**. |
| **Logical Disk Manager** | [**Менеджер логических дисков**] Позволяет создавать динамические диски (способные увеличивать свой размер). |
| **Automatic Updates** | **Автоматическое обновление** Windows. |
| **Task Scheduler** | **Планировщик задач**, запуск программ по расписанию. |
| **Event Log** | Регистрирует **сообщения событий**, выдаваемые программами и самой Windows.  Эту службу невозможно остановить. Посмотрев на записи, можно, например, узнать какие программы «вылетают». |
| *Специальные программы, которые запускают те или иные службы. Разные службы запускаются одной программой (вызов с ключом).* | |
| **svchost.exe** | Главный **процесс для служб, запускаемых из DLL** (динамически загружаемых библиотек). Использование единого процесса для работы нескольких сервисов позволяет уменьшить затраты оперативной памяти и процессорного времени. |
| **dllhost.exe** | **Обработка COM+ процессов** (Component Object Model); **Управление** **приложениями, использующими DLL** (динамически загружаемые библиотеки); **Контроль IIS процессов** (Internet Information Services – проприетарный набор сервисов для нескольких служб Интернета от Microsoft);  **Загрузка среды .NET** (программная платформа). |
| **services.exe** | **Диспетчер управления службами**, ответственный за запуск и остановку системных служб и взаимодействие с ними. |
| **spoolsv.exe** | (*Spooler SubSystem App*) Процесс, ответственный за управление заданиями на **печать** и передачу **факсимильных сообщений**. |
| **winlogon.exe** | Процесс, ответственный за **начало** и **завершение** **сеанса** пользователя. |
| **lsass.exe** | (*Local Security Authority Subsystem Service*) [**Сервер проверки подлинности локальной системы безопасности**] Локальный сервер проверки подлинности, порождающий процесс, ответсвенный за проверку пользователей в службе *Winlogon.* |
| **csrss.exe** | (*Client/Server Runtime Subsystem*) [**Процесс исполнения клиент-сервер**] (обработка консоли, графический режим включения ОС). Завершение этого процесса приведет к сбою, удаление – к синему экрану.  Данный процесс в Windows 7, 8 и Windows 10 отвечает за консольные (выполняемые в режиме командной строки) программы, процесс выключения, запуск другого важного процесса — conhost.exe и другие критические функции системы. |
| **mdm.exe** | (*Machine Debug Manager*) **Дебаг-менеджер** (используется Windows NT Option Pack и Microsoft Developer Studio чтобы обеспечить функции отладки приложений). |
|  | |
| **windefend** | **Служба защитника** Windows. Защитник Windows это антивирус от Microsoft встроенный в Windows 10, так же как в Windows 8 и 7. Он включён по умолчанию и автоматически выключается, если вы установите другой антивирус, при условии что операционная система его распознает. |
| **WUAUSERV** | Стандартная служба обновления Windows, которая используется для **сканирования компьютера, поиска, закачки и установки новых обновлений** с узлов Windows Update или локальных WSUS серверов. |
| **Unix** | |
| **init** | (*initialization*) Программа, **запускающая все остальные процессы**. |
| **(r)syslogd** | Демон, отвечающий за **журналирование**. Ведет запись событий о работе кластера в файл, согласно параметрам, указанным в /etc/syslog.cong (системный журнал). |
| **klogd** | (*kernel log*) Перехватывает и регистрирует все **сообщения ядра**. |
| **cron** | (*Chronos god*) Демон-**планировщик задач**, использующийся для периодического выполнения заданий в определенное время. |
| **systemctl** | Во время загрузки systemd **инициализирует все прочие сервисы и управляет их работой** вплоть до выключения. При необходимости мы можем запустить или остановить нужный процесс, назначить или отменить его автоматический запуск или даже создать собственный сервис. |
| **inetd** | (*internet service*) Демон, при необходимости **запускающий** **некоторые другие** **сетевые процессы**. Прослушивает указанные порты и при подключении клиента запускает соответствующий процесс. |
| **cupsd** | (*Common Unit Printing System*) **Управляет заданиями печати** и обеспечивает **сетевую печать**. |
| **udev** | (*userspace /dev*) Диспетчер устройств, который Обслуживает файлы устройств в каталоге /dev;  **Обрабатывает все действия, выполняемые в пространстве пользователя** при добавлении/отключении внешних устройств, включая загрузку прошивки |

|  |  |
| --- | --- |
| Числа (+диапазоны значений) | |
| Частота процессора | Это — тактовая частота процессора (Core Speed).  Тактовая частота — это количество тактов (операций) процессора в секунду. Тактовая частота процессора пропорциональна частоте шины.  Как правило, чем выше тактовая частота процессора, тем выше его производительность.  Тактовая частота показывает нам, сколько процессор может произвести вычислений в единицу времени.  Тактовая частота современных процессоров, в основном, составляет 1,0-4ГГц. |
| Размер оперативной памяти | RAM/ОЗУ – энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.  Ограничения по объёму оперативной памяти:1) ограничения на длину адреса со стороны процессора, 2) ограничения со стороны ОС. 4-гб при 32-разрядном адресе.  Размер ОП в байтах - От нескольких килобайт на старых компьютерах, до 1-64 гб в среднем сейчас. |
| Размер жёсткого диска | максимальное количество информации, которое способен вместить жёсткий магнитный диск.  256ТБ – Максимальный размер файловой системы NTFS (теор. – 1ЙБ).  3.5 Мб - 10 Тб |
| Размер кластера | единица хранения данных в таблице размещения файлов, объединяющая группу секторов. Например, на дисках с размером секторов в 512 байт, 512-байтный кластер содержит один сектор, тогда как 4-килобайтный кластер содержит восемь секторов.  Стандартно - 4 кб, от 512 б до 64 Кб |
| Количество открытых файлов | DOS - 255, в Windows NT 2048 / 16384, Linux ~600 000 (общесистемно, на процесс порядка тысяч). |
| Количество запущенных процессов и потоков | зависит от объёма ОП, порядка тысяч и десятков тысяч (~ до 30 000).  Количество процессов определяется числом строк в таблице процессов, максимум 32768, количество потоков то же. |
| Размер файла подкачки | системный файл на жестком диске компьютера, который Windows использует, чтобы компенсировать нехватку оперативной памяти, если приложениям ее не хватает. Вместо того, чтобы закрыть приложение, которому не хватает RAM, Windows скидывает его в файл подкачки и при необходимости возвращает обратно. Обычно объём файла подкачки задаётся в 1,5 раза больше общего объёма оперативной памяти. |
| Размер ядра системы (в памяти и на диске) | на диске ~ 3-5 Мб, в памяти порядка десятков / сотен мегабайт (распаковано). Примерно 3 Мб, в памяти ядро резервирует 2 Гб у каждого процесса |
| Размер невыгружаемого страничного пула | (область памяти системы, предназначенная для объектов, которые нельзя выгружать на диск и которые должны оставаться в выделенной им физической памяти) - порядка десятков / пары сотен Мб, зависит от размера памяти и битности системы.  Норма: 90-150 Мб; максимум 4 Гб (но не стоит). |
| Количество файловых операций в час | есть параметр IOPS (операций в секунду), диапазон ~ hdd 100 -2000, ssd 8000- 500 000 |
| Размер видеопамяти | Объем видеопамяти – суммарная емкость памяти видеокарты, которая хранит изображение для вывода на монитор.  64 Кб - 4 Гб |
| Количество клавиш на клавиатуре | 64-200 (103 у меня). |
| Количество клавиш на мышке | 2-20 |

Файл подкачки: В файл pagefile.sys при нехватке памяти отправляются обычные приложения, которые установленные из разных источников. А в swapfile.sys — встроенные приложения Windows 10 и приложения, установленные из Магазина.

|  |
| --- |
| Адреса в вебе (+что там) |

[**http://wikipedia.org/wiki/First\_Draft\_of\_a\_Report\_on\_the\_EDVAC**](http://wikipedia.org/wiki/First_Draft_of_a_Report_on_the_EDVAC)

**Документ фон Неймана (о принципах построения компьютеров)**

Первый проект отчёта о EDVAC   
незаконченный 101-страничный документ, написанный [Джоном фон Нейманом](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD,_%D0%94%D0%B6%D0%BE%D0%BD_%D1%84%D0%BE%D0%BD) и распространённый 30 июня 1945 года [Германом Голдстайном](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%B4%D1%81%D1%82%D0%B0%D0%B9%D0%BD,_%D0%93%D0%B5%D1%80%D0%BC%D0%B0%D0%BD), куратором со стороны Армии США проектов [ЭНИАК](https://ru.wikipedia.org/wiki/%D0%AD%D0%9D%D0%98%D0%90%D0%9A) и [EDVAC](https://ru.wikipedia.org/wiki/EDVAC). В документе впервые опубликовано описание логического устройства вычислительной машины с хранимой в памяти программой. Эта концепция в дальнейшем стала известна как «[архитектура фон Неймана](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%84%D0%BE%D0%BD_%D0%9D%D0%B5%D0%B9%D0%BC%D0%B0%D0%BD%D0%B0)».

[**http://awards.acm.org/**](http://awards.acm.org/)

**награды Asociation for Computing Machinery** (виды наград, награжденные, информация по церемониям), в том числе и Тьюринговская премия

[**http://www.levenez.com/**](http://www.levenez.com/)

**Эрик Левенец**

на сайте выложена история Unix, Windows и языков программирования в виде диаграмм

[**http://www.microsoft.com/technet/sysinternals/**](http://www.microsoft.com/technet/sysinternals/)

веб-узел, на котором размещены **усовершенствования сервисных программ Windows** (системные утилиты)

[**http://cs.usu.edu.ru/docs/pe/**](http://cs.usu.edu.ru/docs/pe/)

**пособие по PE-файлам**

(Portable Executable Files, исполняемые файлы для любой архитектуры процессора)

Структура исполняемых файлов Win32 и Win64

[**http://cs.usu.edu.ru/study/presentfs.html**](http://cs.usu.edu.ru/study/presentfs.html)

**пособие по файловым системам**

[**http://kernel.org/**](http://kernel.org/)

ядро **linux** (официальный сайт)

[**http://freebsd.org/**](http://freebsd.org/)

**FreeBSD** (официальный сайт)

FreeBSD – an operating system used to power modern servers, desktops, and embedded [platforms](https://www.freebsd.org/platforms/)

Свободная Unix-подобная операционная система. FreeBSD хорошо зарекомендовала себя как система для построения интранет и интернет-сетей и серверов. Она предоставляет надёжные сетевые службы и эффективное управление памятью.

[**http://debian.org/**](http://debian.org/)

**debian** (официальный сайт)

[**http://ubuntu.com/**](http://ubuntu.com/)

**ubuntu** (официальный сайт)

[**http://microsoft.com/**](http://microsoft.com/)

официальная страница **Microsoft**

[**http://redhat.com/**](http://redhat.com/)

**Red Hat** (официальная страница)

американская компания, производитель программного обеспечения на основе операционной системы Linux

[**http://www.gnu.org/philosophy/philosophy.html**](http://www.gnu.org/philosophy/philosophy.html)

**философия проекта GNU**