**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, применением теории операторов к квантовой механике, а также как участник Манхэттенского проекта и как создатель теории игр и концепции клеточных автоматов.

**Фред Брукс \*)**

американский учёный в области теории вычислительных систем, автор книги «Мифический человеко-месяц или Как создаются программные системы» (наблюдения в ходе разработки OS/360). Управлял разработкой OS/360 в IBM. Награждён в 1999 году премией Тьюринга за исторически значимый вклад в компьютерную архитектуру, операционные системы, и проектирование программного обеспечения. Всего 9 наград.

**Эдсгер Дейкстра \*)**

отстаивал необходимость математического подхода к программированию, активно участвовал в разработке языка программирования Алгол и написал первый компилятор Алгол-60. Будучи одним из авторов концепции структурного программирования, он 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, известный как алгоритм Дейкстры. В 1972 году Дейкстра стал лауреатом премии Тьюринга (за Алгол и публикации).

**Даг Энгельбарт \*)**

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. Премия Тьюринга в 1997 за вдохновляющее предвидение будущего развития интерактивных вычислений и изобретение ключевых технологий помогающих это предвидение реализовать.

**Кен Томпсон \*)**

создатель Си и Unix. Так же создал язык B (предшественник Си), версию редактора QED для CTSS (с регулярными выражениями) и ed (стандартный редактор Unix). Премия Тьюринга совместно с Ритчи в 1983 за разработку общей теории операционных систем и в частности за создание Unix.

**Дэнис Ритчи \*)**

создатель ALTRAN, B, BCPL, Си, Multics и Unix. Соавтор книги «Язык программирования Си». Премия Тьюринга совместно с Томпсоном в 1983 за разработку общей теории операционных систем и в частности за создание Unix. Сейчас участвует в разработке ОС Plan 9 и Inferno, а также языка программирования Limbo.

**Кен Олсен**

онователь DEC, создатель PDP-1 (Personal Digital Processor).

**Гарри Килдалл**

разработчик CP/M (Control Program for Microcomputers, 1975). CP/M состояла из базовой системы ввода-вывода (BIOS), дисковой ОС (BDOS), командная строка (CCP). Код быд аппаратно независимым, поэтому легко портировался.

**Тим Паттерсон**

разработчик 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 архитектуру Intel. Закрыл убыточные проекты Apple. При нем создается iPod, iPhone, iTunes Store и MacBookAir.

**Дейв Кэтлер**

Создатель Windows NT.

**Ричард Столлмэн**

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 GNU Emacs, Коллекции компиляторов GNU (GCC) и Отладчика GNU (GDB).

**Эндрю Таненбаум**

профессор Амстердамского свободного университета. Создатель Minix и RFID-вируса. Главный разработчик «Amsterdam Compiler Kit». 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).

***\*) - Тьюринговская премия***

**Принципы устройства современных вычислителей (+примеры нарушений):**

**двоичное кодирование**

инфа кодируется с помошью одного набора кодов состоящих из 1 и 0 (двоичная система), но есть разные кодировки, поэтому не всегда один код соответсвует одной и той же инфе. Машинка "СЕтунь", сделанная в СССР, основана на троичной логике.

**адресуемость памяти (RAM)**

каждая ячейка памяти имеет АДРЕС,по которому из неё можно взять инфу.

**программное управление (хранимая в памяти программа)**

процессор своей шлангой бегает по инструкциям и делает одну команду за другой при этом есть некий счётчик (ip), который содержит адрес текущей команды.

**однородность памяти**

и программы, и инструкции хранятся в одной памяти. Нарушается хотя бы наличием кэш-памяти, не говоря уже о том, что она разделена на память для инструкций и память для данных.

**наличие пространства ввода/вывода**

некоторые адреса оперативной памяти, известные процессору, на самом деле в RAM не ведут, а ведут в контроллеры внешних устройств.

**аппаратная поддержка стека (LIFO)**

наличие в процессоре регистра SP, указывающего на вершину стека.

**механизм прерываний (внешних и внутренних)**

прерывания - сигналы, по которым управление передается обработчику прерываний (распознание прерывания, предача управления обработчику, корректное возвращение к прерванной программе).

**уровни привилегий, привилегированные команды (защищённый режим)**

4 кольца (0,1,2,3) защиты процессоров семейства х86. Самая крутизна - 0. Там работают дрова (обычно, к примеру в Винде ХР) и ядро. Всё остальное работает на 3-ем. В зависимости от уровня в контексте данного потока (процесса) может быть выполнена данная инструкция или нет. Если ктото хочет больше чем надо то получается #GP (внутреннее прерывание GeneralProtection). В RISC и MISC процах такого нет.

**Типы прерываний (+примеры, происхождение, обработка и применение):**

**Внутренние**

(иключения) возникают в ходе выполнения какой-то инструкции. Внутри такта. При делении на ноль,ошибке защиты памяти...

**Внешние**

(аппаратные) поступают с перифирии: устройства, оператор за терминалом. Такие прерывания поступают на обработку между выполнением двух соседних инструкций.

**Немаскируемые**

прерывания с наивысшим приоритетом, т.к. его нельзя замаскировать (когда создается очередь прерываний, то выполняется самое главное, а остальные маскируются). Синий экран - пример.

**Программные**

способ вызова функций (подпрограмм), т.е. переход на другие инструкции для их выполнения.

**Команды CMD (+что делают, что возвращают, опции, расширения):**

**ver**

версия Windows

**date**

показывает текущую дату и спрашивает ее изменение (/T - не будет спрашивать)

**time**

то же самое, что date, но работает с текущим временем

**at**

Команда AT предназначена для запуска команд и программ в указанное время

по определенным дням. Для использования команды AT необходимо, чтобы была

запущена служба расписаний.

AT [\\имя\_компьютера] [ [ИД] [/DELETE] | /DELETE [/YES]]

AT [\\имя\_компьютера] время [/INTERACTIVE]

[ /EVERY:день[,...] | /NEXT:день[,...]] "команда"

\\имя\_компьютера Имя удаленного компьютера. Если этот параметр опущен,

используется локальный компьютер.

код Порядковый номер запланированной задачи.

/delete Отмена запланированной задачи. Если код задачи опущен,

отменяются все задачи, запланированные для указанного

компьютера.

/yes Отмена запроса на подтверждение при отмене всех

запланированных задач.

время Время запуска команды.

/interactive Разрешение взаимодействия задачи с пользователем,

работающим на компьютере во время запуска задачи.

/every:день[,...] Запуск задачи осуществляется по указанным дням недели или

месяца. Если дата опущена, используется текущий день

месяца.

/next:день[,...] Задача будет запущена в следующий указанный день недели

(например, в следующий четверг). Если дата опущена,

используется текущий день месяца.

"команда" Команда Windows NT или имя пакетного файла.

**cmd**

*запуск новой копии cmd.exe "cmd /a||/u(вывод результатов в формате ANSI||Unicode) /q(отключение режима ECHO) /d(отключение выполнения команд AutoRun из реестра) /e:on||off(режим расширенной обработки команд) /f:on||off(символы завершения имен файлов и папок) /v:on||off(отложенное расширение переменных среды) /s(изменение поведения после с и к) [/c||/k строка](выполнение строки с завершением||без завершения)"*

Запуск новой копии интерпретатора команд Windows.

CMD [/A | /U] [/Q] [/D] [/E:ON | /E:OFF] [/F:ON | /F:OFF] [/V:ON | /V:OFF]

[[/S] [/C | /K] строка]

/C Выполнение указанной команды (строки) с последующим завершением.

/K Выполнение указанной команды (строки) без последующего завершения.

/S Изменение поведения после /C или /K (см. ниже)

/Q Отключение режима вывода команд на экран (ECHO).

/D Отключение выполнения команд AutoRun из реестра (см. ниже)

/A Вывод результатов выполнения команд в формате ANSI.

/U Вывод результатов выполнения команд в формате UNICODE.

/T:fg Выбор цвета переднего плана/фона (более подробно см. COLOR /?)

/E:ON Разрешение расширенной обработки команд (см. ниже)

/E:OFF Запрет расширенной обработки команд (см. ниже)

/F:ON Разрешение символов завершения имен файлов и папок (см. ниже)

/F:OFF Запрет символов завершения имен файлов и папок (см. ниже)

/V:ON Разрешение отложенного расширения переменных среды с применением

символа '!' в качестве разделителя. Например, /V:ON разрешает

использовать !var! в качестве расширения переменной var во время

выполнения. Синтаксис var служит для расширения переменных

при вводе, что приводит к совсем другим результатам внутри

цикла FOR.

/V:OFF Запрет отложенного расширения переменных среды.

Чтобы указать в одной строке несколько команд, следует разделить их

символами '&&' и заключить в кавычки. Кроме того, из соображений

совместимости, /X означает то же, что и /E:ON, /Y то же, что и /E:OFF

и /R то же, что и /C. Все прочие ключи командной строки игнорируются.

Если указаны ключи /C или /K, то остальная часть командной строки после

такого ключа обрабатывается как командная строка, а обработка символов

кавычек (") ведется по следующим правилам:

1. Если выполняются все перечисленные ниже условия, то символы

кавычек в командной строке сохраняются:

- ключ /S отсутствует

- есть ровно два символа кавычек

- между ними нет других специальных символов, как то: &<>()@^|

- между ними имеются один или несколько пробелов

- строка, заключенная в кавычки, является именем исполнимого

файла.

2. В противном случае, проверяется первый символ, и если он является

символом кавычек, то он удаляется, также удаляется последний

символ кавычек в командной строке, а весь текст после этого

последнего символа кавычек сохраняется.

Если ключ /D НЕ УКАЗАН в командной строке, то при запуске CMD.EXE

выполняется проверка значений переменных REG\_SZ или REG\_EXPAND\_SZ

для следующих разделов системного реестра:

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\AutoRun

и/или

HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\AutoRun

и если одна из них или обе они присутствуют, то сначала выполняются они.

По умолчанию расширенная обработка команд включена. Чтобы запретить

расширенную обработку для конкретного вызова, используется параметр /E:OFF.

Можно включить или отключить расширенную обработку команд для всех вызовов

CMD.EXE на данном компьютере или для данного сеанса пользователя, задав с

помощью REGEDIT.EXE в системном реестре значение 0x1 или 0x0 для параметров

REG\_DWORD для следующих разделов:

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\EnableExtensions

и/или

HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\EnableExtensions

Параметры пользователя переопределяют параметры компьютера.

Параметры командной строки переопределяют соответствующие параметры реестра.

В пакетном файле аргументы SETLOCAL ENABLEEXTENSIONS или DISABLEEXTENSIONS

переопределяют параметр /E:ON или /E:OFF. Для получения дополнительных

сведений введите "SETLOCAL /?".

При расширенной обработке команд изменения и/или добавления затрагивают

следующие команды:

DEL или ERASE

COLOR

CD или CHDIR

MD или MKDIR

PROMPT

PUSHD

POPD

SET

SETLOCAL

ENDLOCAL

IF

FOR

CALL

SHIFT

GOTO

START (изменен также вызов внешних команд)

ASSOC

FTYPE

Для получения подробных сведений введите "имя\_команды /?".

Отложенное расширение переменных среды НЕ ВКЛЮЧЕНО по умолчанию. Можно

включить или отключить отложенное расширение переменных среды для конкретного

вызова CMD.EXE с помощью параметра /V:ON или /V:OFF. Можно включить или

отключить отложенное расширение переменных среды для всех вызовов CMD.EXE на

данном компьютере или для данного пользователя, задав с помощью команды

REGEDIT.EXE в системном реестре значение 0x1 или 0x0 для параметров REG\_DWORD

для следующих разделов:

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\DelayedExpansion

и/или

HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\DelayedExpansion

Параметры пользователя переопределяют параметры компьютера. Параметры командной

строки, в свою очередь, переопределяют параметры реестра.

В пакетном файле аргументы SETLOCAL ENABLEDELAYEDEXPANSION или

DISABLEDELAYEDEXPANSION переопределяют параметр /V:ON или /V:OFF. Для получения

дополнительных сведений введите "SETLOCAL /?".

Если отложенное расширение переменных среды включено, то для замены текущего

значения переменной среды во время выполнения может использоваться символ

восклицательного знака "!".

Можно включить или отключить завершение имен файлов и папок для

конкретного вызова CMD.EXE с помощью ключей /F:ON или /F:OFF.

Завершение имен можно включить или отключить для

всех вызовов CMD.EXE на данном компьютере или для данного пользователя,

с помощью REGEDIT.EXE задав значения REG\_DWORD в системном реестре

для следующих разделов:

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\CompletionChar

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor\PathCompletionChar

и/или

HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\CompletionChar

HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor\PathCompletionChar

установив их шестнадцатеричные значения равными коду управляющего символа,

используемого для конкретной функции (например, 0x4 для CTRL-D или 0x6 для

CTRL-F). Параметры пользователя перекрывают параметры компьютера.

Ключи командной строки, в свою очередь, перекрывают параметры реестра.

Если завершение включено с помощью ключа командной строки /F:ON,

то используются два управляющих символа: CTRL-D для имен папок и CTRL-F

для имен файлов. Чтобы отключить конкретный символ завершения в системном

реестре, в качестве кода специального символа используется шестнадцатеричное

значение символа пробела (0x20).

Завершение вызывается при вводе одного из этих двух специальных символов.

Функция завершения берет строку пути слева от точки ввода, дописывает

к ней символ шаблона, если таковой отсутствует, а затем строит список

путей, которые соответствуют полученному определению. После этого

выводится первый элемент этого списка соответствующих путей. Если же

список пуст, то подается звуковой сигнал и ничего не выводится.

После этого повторный ввод того же самого специального символа приводит

к циклическому перебору всех соответствующих путей. Нажатие клавиши

<Shift> при вводе управляющего символа позволяет просматривать список

путей в обратном порядке. Если изменить выведенный путь, а затем снова

ввести управляющий символ, сохраненный список очищается и строится новый

список путей. То же самое происходит, если переключиться с одного символа

завершения на другой. Единственное отличие при использовании символа

завершения файла состоит в том, что при этом для построения списка

соответствия берется и путь, и имя файла, а при использовании символа

завершения папки берется только путь. Если символ завершения используется

в одной из встроенных команд манипулирования папками (CD, MD или RD), то

всегда подразумевается символ завершения папок.

Символ завершения правильно работает и с именами файлов, содержащими

пробелы или иные специальные символы, если при этом строка соответствия

заключена в кавычки. Кроме того, если сместить точку ввода влево, а затем

использовать символ завершения внутри строки, оставшийся справа от точки

ввода текст будет отброшен.

Специальные символы, которые требуют обязательного заключения в кавычки:

<пробел>

&()[]{}^=;!'+,`~

**command**

*запуск новой копии обработчика MS-DOS command.com "command путь устройство\_ввода\_вывода /E:nnnnn /P(запрет на выход) [/C строка](выполнение строки) /MSG(сохранение ошибок в памяти)"*

Starts a new instance of the MS-DOS command interpreter.

COMMAND [[drive:]path] [device] [/E:nnnnn] [/P] [/C string] [/MSG]

[drive:]path Specifies the directory containing COMMAND.COM file.

device Specifies the device to use for command input and output.

/E:nnnnn Sets the initial environment size to nnnnn bytes.

/P Makes the new command interpreter permanent (can't exit).

/C string Carries out the command specified by string, and then stops.

/MSG Specifies that all error messages be stored in memory. You

need to specify /P with this switch.

**start**

*запуск указанной программы/команды в отдельном окне "заголовок [/d путь] /I /min(свернутое) /max(развернутое) /separate||/shared(16 разрядов) /low||/normal||/high||/realtime||/abovenormal||/belownormal(приоритет) /affinity /wait(ожидание завершения) /b(без создания нового окна) строка-команда"*

Запуск указанной программы или команды в отдельном окне.

START ["заголовок"] [/D путь] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED]

[/LOW | /NORMAL | /HIGH | /REALTIME| /ABOVENORMAL | /BELOWNORMAL]

[/AFFINITY <маска соответствия>] [/WAIT] [/B]

[команда/программа] [параметры]

"заголовок" Заголовок окна.

путь Рабочий каталог.

B Запуск приложения без создания нового окна с отключением

обработки сочетания клавиш ^C. Если приложение не

обрабатывает сочетание клавиш ^C самостоятельно,

единственным способом его прерывания является

использование сочетания клавиш ^Break.

I Новой средой станет исходная среда, переданная

cmd.exe, а не текущая среда.

MIN Запуск команды/программы в свернутом окне.

MAX Запуск команды/программы в развернутом окне.

SEPARATE Запуск 16-разрядной программы Windows в отдельной

области памяти.

SHARED Запуск 16-разрядной программы Windows в общей

области памяти.

LOW Запуск приложения с приоритетом IDLE.

NORMAL Запуск приложения с приоритетом NORMAL.

HIGH Запуск приложения с приоритетом HIGH.

REALTIME Запуск приложения с приоритетом REALTIME.

ABOVENORMAL Запуск приложения с классом приоритета ABOVENORMAL

BELOWNORMAL Запуск приложения с классом приоритета BELOWNORMAL

AFFINITY Новое приложение будет иметь маску соответствия процессора,

указанную шестнадцатеричным значением.

WAIT Запуск приложения с ожиданием его завершения.

команда/программа

Если это внутренняя команда cmd.exe или пакетный файл,

обработчик команд (cmd.exe) запускается с ключом /K.

Это означает, что окно не будет закрыто после завершения

команды.

Если это не внутренняя команда cmd.exe и не пакетный файл,

значит, это программа, которая будет запущена в графическом

или текстовом окне.

параметры Параметры, передаваемые команде/программе.

Примечание: параметры SEPARATE и SHARED не поддерживаются на 64-разрядных

платформах.

Изменения вызова внешних команд из командной строки или с помощью команды

START при включении расширенной обработки команд:

Для вызова неисполняемых файлов через механизм сопоставления типов файлов

достаточно просто ввести имя файла в командной строке. Например, команда

WORD.DOC запускает приложение, сопоставленное расширению имени файла .DOC.

Сведения о создании подобных сопоставлений из пакетных файлов приведены

в описаниях команд ASSOC и FTYPE.

При запуске 32-разрядного приложения с графическим интерфейсом пользователя

обработчик команд CMD.EXE не ожидает завершения работы приложения перед

возвратом к приглашению командной строки. Подобное новое поведение НЕ

относится к запуску приложений из пакетных файлов.

При выполнении командной строки, первым элементом которой является текстовая

строка "CMD" без расширения имени файла или указания пути, она заменяется

значением переменной COMSPEC. Это предотвращает запуск CMD.EXE из

текущей активной папки, если таковая программа там имеется.

Если первый элемент командной строки не содержит расширения имени файла,

обработчик команд CMD.EXE использует значение переменной среды PATHEXT,

чтобы определить расширения имен исполняемых файлов и порядок поиска

нужного файла. По умолчанию для переменной PATHEXT задается

значение:

.COM;.EXE;.BAT;.CMD

Обратите внимание, что этот синтаксис подобен синтаксису для переменной

PATH, то есть отдельные элементы разделяются точкой с запятой.

Если при поиске исполняемого файла нет соответствия ни одному из расширений,

выполняется проверка соответствия указанного имени папки. Если имя папки

соответствует указанному, то команда START запускает EXPLORER для этого

пути. Если такое действие выполняется из командной строки, то оно

эквивалентно выполнению команды CD /D для указанного пути.

**title**

смена заголовка окна командной строки

**prompt**

*изменение приглашения командной строки*

Изменение приглашения командной строки cmd.exe.

PROMPT [текст]

текст Новое приглашение командной строки.

Приглашение может включать обычные символы и следующие коды:

$A & (амперсанд)

$B | (вертикальная черта)

$C ( (левая круглая скобка)

$D Текущая дата

$E ESC (символ ASCII с кодом 27)

$F ) (правая круглая скобка)

$G > (знак "больше")

$H BACKSPACE (удаление предыдущего символа)

$L < (знак "меньше")

$N Текущий диск

$P Текущие диск и каталог

$Q = (знак равенства)

$S (пробел)

$T Текущее время

$V Номер версии Windows

$\_ Возврат каретки и перевод строки

$$ $ (символ доллара)

Когда расширенная обработка команд включена, команда PROMPT поддерживает

следующие дополнительные символы форматирования:

$+ Отображение нужного числа знаков плюс (+) в зависимости от текущей

глубины стека каталогов PUSHD, по одному знаку на каждый сохраненный

путь.

$M Отображение полного имени удаленного диска, связанного с именем

текущего диска, или пустой строки, если текущий диск не является

сетевым.

**pause**

приостановка выполнения и вывод press any key

**echo**

вывод сообщений и переключение режима ECHO

**cls**

чистка поля командной строки

**assoc**

*просмотр и изменение сопоставления расширения и типа файла*

Просмотр и изменение сопоставлений файлов.

ASSOC [.рсш[=[типФайла]]]

.рсш Расширение имени файла, сопоставляемое типу файлов

типФайла Тип файлов, сопоставляемый расширению имени файлов

Команда ASSOC без параметров выводит список сопоставлений типов файлов.

Если указано только расширение имени файла, выводится сопоставленный тип

файлов для расширения. Если после знака равенства не указан тип файлов,

команда удалит текущее сопоставление для указанного расширения.

**ftype**

*изменение команды/программы открытия файлов указанного типа*

Просмотр и изменение типов файлов, сопоставленных с расширением имен файлов

FTYPE [типФайлов[=[команднаяСтрокаОткрытия]]]

типФайлов Тип файлов для просмотра или изменения

команднаяСтрокаОткрытия Команда, используемая для открытия файлов

указанного типа.

Команда FTYPE без параметров выводит текущий список типов файлов, для которых

определены командные строки открытия. Если указан только тип файла, FTYPE

выводит командную строку открытия для этого типа файлов. Если после знака

равенства не указана строка открытия, FTYPE удалит текущее сопоставление для

указанного типа файлов. При вызове командной строки переменные %0

и %1 заменяются на имя файла, запускаемого с помощью сопоставления.

Вместо переменной %\* подставляются все параметры, а переменные %2, %3

и т.д. заменяются, соответственно, на первый, второй и другие параметры.

Вместо переменной %~n подставляются все оставшиеся параметры, начиная с n,

где n является числом от 2 до 9. Например:

ASSOC .pl=PerlScript

FTYPE PerlScript=perl.exe %1 %\*

Эти команды позволят вызывать обработчик команд Perl следующим образом:

script.pl 1 2 3

Если желательно избежать постоянного ввода расширения имен файлов, введите

следующую команду:

set PATHEXT=.pl;%PATHEXT%

Теперь обработчик команд вызывается еще проще:

script 1 2 3

**type**

вывод содержимого одного/нескольких файлов

**more**

*последовательный вывод данных по частям (/e - дополнительные возможности, /c - очистка экрана перед каждым выводом, /p - учет символов перевода страницы /s - сжатие пустых строк в одну /tn - замена табуляции n пробелами +n - начало вывода первого файла с n строки*

Последовательный вывод данных по частям размером 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]

[диск:][путь]имя\_файла Файл, отображаемый

по фрагментам.

имя\_команды Команда, вывод которой

отображается на экране.

/E Разрешение использования дополнительных возможностей.

/C Очистка экрана перед выводом каждой страницы.

/P Учет символов перевода страницы.

/S Сжатие нескольких пустых строк в одну строку.

/Tn Замена символов табуляции n пробелами (по умолчанию n = 8).

Стандартный набор ключей можно поместить

в переменную среды MORE.

+n Начало вывода первого файла со строки с номером n.

файлы Список отображаемых файлов. Для разделения имен файлов

в списке используйте пробелы.

Если использование дополнительных возможностей разрешено, в ответ

на приглашение -- More -- можно вводить следующие команды:

P n Вывод следующих n строк.

S n Пропуск следующих n строк.

F Вывод следующего файла.

Q Завершение работы.

= Вывод номера строки.

? Вывод строки подсказки.

<пробел> Вывод следующей страницы.

<ENTER> Вывод следующей строки.

**find**

*поиск строки (/V вывод не содержащих /c вывод числа найденных строк /N вывод номеров /I без учета регистра*)

Поиск текстовой строки в одном или нескольких файлах.

FIND [/V] [/C] [/N] [/I] [/OFF[LINE]] "строка" [[диск:][путь]имя\_файла[ ...]]

/V Вывод всех строк, НЕ содержащих заданную строку.

/C Вывод только общего числа строк, содержащих заданную строку.

/N Вывод номеров отображаемых строк.

/OFF[LINE] Не пропускать файлы с установленным атрибутом "Автономный".

/I Поиск без учета регистра символов.

"строка" Искомая строка.

[диск:][путь]имя\_файла

Один или несколько файлов, в которых выполняется поиск.

Если путь не задан, команда FIND выполняет поиск в тексте консоли

либо в тексте, переданном по конвейеру другой командой.

**sort**

поиск строки (/V вывод не содержащих /c вывод числа найденных строк /N вывод номеров /I без учета регистра)

**dir**

*вывод списка файлов, подкаталогов указанной папки (также размер, метку тома, дату изменения) (/A - отображение файлов с указанными атрибутами, /b вывод только имен, /o сортировка файлов по параметрам, /Q вывод владельца, /s вывод файлов из подкаталогов, /W вывод в несколько столбцов /X отображение коротких имен файлов, /T - выбор поля времени: создание, последнее изменение, использование)*

Вывод списка файлов и подкаталогов в указанном каталоге.

DIR [диск:][путь][имя файла] [/A[[:]атрибуты]] [/B] [/C] [/D] [/L] [/N]

[/O[[:]порядок сортировки]] [/P] [/Q] [/R] [/S] [/T[[:]время]] [/W] [/X] [/4]

[диск:][путь][имя файла]

Диск, каталог или имена файлов для включения в список.

/A Отображение файлов с указанными атрибутами.

атрибуты D Каталоги. R Файлы, доступные только для чтения.

H Скрытые файлы. A Файлы, готовые для архивирования.

S Системные файлы. I Файлы с неиндексированным содержимым.

L Точки повторной обработки. - Префикс "-" имеет значение НЕ.

/B Вывод только имен файлов.

/C Применение разделителя групп разрядов при выводе размеров файлов.

Используется по умолчанию. Чтобы отключить применение

разделителя групп разрядов, задайте ключ /-C.

/D Вывод списка в нескольких столбцах с сортировкой по столбцам.

/L Использовать нижний регистр для имен файлов.

/N Новый формат длинного списка, имена файлов выводятся в крайнем

правом столбце.

/O Сортировка списка отображаемых файлов.

порядок N По имени (по алфавиту)

сортировки S По размеру (начиная с минимального)

E По расширению (по алфавиту)

D По дате и времени (начиная с самого старого)

G Начать список с каталогов. - Префикс "-" обращает порядок.

/P Пауза после заполнения каждого экрана.

/Q Вывод сведений о владельце файла.

/R Отображение альтернативных потоков данных этого файла.

/S Отображение файлов из указанного каталога и всех его

подкаталогов.

/T Выбор поля времени для сортировки.

время C Создание.

A Последнее использование.

W Последнее изменение.

/W Вывод списка в несколько столбцов.

/X Отображение коротких имен для файлов, чьи имена не соответствуют

стандарту 8.3. Формат аналогичен выводу с ключом /N, но короткие

имена файлов выводятся слева от длинных. Если короткого имени у

файла нет, вместо него выводятся пробелы.

/4 Вывод номера года в четырехзначном формате.

Стандартный набор ключей можно записать в переменную среды DIRCMD. Для отмены

их действия введите в команде те же ключи с префиксом "-", например: /-W.

**cd**

*вывод имени или смена текущего каталога .. – домашний*

Вывод имени либо смена текущего каталога.

CHDIR [/D] [диск:][путь]

CHDIR [..]

CD [/D] [диск:][путь]

CD [..]

.. обозначает переход в родительский каталог.

Команда CD диск: отображает имя текущего каталога указанного диска.

Команда CD без параметров отображает имена текущих диска и каталога.

Параметр /D используется для одновременной смены

текущих диска и каталога.

Изменение команды CHDIR при включении расширенной обработки команд:

Имя текущего каталога в строке вызова преобразуется к тому же регистру

символов, что и для существующих имен на диске. Так, команда CD C:\TEMP

на самом деле сделает текущим каталог C:\Temp, если он существует на диске.

Команда CHDIR перестает рассматривать пробелы как разделители, что позволяет

перейти в подкаталог, имя которого содержит пробелы, не заключая все имя

каталога в кавычки. Например:

cd \winnt\profiles\username\programs\start menu

приводит к тому же результату, что и:

cd "\winnt\profiles\username\programs\start menu"

При отключении расширенной обработки команд используется только второй вариант.

**md**

*создание каталога*

Создание каталога.

MKDIR [диск:]путь

MD [диск:]путь

Изменение команды MKDIR при включении расширенной обработки команд:

Команда MKDIR создает при необходимости все промежуточные каталоги в пути.

Например, если \a не существует, то:

mkdir \a\b\c\d

приводит к тому же результату, что и:

mkdir \a

chdir \a

mkdir b

chdir b

mkdir c

chdir c

mkdir d

При отключении расширенной обработки команд используется только второй вариант.

**rd**

*удаление каталога /s - удаление дерева, /q - без запроса*

Удаление каталога.

RMDIR [/S] [/Q] [диск:]путь

RD [/S] [/Q] [диск:]путь

/S Удаление дерева каталогов, т. е. не только указанного каталога,

но и всех содержащихся в нем файлов и подкаталогов.

/Q Отключение запроса подтверждения при удалении дерева каталогов

с помощью ключа /S.

**del**

*удаление файлов /p - запрос подтверждения /f - принудительно удалять доступные только для чтения /q отключение запроса на удаление /a - отбор по атрибутам*

Удаление одного или нескольких файлов.

DEL [/P] [/F] [/S] [/Q] [/A[[:]атрибуты]] имена

ERASE [/P] [/F] [/S] [/Q] [/A[[:]атрибуты]] имена

имена Список из одного или нескольких файлов или каталогов.

Для удаления группы файлов можно использовать подстановочные

знаки. Если указан каталог, будут удалены все файлы в этом

каталоге.

/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.

атрибуты R Файлы, доступные только для чтения.

S Системные файлы.

H Скрытые файлы.

A Файлы, готовые для архивирования.

I Файлы с неиндексированным содержимым.

L Точки повторной обработки.

- Префикс "-" имеет значение НЕ.

Изменение команд DEL и ERASE при включении расширенной обработки команд:

Результаты вывода для ключа /S принимают обратный характер, то есть выводятся

только имена удаленных файлов, а не файлов, которые не удалось найти.

**ren**

переименование файла

**move**

*перемещение файлов /Y - перезаписывать существующие без предупреждения*

Перемещение файлов и переименование файлов и папок

Перемещение одного или более файлов:

MOVE [/Y | /-Y] [диск:][путь]имя\_файла1[,...] назначение

Переименование папки:

MOVE [/Y | /-Y] [диск:][путь]имя\_папки новое\_имя\_папки

[диск:][путь]имя\_файла1 Определяет местоположение файла или файлов, которые

необходимо переместить.

Назначение Определяет новое местоположение файла. Назначение

может состоять из буквы диска (с последующим

двоеточием), имени папки или их комбинации. При

перемещении только одного файла, можно указать и его

новое имя, если хотите выполнить его одновременное

переименование при перемещении.

[диск:][путь]имя\_папки Определяет папку, которую необходимо переименовать.

новое\_имя\_папки Определяет новое имя папки.

/Y Перезаписывать существующие файлы назначения без

предупреждения.

/-Y Предупреждать при перезаписи существующего файла

назначения.

Ключ /Y может присутствовать в значении переменной среды окружения COPYCMD.

Оно может перекрываться ключом /-Y в командной строке. По умолчанию

используется предупреждение о перезаписи, если только команда MOVE не

выполняется как часть пакетного файла.

**copy**

*копирование файлов /Y - перезаписывать существующие файлы без предупреждения*

Копирование одного или нескольких файлов в другое место.

COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/L] [/A | /B] источник [/A | /B]

[+ источник [/A | /B] [+ ...]] [результат [/A | /B]]

источник Имена одного или нескольких копируемых файлов.

/A Файл является текстовым файлом ASCII.

/B Файл является двоичным файлом.

/D Указывает на возможность создания зашифрованного файла

результат Каталог и/или имя для конечных файлов.

/V Проверка правильности копирования файлов.

/N Использование, если возможно, коротких имен при копировании

файлов, чьи имена не удовлетворяют стандарту 8.3.

/Y Подавление запроса подтверждения на перезапись существующего

конечного файла.

/-Y Обязательный запрос подтверждения на перезапись существующего

конечного файла.

/Z Копирование сетевых файлов с возобновлением.

/L Если источник является символической ссылкой, копирование

ссылки вместо реального файла, на который указывает ссылка.

Ключ /Y можно установить через переменную среды COPYCMD.

Ключ /-Y командной строки переопределяет такую установку.

По умолчанию требуется подтверждение, если только команда COPY

не выполняется в пакетном файле.

Чтобы объединить файлы, укажите один конечный и несколько исходных файлов,

используя подстановочные знаки или формат "файл1+файл2+файл3+...".

**xcopy**

*копирование файлов и структур каталогов /y - подавление запроса существующих файлов /s - копирование только непустых каталогов /l вывод имен копируемых файлов /k - копирование атрибутов /h копирование скрытых и системных*

Копирование файлов и структур каталогов.

XCOPY источник [целевой\_объект] [/A | /M] [/D[:дата]] [/P] [/S [/E]] [/V] [/W]

[/C] [/I] [/Q] [/F] [/L] [/G] [/H] [/R] [/T] [/U]

[/K] [/N] [/O] [/X] [/Y] [/-Y] [/Z] [/B]

[/EXCLUDE:файл1[+файл2][+файл3]...]

источник Копируемые файлы.

целевой\_объект Расположение или имена новых файлов.

/A Копирование только файлов с установленным архивным атрибутом;

сам атрибут при этом не изменяется.

/M Копирование только файлов с установленным архивным атрибутом;

после копирования атрибут снимается.

/D:m-d-y Копирование файлов, измененных не ранее указанной даты.

Если дата не указана, заменяются только конечные файлы,

более старые, чем исходные.

/EXCLUDE:файл1[+файл2][+файл3]...

Список файлов, содержащих строки. Каждая строка должна

располагаться в отдельной строке. Если какая-либо из строк

совпадает с любой частью абсолютного пути к копируемому файлу,

такой файл исключается из операции копирования. Например,

указав строку \obj\ или .obj, можно исключить все файлы из

папки obj или все файлы с расширением OBJ соответственно.

/P Вывод запросов перед созданием каждого нового файла.

/S Копирование только непустых каталогов с подкаталогами.

/E Копирование каталогов с подкаталогами, включая пустые.

Эквивалентен сочетанию ключей /S /E. Совместим с ключом /T.

/V Проверка размера каждого нового файла.

/W Вывод запроса на нажатие клавиши перед копированием.

/C Продолжение копирования вне зависимости от наличия ошибок.

/I Если целевой объект не существует и копируется несколько

файлов, считается, что целевой объект задает каталог.

/Q Запрет вывода имен копируемых файлов.

/F Вывод полных имен исходных и целевых файлов.

/L Вывод имен копируемых файлов.

/G Копирование зашифрованных файлов в целевой каталог, не

поддерживающий шифрование.

/H Копирование, среди прочих, скрытых и системных файлов.

/R Перезапись файлов, предназначенных только для чтения.

/T Создание структуры каталогов без копирования файлов. Пустые

каталоги и подкаталоги не включаются в процесс копирования. Для

создания пустых каталогов и подкаталогов используйте сочетание

ключей /T /E.

/U Копирование только файлов, уже имеющихся в целевом каталоге.

/K Копирование атрибутов. При использовании команды XСOPY обычно

сбрасываются атрибуты "Только для чтения".

/N Использование коротких имен при копировании.

/O Копирование сведений о владельце и данных ACL.

/X Копирование параметров аудита файлов (подразумевает

ключ /O).

/Y Подавление запроса подтверждения на перезапись существующего

целевого файла.

/-Y Запрос подтверждения на перезапись существующего целевого

файла.

/Z Копирование сетевых файлов с возобновлением.

/B Копирование символической ссылки вместо ее целевого объекта.

/J Копирование с использованием небуферизованного ввода/вывода.

Рекомендуется для очень больших файлов.

Ключ /Y можно установить через переменную среды COPYCMD.

Ключ /-Y командной строки переопределяет такую установку.

**attrib**

*работа с атрибутами файлов*

Отображение или изменение атрибутов файлов.

ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [+I | -I]

[диск:][путь][имя\_файла] [/S [/D] [/L]]

+ Установка атрибута.

- Снятие атрибута.

R Атрибут "Только чтение".

A Атрибут "Архивный".

S Атрибут "Системный".

H Атрибут "Скрытый".

I Атрибут "Неиндексированное содержимое".

[диск:][путь][имя файла]

Указание файла или набора файлов для обработки.

/S Обработка файлов с указанными именами в текущем каталоге

и во всех его подкаталогах.

/D Обработка файлов и каталогов.

/L Работать с атрибутами самой символической ссылки,

а не целевого объекта этой символической ссылки.

**fc**

*сравнение файлов (наборов) и вывод различий (не сравнивает размеры, но с /b - показывает 16ричное представление и показывает какой файл длиннее)*

Сравнение двух файлов или двух наборов файлов

и вывод различий между ними.

FC [/A] [/C] [/L] [/LBn] [/N] [/OFF[LINE]] [/T] [/U] [/W]

[/nnnn][диск1:][путь1]имя\_файла1 [диск2:][путь2]имя\_файла2

FC /B [диск1:][путь1]имя\_файла1 [диск2:][путь2]имя\_файла2

/A Вывод только первой и последней строк для каждой группы различий.

/B Сравнение двоичных файлов.

/C Сравнение без учета регистра символов.

/L Сравнение файлов в формате ASCII.

/LBn Максимальное число несоответствий

для заданного числа строк.

/N Вывод номеров строк при сравнении текстовых файлов ASCII.

/OFF[LINE] Не пропускать файлы с установленным атрибутом "Автономный".

/T Символы табуляции не заменяются эквивалентным числом пробелов.

/U Сравнение файлов в формате UNICODE.

/W Пропуск пробелов и символов табуляции при сравнении.

/nnnn Число последовательных совпадающих строк, которое

должно встретиться после группы несовпадающих.

[диск1:][путь1]имя\_файла1

Указывает первый файл или набор файлов для сравнения.

[диск2:][путь2]имя\_файла2

Указывает второй файл или набор файлов для сравнения.

**if**

*условный переход проверка существования, сравнение, проверка ошибки предыдущей команды*

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 Условие является истинным, 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.

Предложение ELSE должно располагаться в той же строке, что и команда, следующая

за ключевым словом IF. Например:

IF EXIST имя\_файла. (

del имя\_файла.

) ELSE (

echo имя\_файла. missing.

)

Следующий пример содержит ОШИБКУ, поскольку команда del должна заканчиваться

переходом на новую строку:

IF EXIST имя\_файла. del имя\_файла. ELSE echo имя\_файла. missing

Следующий пример также содержит ОШИБКУ, поскольку команда ELSE должна

располагаться в той же строке, что и команда, следующая за IF:

IF EXIST имя\_файла. del имя\_файла.

ELSE echo имя\_файла. missing

Вот правильный пример, где все команды расположены в одной строке:

IF EXIST имя\_файла. (del имя\_файла.) ELSE echo имя\_файла. missing

Изменение команды IF при включении расширенной обработки команд:

IF [/I] строка1 оператор\_сравнения строка2 команда

IF CMDEXTVERSION число команда

IF DEFINED переменная команда

где оператор\_сравнения принимает следующие значения:

EQL - равно

NEQ - не равно

LSS - меньше

LEQ - меньше или равно

GTR - больше

GEQ - больше или равно,

а ключ /I, если он указан, задает сравнение текстовых строк без учета

регистра. Ключ /I можно также использовать и в форме строка1==строка2

команды IF. Сравнения проводятся по общему типу данных, так что если

строки 1 и 2 содержат только цифры, то обе строки преобразуются в числа,

после чего выполняется сравнение чисел.

Условие CMDEXTVERSION применяется подобно условию ERRORLEVEL, но значение

сравнивается с внутренним номером версии текущей реализации расширенной

обработки команд. Первая версия имеет номер 1. Номер версии будет

увеличиваться на единицу при каждом добавлении существенных возможностей

расширенной обработки команд. Если расширенная обработка команд отключена,

условие CMDEXTVERSION никогда не бывает истинно.

Условие DEFINED применяется подобно условию EXIST, но принимает в качестве

аргумента имя переменной среды и возвращает истинное значение, если эта

переменная определена.

Строка %ERRORLEVEL% будет развернута в строковое представление текущего

значения кода ошибки ERRORLEVEL, за исключением ситуации, когда уже имеется

переменная среды с именем ERRORLEVEL; в подобном случае подставляется

значение этой переменной. Например, с помощью данной строки можно выполнить

следующее:

goto answer%ERRORLEVEL%

:answer0

echo Получен код возврата 0

:answer1

echo Получен код возврата 1

Допускается и применение описанных выше операторов числового сравнения:

IF %ERRORLEVEL% LEQ 1 goto okay

Строка %CMDCMDLINE% будет развернута в исходную командную строку, переданную

CMD.EXE до любой обработки, за исключением ситуации, когда уже определена

переменная среды с именем CMDCMDLINE; в подобном случае подставляется

значение этой переменной.

Строка %CMDEXTVERSION% будет развернута в строку, представляющую собой

текущее значение CMDEXTVERSION, за исключением ситуации, когда уже имеется

переменная среды с именем CMDEXTVERSION; в подобном случае подставляется

значение этой переменной.

**for**

*работа с набором (файлом, файлами, выводом команды, переменной) /l - цикл /r выполнение для текущего каталога, /f - для резки строки /d - для каталогов*

Выполнение указанной команды для каждого файла набора.

FOR %переменная IN (набор) DO команда [параметры]

%переменная Однобуквенный подставляемый параметр.

(набор) Определяет набор, состоящий из одного или нескольких

файлов. Допускается использование подстановочных

знаков.

команда Команда, которую следует выполнить для каждого файла.

параметры Параметры и ключи для указанной команды.

В пакетных файлах для команды FOR используется запись

%%переменная вместо %переменная. Имена переменных

учитывают регистр букв (%i отличается от %I).

Если включена расширенная обработка команд, поддерживаются

следующие дополнительные формы команды FOR:

FOR /D %переменная IN (набор) DO команда [параметры]

Если набор содержит подстановочные знаки, команда выполняется

для всех подходящих имен каталогов, а не имен файлов.

FOR /R [[диск:]путь] %переменная IN (набор) DO команда [параметры]

Выполнение команды для каталога [диск:]путь, а также для всех

подкаталогов этого пути. Если после ключа /R не указано имя

каталога, используется текущий каталог. Если набор - это

одиночный символ точки (.), команда просто перечисляет дерево каталогов.

FOR /L %переменная IN (начало,шаг,конец) DO команда [параметры]

Набор раскрывается в последовательность чисел с заданными

началом, концом и шагом приращения. Так, набор (1,1,5)

раскрывается в (1 2 3 4 5), а набор (5,-1,1) - в (5 4 3 2 1)

FOR /F ["ключи"] %переменная IN (набор-файлов) DO команда [параметры]

FOR /F ["ключи"] %переменная IN ("строка") DO команда [параметры]

FOR /F ["ключи"] %переменная IN ('команда') DO команда [параметры]

или, если использован параметр usebackq:

FOR /F ["ключи"] %переменная IN (набор-файлов) DO команда [параметры]

FOR /F ["ключи"] %переменная IN ('строка') DO команда [параметры]

FOR /F ["ключи"] %переменная IN (`команда`) DO команда [параметры]

Набор файлов содержит имена одного или нескольких файлов,

которые по очереди открываются, читаются и обрабатываются.

Обработка состоит в чтении файла, разбивке его на отдельные

строки текста и разборе каждой строки в ноль или более подстрок.

Затем вызывается тело цикла "for", при выполнении которого

каждая найденная подстрока используется в качестве значения

переменной. По умолчанию ключ /F выделяет из каждой строки

каждого файла первую отделенную пробелами подстроку. Пустые

строки в файле пропускаются. Необязательный параметр "ключи"

служит для переопределения правил разбора по умолчанию. Он

представляет собой заключенную в кавычки строку, содержащую

одно или несколько ключевых слов для определения параметров

разбора. Ключевые слова:

eol=c - знак начала комментария в конце строки

(допускается задание только одного знака).

skip=n - число пропускаемых при обработке строк в начале

файла.

delims=xxx - набор разделителей вместо используемых по умолчанию

пробела и знака табуляции.

tokens=x,y,m-n - номера подстрок из каждой строки, передаваемые в тело

цикла "for" для каждой итерации. При использовании

этого ключа выделяются дополнительные имена

переменных. Формат m-n представляет собой диапазон

подстрок с номерами от m по n. Если последний знак

в строке tokens= является звездочкой, создается

дополнительная переменная, значением которой будет

весь оставшийся текст в строке после разбора

последней подстроки.

usebackq - применение новой семантики, при которой строка,

заключенная в обратные кавычки, выполняется как

команда, строка, заключенная в прямые одиночные

кавычки, является строкой-литералом, а двойные

кавычки могут использоваться для задания имен

файлов в наборе.

Поясняющий пример:

FOR /F "eol=; tokens=2,3\* delims=, " %i in (мойфайл.txt) do @echo %i %j %k

разбирает файл myfile.txt, пропуская все строки, которые

начинаются с символа точки с запятой, и передает вторую и

третью подстроки из каждой строки в тело цикла "for", причем

подстроки разделяются запятыми и/или пробелами.

В теле цикла переменная %i используется для второй

подстроки, %j - для третьей, а %k получает все оставшиеся

подстроки после третьей. Имена файлов, содержащие пробелы,

необходимо заключать в двойные кавычки. Чтобы использовать

двойные кавычки, необходимо использовать параметр usebackq,

иначе двойные кавычки будут восприняты как определение

строки-литерала для разбора.

Переменная %i явно объявлена в инструкции "for", а переменные

%j и %k объявляются неявно с помощью ключа tokens=.

Ключ tokens= позволяет извлечь из одной строки файла до

26 подстрок, при условии, что это не приведет к попытке

объявить переменную после буквы "z" или "Z". Следует помнить,

что имена переменных FOR являются однобуквенными,

с учетом регистра, глобальными, поэтому одновременно

не может быть активно более 52 переменных.

Логику разбора команды FOR /F можно также использовать

на явно заданной строке, заключив ее в одиночные кавычки и

указав ее в качестве параметра в скобках. Она будет разобрана

как одиночная строка из входного файла.

Наконец, команда FOR /F позволяет разобрать выходные

данные другой команды. Для этого следует использовать

в качестве параметра в скобках строку в обратных одиночных

кавычках. Эта строка передается для выполнения дочернему

обработчику команд CMD.EXE, а вывод этой команды

сохраняется в памяти и разбирается так, как если бы это

был файл. Следующий пример,

FOR /F "usebackq delims==" %i IN (`set`) DO @echo %i,

выведет перечень имен всех переменных среды в текущей

среде.

Операции подстановки ссылок на переменные команды FOR

также расширены. Допускается применение следующих

синтаксических конструкций:

%~I - из переменной %I удаляются обрамляющие кавычки (")

%~fI - переменная %I расширяется до полного имени файла

%~dI - из переменной %I выделяется только имя диска

%~pI - из переменной %I выделяется только путь к файлу

%~nI - из переменной %I выделяется только имя файла

%~xI - из переменной %I выделяется расширение имени файла

%~sI - полученный путь содержит только короткие имена

%~aI - переменная %I расширяется до атрибутов файла

%~tI - переменная %I расширяется до даты /времени файла

%~zI - переменная %I расширяется до размера файла

%~$ПУТЬ:I - проводится поиск по каталогам, заданным в

переменной среды ПУТЬ, и переменная %I

заменяется на полное имя первого найденного

файла. Если переменная ПУТЬ не определена или

в результате поиска не найден ни один файл,

то этот модификатор заменяется на пустую

строку.

При объединении нескольких операторов можно получить

следующие результаты:

~dpI - переменная I раскрывается в имя диска и путь

~nxI - переменная I раскрывается в имя файла и его

расширение

~fsI - переменная I раскрывается в полный путь с

короткими именами

~dp$ПУТЬ:I - проводится поиск по каталогам, заданным в

переменной среды ПУТЬ, и переменная I

раскрывается в имя диска и путь к первому

найденному файлу.

~ftzaI - переменная I раскрывается в строку, подобную

выдаваемой командой DIR

В приведенных выше примерах переменные I и ПУТЬ можно

заменить на другие допустимые значения. Синтаксическая

конструкция с символами ~ заканчивается допустимым именем

переменной цикла FOR.

Для имен переменных рекомендуется использовать заглавные

буквы, например, I, что делает эту конструкцию более удобной

для чтения и предотвращает ошибочное принятие их за

модификаторы, которые не различают регистр.

**set**

*работа с переменными cmd.exe (/a - счетное выражение /p - ввод пользователя)*

Вывод, задание и удаление переменных среды cmd.exe.

SET [переменная=[строка]]

переменная Имя переменной среды.

строка Строка символов, присваиваемая указанной переменной.

Введите SET без параметров для вывода текущих переменных среды.

Изменение команды SET при включении расширенной обработки команд:

Если при вызове указать только имя переменной без знака равенства и значения,

команда SET выведет значения всех переменных, имя которых начинается

с указанной строки. Например:

SET P

Эта команда выведет значения всех переменных, имена которых начинаются с 'P'.

Если имя переменной не найдено в текущей среде, при возврате команда SET

установит значение ошибки ERRORLEVEL 1.

Команда SET не допускает использование знака равенства в имени

переменной среды.

Команда SET поддерживает два дополнительных ключа:

SET /A выражение

SET /P variable=[promptString]

Ключ /A указывает, что строка справа от знака равенства является числовым

выражением, значение которого вычисляется. Обработчик выражений очень

прост и поддерживает следующие операции, перечисленные в порядке убывания

приоритета:

() - группировка

! ~ - - унарные операторы

\* / % - арифметические операторы

+ - - арифметические операторы

<< >> - двоичный сдвиг

& - двоичное И

^ - двоичное исключающее ИЛИ

| - двоичное ИЛИ

= \*= /= %= += -= - присвоение

&= ^= |= <<= >>=

, - разделитель операторов

При использовании любых логических или двоичных операторов необходимо

заключить строку выражения в кавычки. Любые нечисловые строки в выражении

рассматриваются как имена переменных среды, значения которых преобразуются

в числовой вид перед использованием. Если переменная с указанным именем

не определена в системе, вместо нее подставляется нулевое значение. Это

позволяет выполнять арифметические операции со значениями переменных среды,

причем не нужно вводить знаки % для получения значений. Если команда

SET /A вызывается из командной строки, а не из пакетного файла, она выводит

окончательное значение выражения. Слева от любого оператора присваивания

должно стоять имя переменной среды. Числовые значения рассматриваются как

десятичные, если перед ними не стоит префикс 0x для шестнадцатеричных чисел,

и 0 для восьмеричных чисел. Например, числа 0x12,

и 022 обозначают десятичное число 18. Обратите внимание на запись

восьмеричных числе: 08 и 09 не являются допустимыми числами, так как в

восьмеричной системе исчисления цифры 8 и 9 не используются.

Ключ /P позволяет установить значение переменной для входной строки, введенной

пользователем. Показывает указанное приглашение promptString перед чтением

введенной строки. Приглашение promptString может быть пустым.

Подстановка переменной среды может быть расширена следующим образом:

%PATH:str1=str2%

расширит действие переменной среды PATH, заменяя каждое вхождение "str1" в

расширенном результате на "str2". "str2" может быть пустой строкой для

эффективного удаления вхождений "str1" из расширенного вывода. "str1" может

начинаться со звездочки, и в этом случае это будет соответствовать любому

началу расширенного вывода до первого вхождения оставшейся части "str1".

Можно также использовать строки расширения.

%PATH:~10,5%

расширит действие переменной среды PATH, затем использует только 5

символов, которые начинаются с 11-го символа (пропустив 10) расширенного

результата. Если длина не указана, по умолчанию используется оставшееся

значение переменной. Если оба значения (длина и число пропускаемых символов)

отрицательны, следующим используемым значением будет длина значения переменной

среды, добавленной к указанному значению пропуска или указанной длины.

%PATH:~-10%

извлечет последние 10 символов переменной PATH.

%PATH:~0,-2%

извлечет все символы переменной PATH, за исключением 2-х последних.

Наконец, добавлена поддержка связывания времени выполнения для переменных среды

окружения. По умолчанию эта поддержка отключена. Ключ /V командной строки

CMD.EXE позволяет включать и выключать ее. Для вызова справки, наберите CMD /?

Связывание времени выполнения для переменных среды окружения полезно при обходе

ограничений раннего связывания, которое происходит при первом чтении текстовой

строки, а не при ее выполнении. Следующий пример демонстрирует возникающую

проблему при использовании раннего связывания переменных:

set VAR=before

if "%VAR%" == "before" (

set VAR=after

if "%VAR%" == "after" @echo Тело внутреннего оператора сравнения

)

Данное сообщение не будет выводиться, т.к. %VAR% в ОБОИХ выражениях IF

подставляется в момент первого использования в первом IF, в том числе и в тело

первого ветвления IF, которое является составным выражением. В IF внутри

составного выражения в действительности сравниваются значения "before" и

"after", что заведомо ложно. Следующий пример демонстрирует подобную ошибку:

set LIST=

for %i in (\*) do set LIST=%LIST% %i

echo %LIST%

в данном случае список файлов текущей папки никогда не будет построен. Вместо

этого, значением переменной LIST будет имя последнего найденного файла.

И вновь, это случилось потому, что %LIST% подставляется всего один раз -

в момент обработки выражения FOR, когда список еще пуст.

Фактически, приведенный фрагмент эквивалентен следующему примеру:

for %i in (\*) do set LIST= %i

в котором имя последнего найденного файла сохраняется в переменной LIST.

Связывание времени выполнения для переменных среды окружения происходит при

использовании специального символа (восклицательного знака), обозначающего

проведение сопоставления во время выполнения. Если включена поддержка

связывания времени выполнения, то для достижения ожидаемых результатов

приведенные выше фрагменты должны быть изменены следующим образом:

set VAR=before

if "%VAR%" == "before" (

set VAR=after

if "!VAR!" == "after" @echo Тело внутреннего оператора сравнения

)

set LIST=

for %i in (\*) do set LIST=!LIST! %i

echo %LIST%

При включенной расширенной обработке команд доступны несколько переменных

среды, которые не отображаются в списке, отображаемом при вызове команды SET.

Значения этих переменных вычисляются динамически каждый раз при их извлечении.

Если пользователь явно задает переменные с одним из этих имен,

то это определение перекрывает соответствующее динамическое определение,

описанное ниже:

%CD% - раскрывается в строку текущей директории.

%DATE% - раскрывается в текущую дату, используя тот же формат команды DATE.

%TIME% - раскрывается в текущую дату, используя формат команды TIME.

%RANDOM% - раскрывается в случайное десятичное число между 0 и 32767.

%ERRORLEVEL% - раскрывается в текущее значение ERRORLEVEL

%CMDEXTVERSION% - раскрывается в текущее значение версии

расширенной обработки команд.

%CMDCMDLINE% - раскрывается в исходную командную строку, которая вызвала

командный процессор.

**setlocal**

*начало локальных изменений среды*

Начало локализации изменений среды в пакетном файле. Изменения среды,

внесенные после выполнения команды SETLOCAL, 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, выполненных в этом пакетном файле.

SETLOCAL

Изменение команды SETLOCAL при включении расширенной обработки команд:

Команда SETLOCAL принимает необязательные аргументы:

ENABLEEXTENSIONS / DISABLEEXTENSIONS

Это позволяет включить или отключить расширенную обработку команд.

Эти аргументы переопределяют параметры CMD /E:ON или /E:OFF. Для

получения дополнительных сведений введите CMD /?.

ENABLEDELAYEDEXPANSION и DISABLEDELAYEDEXPANSION

включают или отключают отложенное расширение переменных среды. Эти

аргументы переопределяют параметры CMD /V:ON или /V:OFF. Для

получения дополнительных сведений введите CMD /?.

Эти изменения действуют до выполнения соответствующей команды ENDLOCAL,

независимо от их настройки до команды SETLOCAL.

Если команда SETLOCAL вызывается с аргументом, она устанавливает значение

ERRORLEVEL. Если указан один из двух допустимых аргументов, это значение будет

равно нулю, в противном случае возвращается единица. Эту команду можно

использовать в пакетных файлах, чтобы определить доступность расширенной

обработки команд, следующим образом:

VERIFY OTHER 2>nul

SETLOCAL ENABLEEXTENSIONS

IF ERRORLEVEL 1 echo Не удается включить расширенную обработку

Эта последовательность команд работает, так как в прежних версиях CMD.EXE

команда SETLOCAL НЕ устанавливала значение ERRORLEVEL. Команда VERIFY с

недопустимым аргументом необходима для установки ненулевого значения

ERRORLEVEL.

**endlocal**

*конец локальных изменений среды*

Завершение локализации изменений среды в пакетном файле. Изменения среды,

внесенные после выполнения команды ENDLOCAL, не являются локальными

относительно текущего пакетного файла; их прежние значения не будут

восстановлены по завершении этого пакетного файла.

ENDLOCAL

Изменение команды ENDLOCAL при включении расширенной обработки команд:

Если соответствующая команда SETLOCAL включила или отключила расширенную

обработку команд с помощью новых параметров ENABLEEXTENSIONS или

DISABLEEXTENSIONS, то после выполнения команды ENDLOCAL восстанавливается

состояние расширенной обработки команд, каким оно было до выполнения

команды SETLOCAL.

**pushd**

*смена директории (можно сетевую - создаются временные диски)*

Сохраняет текущий каталог для использования командой POPD, а затем

переходит в указанный каталог.

PUSHD [путь | ..]

путь Каталог, который следует назначить текущим.

Когда расширенная обработка команд включена, команда PUSHD допускает ввод

сетевых путей в дополнение к обычным именам дисков и путям.

Если указан сетевой путь, команда PUSHD создает временное имя диска,

указывающее на заданный сетевой ресурс, а затем производит смену текущего

диска и каталога, используя вновь определенное имя диска.

Выделение временных имен дисков проводится в обратном порядке начиная

с Z:, причем выбирается первое свободное имя диска.

**popd**

*удаление временных имен дисков*

Переходит в каталог, сохраненный командой PUSHD.

POPD

Когда расширенная обработка команд включена, команда POPD удаляет временные

имена дисков, созданные командой PUSHD для сетевых ресурсов, при удалении

соответствующего диска из стека каталогов.

**shift**

*сдвиг параметров пакетного файла /n - начало сдвига с номера n*

Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла.

SHIFT [/n]

Команда SHIFT при включении расширенной обработки команд поддерживает ключ

/n, задающий начало сдвига параметров с номера n, где n может быть от 0 до 9.

Например, в следующей команде:

SHIFT /2

%3 заменяется на %2, %4 на %3 и т.д., а %0 и %1 остаются без изменений.

**goto**

*передача управления строке пакфайла в которой есть указанная метка*

Передача управления содержащей метку строке пакетного файла.

GOTO метка

метка Строка пакетного файла, оформленная как метка.

Метка должна находиться в отдельной строке и начинаться с двоеточия.

Изменение команды GOTO при включении расширенной обработки команд:

Команда GOTO принимает в качестве метки перехода строку :EOF, которая вызывает

передачу управления в конец текущего пакетного файла. Это позволяет легко

выйти из пакетного файла без определения каких-либо меток. Команда CALL /?

выводит описание расширенных возможностей команды CALL, делающих эту функцию

особенно полезной.

**call**

*вызов строки с меткой или исполняемого файла*

Вызов одного пакетного файла из другого.

CALL [диск:][путь]имя\_файла [параметры]

параметры Набор параметров командной строки, необходимых

пакетному файлу.

Изменение команды CALL при включении расширенной обработки

команд:

Команда CALL допускает использование меток в качестве

адресата вызова. Применяется следующий синтаксис:

CALL :метка аргументы

При вызове создается новый контекст текущего пакетного файла

с заданными аргументами, и управление передается на инструкцию,

расположенную сразу после метки. Для выхода из такого

пакетного файла необходимо дважды достичь его конца. Первый

выход возвращает управление на инструкцию, расположенную

сразу после строки CALL, а второй выход завершает выполнение

пакетного файла. Команда GOTO /? выводит описание

расширения GOTO :EOF, позволяющее выполнить быстрый

возврат из пакетного файла.

Кроме того, раскрытие ссылок на аргументы пакетных файлов

(%0, %1 и т.д.) изменено следующим образом:

%\* в пакетном файле обозначает все аргументы

(например, %1 %2 %3%4 %5 ...)

Операции подстановки параметров пакетного файла (%n)

также расширены. Допускается применение следующих

синтаксических конструкций:

~I - переменная I раскрывается с удалением

обрамляющих кавычек (")

~fI - переменная I раскрывается в полное имя файла

~dI - переменная I раскрывается в букву диска

~pI - переменная I раскрывается в путь

~nI - переменная I раскрывается в имя файла

~xI - переменная I раскрывается в расширение файла

~sI - раскрытый путь содержит только короткие имена

~aI - переменная I раскрывается в атрибуты файла

~tI - переменная I раскрывается в дату/время файла

~zI - переменная I раскрывается в размер файла

~$ПУТЬ:I - проводится поиск по каталогам, заданным в

переменной среды ПУТЬ, и переменная I

раскрывается в полное имя первого найденного

файла. Если указанная переменная среды

не определена или в результате поиска не найден

ни один файл, то этот модификатор заменяется на

пустую строку.

При объединении модификаторов получаются комбинированные

результаты::

%~dp1 - переменная %1 заменяется только на имя диска и путь

%~nx1 - переменная %1 заменяется только на имя файла и его

расширение

%~dp$ПУТЬ:1 - проводится поиск по каталогам, заданным в переменной

среды ПУТЬ, и переменная %1 заменяется на имя диска

и путь к первому найденному файлу.

%~ftza1 - переменная %1 заменяется на строку, выдаваемую

командой DIR

В приведенных выше примерах переменные %1 и ПУТЬ можно заменить на

любые другие допустимые значения. Синтаксическая конструкция %~

должна завершаться правильным номером аргумента. Модификаторы %~

не могут использоваться вместе с оператором %\*

**exit**

*выход из cmd или пакетного файла (ключ /b, можно указать код\_выхода и errorlevel станет равным ему)*

Завершает программу CMD.EXE (интерпретатор команд) или текущий пакетный

файл-сценарий.

EXIT [/B] [exitCode]

/B Предписывает завершить текущий пакетный файл-сценарий вместо

завершения CMD.EXE. Если выполняется вне пакетного

файла-сценария, то будет завершена программа CMD.EXE

exitCode Указывает цифровое значение. Если указан ключ /B, определяет

номер для ERRORLEVEL. В случае завершения работы CMD.EXE,

устанавливает код завершения процесс с данным номером.

**fdisk**

вывод формата диска

**format**

*форматирование диска /fs:файловая\_система /v:метка\_тома /x - отключение тома, /a:размер\_кластера*

Форматирование диска для работы с Windows.

FORMAT том [/FS:файловая\_система] [/V:метка] [/Q] [/A:размер] [/C] [/X]

[/P:проходы] [/S:состояние]

FORMAT том [/V:метка] [/Q] [/F:размер] [/P:проходы]

FORMAT том [/V:метка] [/Q] [/T:дорожки /N:секторы] [/P:проходы]

FORMAT том [/V:метка] [/Q] [/P:проходы]

FORMAT том [/Q]

том Определяет букву диска (с последующим двоеточием),

точку подключения или имя тома.

/FS:файловая\_система Указывает тип файловой системы (FAT, FAT32, NTFS,

или UDF).

/V:метка Метка тома.

/Q Быстрое форматирование. Перекрывает параметр /P.

/C Только для NTFS: Установка режима сжатия по умолчанию для

всех файлов, создаваемых на новом томе.

/X Инициирует отключение тома, в качестве первого действия, если

это необходимо. Все открытые дескрипторы тома будут неверны.

/R:редакция только для UDF: Форматирование в указанной версии UDF

(1.00, 1.02, 1.50, 2.00, 2.01, 2.50). По умолчанию

используется редакция 2.01.

/D Только UDF 2.50: Метаданные будут продублированы.

/A:размер Заменяет размер кластера по умолчанию. В общих случаях

рекомендуется использовать размеры кластера по умолчанию.

NTFS поддерживает размеры 512, 1024, 2048, 4096, 8192, 16 КБ,

32 КБ, 64K.

FAT поддерживает размеры 512, 1024, 2048, 4096, 8192, 16 КБ,

32 КБ, 64 КБ, (128 КБ, 256 КБ для размера сектора > 512 байт).

FAT32 поддерживает размеры 512, 1024, 2048, 4096, 8192, 16 КБ,

32 КБ, 64 КБ, (128 КБ, 256 КБ для размера сектора > 512 байт).

exFAT поддерживает размеры 512, 1024, 2048, 4096, 8192, 16 КБ,

32 КБ, 64 КБ,

128 КБ, 256 КБ, 512 КБ, 1 МБ, 2 МБ, 4 МБ, 8 МБ, 16 МБ, 32 МБ.

Файловые системы FAT и FAT32 налагают следующие ограничения

на число кластеров тома:

FAT: число кластеров <= 65 526

FAT32: 65 526 < число кластеров < 4 177 918

Выполнение команды FORMAT будет немедленно прервано, если

будет обнаружено нарушение указанных выше ограничений,

используя указанный размер кластеров.

Сжатие NTFS не поддерживается для кластеров размером

более 4096.

/F:размер Указывает размер форматируемых гибких дисков (1,44)

/T:дорожки Число дорожек на каждой стороне диска.

/N:секторы Число секторов на каждой дорожке.

/P:раз Обнуляет каждый сектор тома указанное число раз. Этот параметр

не действует с параметром /Q

/S:состояние Где "состояние" может принимать значения "enable" или

"disable"

Короткие имена по умолчанию включены

**sys**

перенос на диск системных файлов DOS

**label**

*работа с метками тома*

Создание, изменение и удаление меток тома для дисков.

LABEL [диск:][метка]

LABEL [/MP] [том] [метка]

диск: Определяет букву диска.

метка Определяет метку тома.

/MP Определяет точку подключения тома, или

имя тома.

том Определяет букву диска (с последующим двоеточием),

точку подключения или имя тома. Если указано имя тома,

флаг /MP необязателен.

**chkdsk**

*проверка диска /f исправление ошибок на диске /r поиск поврежденных секторов /l задание размера файла журнала (NTFS)*

Проверка диска и вывод отчета.

CHKDSK [том[[путь]имя\_файла]] [/F] [/V] [/R] [/X] [/I] [/C] [/L[:размер]] [/B]

Том Определяет точку подключения, имя тома или букву проверяемого

диска с двоеточием.

имя\_файла Файлы, проверяемые на наличие фрагментации (только FAT/FAT32).

/F Исправление ошибок на диске.

/V Для FAT/FAT32: вывод полного пути и имени каждого файла

на диске.

Для NTFS: вывод сообщений об очистке (при их наличии).

/R Поиск поврежденных секторов и восстановление уцелевшего

содержимого (требует /F).

/L:размер Только для NTFS: задание размера файла журнала (в КБ).

Если размер не указан, выводится текущее значение размера.

/X Предварительное отключение тома (при необходимости).

Все открытые дескрипторы для этого тома будут недействительны

(требует /F).

/I Только для NTFS: менее строгая проверка элементов индекса.

/C Только для NTFS: пропуск проверки циклов внутри структуры

папок.

/B Только для NTFS: повторная оценка поврежденных кластеров на

диске (требует /R)

Параметры /I или /C сокращают время выполнения Chkdsk за счет

пропуска некоторых проверок тома.

**chkntfs**

*выодит или изменяет параметры проверки диска во время загрузки*

Выводит или изменяет параметры проверки диска во время загрузки.

CHKNTFS том: [...]

CHKNTFS /D

CHKNTFS /T[:время]

CHKNTFS /X том: [...]

CHKNTFS /C том: [...]

том Определяет букву диска (с последующим двоеточием),

точку подключения или имя тома.

/D Восстанавливает стандартные параметры компьютера; все диски

проверяются при загрузке и CHKDSK запускается при обнаружении

ошибок.

/T:time Изменяет обратный отсчет AUTOCHK на указанный отрезок

времени в секундах. Если время не указано, показывает текущую

настройку.

/X Исключает стандартную проверку диска при загрузке.

Сведения об исключенных ранее дисках не сохраняются.

/C Запрашивает выполнение проверки диска при следующей загрузке;

если на диске обнаружены ошибки, запускается CHKDSK.

Если никакие параметры не указаны, CHKNTFS отображает или состояние

бита обнаружения ошибок для диска или его проверку при следующей перезагрузке.

**scandisk**

проверка диска

**scandskw**

проверка файловой системы

**cacls**

*работа ACL /g разрешение /d запрет /p замена разрешений (права пользоваеля)*

ПРИМЕЧАНИЕ. CACLS считается устаревшей, рекомендуется использовать ICACLS.

Просмотр и изменение списков управления доступом (ACL) к файлам

CACLS имя\_файла [/T] [/M] [/L] [/S[:SDDL]] [/E] [/C] [/G имя:доступ]

[/R имя [...]] [/P имя:доступ [...]] [/D имя [...]]

имя\_файла Просмотр списков управления доступом.

/T Изменение списков управления доступом для указанных файлов

в текущем каталоге и вложенных каталогах.

/L Работа с самой символической ссылкой, а не с объектом.

/M Изменение ACL томов, подключенных к каталогу.

/S Отображение строки SDDL для DACL.

/S:SDDL Замена ACL на указанные в строке SDDL

(не используется с ключами /E, /G, /R, /P и /D).

/E Изменение списка управления доступом вместо его замены.

/C Продолжение при ошибках отказа в доступе.

/G имя:доступ Разрешения для указанных пользователей.

Доступ: R - чтение

W - запись

C - изменение (запись)

F - полный доступ

/R имя Отзыв разрешений для пользователя (только вместе с /E).

/P имя:доступ Замена разрешений указанного пользователя.

Доступ: N - нет

R - чтение

W - запись

C - изменение (запись)

F - полный доступ

/D имя Запрет на доступ пользователя.

Для выбора нескольких файлов используются подстановочные знаки.

В команде можно указать несколько пользователей.

Сокращения:

CI - наследование контейнерами.

ACE наследуется каталогами.

OI - наследование объектами.

ACE наследуется файлами.

IO - только наследование.

ACE не применяется к текущему файлу или каталогу.

ID - наследование.

ACE наследуется от ACL родительского каталога.

**Команды UNIX (+что делают, что возвращают, основные опции):**

**uname**

печатает данные о системе: ядро (-s) его версия (-v) версия релиза (-r) тип машины (-m) процессора (-p) оборудования (-i) и имя системы (-o)

**date**

время и дата

**time**

время выполнения команды

**ps**

инфа о процессах

**bash**

интерпретатор

**read**

считывание переменных

**echo**

вывод строки на экран

**clear**

очистка экрана терминала

**history**

история вызова команд bash

**cat**

печать сожержимого файла

**more**

постраничный вывод

**less**

постраничный вывод с возможностью прокрутки вверх

**find**

поиск файлов

**wc**

подсчет строк слов и символов

**cut**

резка строки

**grep**

поиск строки

**egrep**

поиск строки по регвыр

**sort**

сортировка строк

**yes**

нерерывно печатает указанную строку или "y"

**ls**

список файлов указанной директории

**cd**

смена директории

**mkdir**

создать папку

**rmdir**

удалить папку

**rm**

удаление файла

**mv**

перемещение файла

**cp**

копирование файла

**chmod**

смена прав пользователя на файл

**chown**

смена владельца файла

**diff**

сравнение файлов

**touch**

меняет дату создания файла или создает файл

**du**

размер директории или файла

**df**

информация о файловой системе

**stat**

информация о файле

**if**

условный переход

**for**

цикл

**while**

"пока"

**case**

оператор выбора

**function**

объявление функции

**set**

работа с переменными

**env**

переменные окружения

**export**

маркировка переменных окружений для последующего вызова

**function()**

тоже объявление функции, здесь function - имя функции

**sleep**

задержка

**exit**

выход из bash

**top**

работающие процы в порядке нагрузки на процессор

**tar**

работа с архивами

**gzip**

архиватор

**hd**

шестнадцатиричный дамп

**[ -a|b|c|... ] (test)**

логические условия

**expr**

вычисление значения выражения

**bc**

калькулятор

**fdisk**

управление разделами

**cfdisk**

улучшенный fdisk

**mkfs**

создание ФС

**fsck**

проверка и устранение ошибок ФС

**mount**

смонтировать файловую систему

**umount**

размонтировать файловую систему

**shutdown**

управление питанием

**poweroff**

выключить

**reboot**

послать в... ребут

**su**

переход в режим суперпользователя (root)

**sudo**

исполнение команды от имени root

**passwd**

изменение пароля текущего пользователя

**who**

список пользователей

**w**

пользователи и процессы

**whoami**

досл. "Кто я?" Очевидно: имя текущего пользователя.

**which**

поиск исполняемых файлов в PATH

**Переменные окружения Windows (+что содержат, как и кем используются):**

**windir**

директория Windows

**TEMP**

временные папки приложений

**ProgramFiles**

папка программ

**CommonProgramFiles**

папка с общими файлами программ

**USERPROFILE**

папка текущего юзера

**OS**

версия ос

**PROCESSOR\_IDENTIFIER**

описание процессора

**ComSpec**

путь к командной оболочке

**PROMPT**

параметры командной строки

**Path**

путь поиска исполняемых файлов

**PATHEXT**

Возвращает список расширений файлов, которые рассматриваются операционной системой как исполняемые.

**INCLUDE**

подключаемые заголовочные файлы для компилера (\*.h)

**LIB**

библиотеки (списки экспортируемых функций из dll)

**CD**

Возвращает путь к текущей папке.

**DATE**

текущая дата дд:мм:гггг

**TIME**

текущее время чч:мм:сс,мс

**RANDOM**

случайное число (от 0 до 32767)

**ERRORLEVEL**

код последней ошибки

**CMDEXTVERSION**

версия расширенной обработки команд

**CMDCMDLINE**

строка, по которой была вызвана командная строка

**DIRCMD**

как отображается вывод у dirwindir - директория Windows

**Переменные окружения Unix (+что содержат, как и кем используются):**

**PS1**

формат командной строки

**PS2**

приглашение на ввод

**PS3(?)**

приглашение команды select

**PS4**

приглашение если команда запущена с ключом -x

**PATH**

путь к исполняемым файлам

**RANDOM**

случайное число 0-32767

**PWD**

текущая директория

**HOME**

домашняя директория текущего пользователя

**LOGNAME**

имя текущего залогиненого пользователя

**USER**

имя пользователя

**UID**

ID пользователя

**PPID**

идентификатор текущего Bash

**HOSTNAME**

имя компа

**HOSTTYPE**

описание машины

**\_**

/usr/bin/env ссылка на команду env

**?**

код возврата последней команды

**Утилиты DOS/Win (+что делают, что возвращают, опции и где взять):**

**regedit**

редактор реестра для 16-разрядных систем (первоначально). Начиная с XP и Server 2003 - единый редактор реестра Windows

**regedt32**

редактор реестра для Windows NT 3.x, NT 4.0, 2000, начиная с XP и Server 2003 - программа для запуска regedit (!)

**reg**

консольный редактор реестра

**runas**

запуск от имени (консольный)

**rundll32**

запуск dll

**regsvr32**

регистрация dll

**diruse**

сканирование дерева каталогов и подсчет занимаемого указанными папками места (Resourse Kit)

**oh**

отображает открытые файлы и процессы, их открывшие (Resourse Kit)

**kill**

убивает задачи (Resourse Kit)

**shutdown**

завершение работы

**windiff**

сравнение файлов и директорий (Support Tools)

**choice**

выбор (y/n) (ms-dos)

**dskprobe**

редактирование файловой системы (Resourse Kit)

**diskedit**

редактирование NTFS (Resourse Kit)

**fsutil**

стандартное управление ФС

**procmon**

монитор процессов (sysinternals)

**diskmon**

монитор действий над дисками (sysinternals)

**autoruns**

управление автозагрузками (sysinternals)

**net**

пользование сетью (консоль)

**mstsc**

Использование подключения к удаленному раблчему столу

MSTSC[<файл подключения>][/v:<сервер[:порт]>][/admin][/f[ullscreen]][/w:<ширина> /h:<высота>][/public]|[/span][/multimon][/migrate][/edit "файл подключения"]

"файл подключения" - имя RDP-файла для подключения.

/v:<сервер[:порт]> - удаленный компьютер, к которому выполняется подключение.

/admin - подключение к сеансу для администрирования сервера.

/f - запуск удаленного рабочего стола.

/w:<ширина> - ширина окна удаленного рабочего стола.

/h:<высота> - высота окна удаленного рабочего стола.

/public - запуск удаленного рабочего стола в общем режиме.

/span - сопоставление ширины и высоты рабочего стола с локальным виртуальным рабочим столом и развертывание на несколько мониторов при необходимости. Для развертывания на несколько мониторов все мониторы должны быть размещены так, чтобы формировать прямоугольник.

/multimon - настраивает структуру размещения мониторов сеанса удаленного рабочего стола в соответствии с текущей конфигурацией на стороне клиента.

/edit - открытие указанного RDP-файла подключения для изменения.

/migrate - миграция файлов подключения прежних версий, созданных с помощью диспетчера подключений клиентов, в новые RDP-файлы подключения.

**Ключи реестра (+кем и как используются):**

**HKLM\System\CurrentControlSet\Control\HiveList**

список ульев, загруженных в оперативную память. Улья (файлы кустов) - файлы, из которых состоит реестр Windows.

**HKLM\System\CurrentControlSet\Services**

информация о сервисах (драйверах). Обычно там работает PnP Manager (ищет запрашиваемый драйвер).

**HKLM\Software\Classes**

хранение информации о типах файлов и объектах COM и ActiveX (по умолчанию).

**HKCR\CLSID**

GUID'ы (global unique identifer) объектов COM, известных системе.

**HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer**

настройки Explorer (по умолчанию).

**\*\NameSpace**

GUID чего-то (настройки Explorer)

**HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer**

пользовательские настройки Explorer.

**RunMRU**

список последних команд "Выполнить" (MRU - most-recently used)

**Map Network Drive MRU**

список последних операций с сетевыми дисками

**RecentDocs**

список последних файлов для "Пуск->Недавние документы"

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

настройки системы

**Explorer**

см. выше.

**Advanced**

дополнительные настройки Explorer (отображение скрытых файлов, запуск в разных процессах etc).

**User Shell Folders**

список папок пользователя (Мои документы, моя музыка etc).

**Run**

автозагружаемые программы.

**RunOnce\Setup**

автозагружаемые после авторизации юзера программы.

**Policies**

политики безопасности.

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

настройки входа в систему

**HKCU(LM)\Software\Policies\Microsoft\Internet Explorer**

настройки политик безопасности IE в интернете.

**HKCU\Console**

настройки консоли

**HKCU\Control Panel**

настройки панели управления

**Структура каталогов в Unix (+их назначение):**

**/etc**

файлы настройки системы

**/etc/init.d**

скрипт при загрузке

**/var**

хранение временных файлов, созданных программами

**/var/log**

хранение log-файлов

**/usr**

хранение прикладных программ

**/usr/local**

в BSD - хранение того, что не входит в дистрибутив, в Linux - хранение собственных программ

**/bin**

хранение основных команд

**/sbin**

хранение основных исполняемых файлов

**/home**

домашние каталоги пользователей

**/dev**

файлы устройств

**/mnt**

монтированные временные файловые системы (флэшка, диск)

**/media**

в некоторых системах монтирование происходит сюда

**/proc**

информация о железе

**/root**

???

**/tmp**

хранение временных файлов

**/sys**

???

**Акселераторы Windows (+что и когда делают):**

**Ctrl+Alt+Del+Del**

окно входа в систему, где можно ввести имя пользователя/вызов диспетчера задач

**Ctrl+Alt+Break**

переключение между оконным и полноэкранным режимом

**Ctrl+Alt+End**

**Ctrl+Shift+Esc**

диспетчер задач

**Ctrl+Esc**

меню Пуск

**Ctrl++**

**Shift+F10**

контекстное меню

**Alt+Tab**

переход между окнами

**Alt+Space**

меню окна

**Alt+Esc**

переход между окнами без появления доп. окошка

**Alt+Enter**

в консольных приложениях полноэкранный режим,

**Alt+PrtScr (Alt+Shift+Tab)**

PrtScr одно окошко

Alt+Shift+Tab - Alt+Tab в обратном порядке

**F2**

переименовать, сохранить в нек. приложениях

**F4**

строка адреса

**Esc**

закрытие меню

**F6**

tab в виндовых окошках

**F7**

вызов списка команд в консоли

**Ctrl+F5**

обновление

**Win+Break**

свойства системы

**Win+E**

мой компьютер

**Win+F**

поиск

**Win+D**

свернуть все окна

**Win+R**

выполнить

**Win+L**

выйти из системы

**Win+M**

свертывание всех окон

**Win+Shift+M**

востановление свернутых окон

**Win+Tab**

переход по окнам на панели задач

**Win+Shift+Tab**

в обратном порядке

**Ctrl+Tab**

переключение между закладками

**Ctrl+Left**

на слово влево

**Ctrl+Right**

на слово вправо

**Ctrl+A**

выделить всё

**Ctrl+C**

копировать

**Ctrl+F**

поиск

**Ctrl+G**

перейти

**Ctrl+H**

заменить

**Ctrl+M**

новая строка

**Ctrl+N**

Создать

**Ctrl+O**

открыть

**Ctrl+P**

печать

**Ctrl+S**

сохранить

**Ctrl+V**

вставить

**Ctrl+X**

вырезать

**Ctrl+Z**

отменить

**Ctrl+Ins**

скопировать

**Shift+Del**

вырезать текст или удалить файл мимо корзины

**Shift+Ins**

вставить

**Сочетания клавиш Unix (+что и когда делают, в режимах EMACS/VI):**

**TAB**

выводит список файлов с данным префиксом(переключение между окнами)

**Ctrl+A**

???

**Ctrl+C**

???

**Ctrl+D**

???

**Ctrl+S**

???

**Ctrl+Q**

???

**Ctrl+L**

???

**Ctrl+Z**

???

**Ctrl+Y**

???

**Ctrl+\**

???

**Ctrl+U**

удаляет ВСЮ строку

**Ctrl+...**

???

**Ctrl+Alt+Del**

завершение работы

**Ctrl+Alt+Fn**

переключение между терминалами

**Alt+Fn**

переключение между терминалами

**Ctrl+Alt+Backspace**

???

**Up**

взять предыдущую команду из истории

**Down**

взять следующую команду из истории

**Ctrl+R**

Поиск по истории команд

**Shift+PgUp**

перемещение по терминалу вверх

**Shift+PgDown**

перемещение по терминалу вниз

**Shift|Alt|Ctrl+ScrollLock**

???

**Alt+Sysrq+H|C|B|O|S|...**

???

**~**

домашняя папка

**Расширения файлов (+что внутри, в каких ОС и как используются):**

**exe**

исполняемый (executable) файл в Windows, MS DOS, OS/2, Symbian. Внутри: код программы, метаданные. Соответствует формату исполняемых файлов (PE). Используется для выполнения программ.

**dll**

динамически подключаемая библиотека в Windows, OS/2. Внутри: код, метаданные. Соответсвует формату PE-файлов. Подключается прямой ссылкой в коде программы, использующей DLL.

**ocx**

элементы управления ActiveX.

**scr**

скринсейвер. Чтоб экран не выгорал.

**cpl**

компонент Панели Управления.

**drv**

старый формат драйверов

**vxd**

старый формат драйверов юзерского типа

**com**

исполняемый файл в MS DOS. Внутри: код программы, метаданные.

**sys**

новый формат драйверов и системных файлов

**bin**

бинарный файл.

**ovl**

часть программы загружаемая, когда нужно.

**bat**

batch file, пакетный файл в MS DOS, Windows. Содержит скрипт для выполнения в командном интерпретаторе

**cmd**

пакетный файл в Windows. аналог bat

**inf**

оформлен как ini, содержит инфу (бывает системным). Можно устанавливать

**reg**

файл реестра (что нужно в реестр положить)

**adm**

файл групповой политики

**pif**

файл с инфой для запуска не-Windows программы

**lnk**

ярлык (ссылка) в Windows. Привязан к файлу (каталогу).

**scf**

командный файл Explorer

**url**

ярлык, привязаный к Интернет-ресурсу (Windows).

**msc**

документ консоли

**js**

скрипт на языке JScript

**vbs**

скрипт на языке VBScript

**wsf**

Windows Scripting File

**wsc**

Windows Scripting Component File

**cab**

архив

**zip**

файл(ы), сжатый(е) по методу компрессии ZIP

**rar**

файл(ы), сжатый(е) по методу компрессии RAR

**iso**

образ CD/DVD дисков по стандарту ISO 9660

**msi**

пакетный файл инсталлятора MS Windows Installer. Содержит в себе сценарий установки и установочные файлы

**pwl**

файл с паролем

**Кодировки символов (+размер символа):**

**cp866**

кодовая страница, где все специфические символы европейских алфавитов заменены на кириллицу. Длина 1 байт. Используется в среде DOS и консоли русифицированных систем Microsoft

**windows-1251**

стандартная кодировка кириллицы в Windows. Наиболее полная кириллическая кодировка из 8-битных. Минус - отсутсвие псевдографики и проблемы с буквой "я" из-за проблем с восьмым битом в некоторых программах

**koi8-r (cp878)**

стандарт кириллической кодировки в 1990. Символы была помещена в таблицу так, чтобы позиции кириллических букв фонетически соответствовали позициям латинских. В результате отказа от алфавитного порядка удалось решить проблему срезаемого восьмого бита - кириллический текст заменялся более-менее читаемым латинским.

**iso-8859-5**

8-битная кодовая страница кирилицы стандарта ISO. Отсутствуют некоторые полезные символы. В основном используется в Сербии и Болгарии.

**mac-cyrillic-2000 (100007)**

кириллица в Mac OS

**utf-8**

формат преобразования Юникода: символы с кодом, меньшим 128, преобразуются в ASCII с тем же кодом; остальные записываются в виде последовательностиот 2 до 6 байт. Плюсы: гарантировано распознавание латиницы, цифр и знаков препинания, выигрыш на определенных текстах по сравнению с UTF-16.

**utf-16**

формат преобразования Юникода в 16-битные слова (один из самых первых). Каждый символ так или иначе будет занимать не меньше 2 байт

**Файловые системы (+размеры файла, тома, число файлов, длина имён):**

**FAT12**

(размер файла и тома 16 Мб (если размер кластера - 16 Кб), число файлов 4077, длина имен 8.3 либо 255)

**FAT16**

(размер файла и тома 2 Гб (если размер кластера - 32 Кб), число файлов 65517, длина имен 8.3 либо 255)

**FAT32**

(размер файла 4 Гб, тома 8 Тб (если размер кластера - 32 Кб), число файлов 2^28-25, длина имен 8.3 байт либо 255)

**CDFS (ISO 9660, Joilet, Rock Ridge)**

**ISO 9660** (размер файла 4 Гб, вложенность каталогов 8, имя 8)

**Joilet** (Расширение для Windows, отмена ограничений на имя файла - 64 символа, и на вложенность папок)

**Rock Ridge** (расширение для UNIX, нет ограничения на каталоги)

**UDF**

(Файловая система, которая позволяет работать с CD, как с жестким диском)

**NTFS4**

(размер файла и тома 2 Тб, имя 255 юникод, поддержка сжатия, если размер кластера до 4 Кб)

**NTFS5**

(то же + шифрование)

**Ext2**

(размер 2 Тб, имя 255 символов)

**Ext3**

(файл 1 Тб, устройство 4 Тб, но журналируема)

**Ext4**

(в разработке. inode 256 байтов, отсюда размер увеличился до 2^60 б, но не поддерживается загрузчиком)

**NFS**

(сетевая файловая система)

**Скорость файловых операций (+сравнение и объяснение):**

**Чтения**

(выше, так как не требуется искать свободное место, необходимо найти первый кластер)

**записи**

(ниже, так как необходимо найти место)

**маленьких файлов**

(большая скорость, так как они занимают мало кластеров)

**больших файлов**

(меньшая скорость, так как они могут быть фрагментированы)

**фрагментированных и не очень**

(фрагментированных меньше)

**первый и второй раз**

(второй раз выше, так как применяется кэширование)

**проверки (scandisk, chkdsk)**

scandisk медленнее, но находит больше дефектов

**полного форматирования**

(около 30 Мб/с)

**поиска по имени**

(В NTFS выше, чем в FAT, так как там двоичный поиск)

**поиска по фрагменту**

(В NTFS выше, так как каждому фрагменту соответствует запись)

**Скорость устройств (+сравнение и объяснение):**

**RAM**

10-20 мегабайт/с

**HDD**

50-120 мегабайт/с

**Flash**

5-20 мегабайт/с

**Сеть**

1-100 мбит

**Специальные файлы Windows (+кем и как используются, где лежат):**

**con**

консоль(псевдоустройство вывода)

**prn**

принтер(псевдоустройство вывода)

**aux**

звук

**nul**

псевдоустройство (никуда не выводит)

**comN**

com - порты

**lptN (N=1..9) +создать :)**

(параллельные порты) (N=1..9) +создать (md .\aux\)

**msdos.sys**

в корневом каталоге загрузочного диска, содержит параметры загрузки windows

**autoexec.bat**

батник при старте системы

**config.sys**

корневой, файл, который содержит команды по загрузке драйверов, проверке памяти и т. д.

**ntldr**

начальный загрузчик операционки

**boot.ini**

инициализация при загрузке(используется ntldr)

**pagefile.sys**

файл подкачки

**ntoskrnl.exe**

ядро виндоуз

**hal.dll**

драйвер основной шины(PCI) одна из частей ядра

**ntdll.dll**

главная системная библиотека пользовательского режима(базовый набор api)

**default**

пользователь систем(кусок реестра)

**SAM**

пароли пользователей(кусок реестра)

**system**

файл реестра(реестра)

**lmhosts**

локальный текстовый файл, содержащий информацию о разрешении имен NetBIOS

**Специальные файлы Unix (+кем и как используются, где лежат):**

**zero**

в /dev, корзина

**null**

в /dev, корзина

**[h|s]d[a|b|...]**

*в /dev, первый жесткий диск IDE*

*в /dev, первый SCSI-диск*

**random**

устройство генерации случайных чисел

**urandom +создать :)**

то же самое, но более высокая степень случайности

**passwd**

список пользователей(информация о пользователей)

**group**

стартовый загрузчик

**shadow**

файл с защищаемой информацией о пользователях

**sudoers**

/etc, пользователи, которым разрешено использовать sudo

**fstab**

информация о файловой системе (информация для монтирования)

**mtab**

информация о уже смонтированных системах

**crontab**

/etc, файл расписаний

**services**

/etc, имена сервисов, номера их портов

**hosts**

текстовый файл, который ассоциирует IP адреса с именами узлов, по одному IP адресу в строке.

**cpuinfo**

/proc, информация о процессоре

**swaps**

месторасположение swap - файла

**version**

версия линуха

**uptime**

насколько долго система запущена

**meminfo**

файл, в котором отражается состояние распределения памяти Linux ядром.

**partitions**

информация о разделах жеского диска

**Службы и системные процессы (+их взаимосвязи и способы запуска):**

**Net Logon**

сетевой вход в систему

**Workstation**

обеспечивает сетевые подключения и связь

**Server**

обеспечивает поддержку RPC и общий доступ к файлам, принтерам и именованным каналам

**Windows Time**

устанавливает время на часах компьютера. W32Time поддерживает синхронизацию даты и времени на всех компьютерах, работающих в сети Microsoft Windows

**Print Spooler**

ставит в очередь задания на печать и управляет ими локально и удаленно

**Messenger**

отправляет и принимает сообщения для пользователей и компьютеров

**Logical Disk Manager**

наблюдает за событиями Plug and Play для распознавания новых дисков и передает информацию о томе и/или диске Службе администрирования диспетчера логических дисков (Logical Disk Manager Administrative Service) для их настройки

**Removable Storage**

управляет съемными носителями и соответствующими библиотеками

**Automatic Updates**

автоматическое обновлнение винды

**Task Scheduler**

Планировщик задач, запуск программ по расписанию, процесс svchost.exe, автоматически

**System Event Notification**

Протоколирует системные события и оповещает о них, svchost.exe, автоматически

**Event Log**

регистрирует сообщения событий, выдаваемые программами и Windows.

**svchost.exe**

название главного процесса для служб, запускаемых из динамически загружаемых библиотек (DLL)

**dllhost.exe**

com+ сервисы (загрузка .NET Runtime)

**internat.exe**

процесс Internat.exe запускается вместе с системой и загружает языковые модули, указанные пользователем

**services.exe**

Это диспетчер управления службами, ответственный за запуск и остановку системных служб и взаимодействие с ними

**spoolsv.exe**

ответственна за управление заданиями на печать и передачу факсимильных сообщений

**winlogon.exe**

процесс управляет входом пользователей в систему и выходом из нее

**lsass.exe**

локальный сервер проверки подлинности, порождающий процесс, ответственный за проверку пользователей в службе Winlogon

**csrss.exe**

подсистема клиент/сервер времени исполнения

**mdm.exe**

дебаг - менеджер

**init**

программа в UNIX и Unix-подобных системах, которая запускает все остальные процессы

**(r)syslogd**

Ведёт запись событий о работе кластера в файл, согласно параметрам, указанным в /etc/syslog.conf

**klogd**

**cron**

программа, выполняющая задания по расписаню, позволяет неоднократный запуск заданий

**Числа (+диапазоны значений):**

**Частота процессора**

Количество процессов (выборка - декодирование - исполнение) в секунду, 900-3800 МГц

**Размер оперативной памяти**

Количество байт в ОП, до 4 Гб при 32-разрядной шине адреса

**Размер жёсткого диска**

Количество байт на жестком диске, до 4000 Гб

**Размер кластера**

Стандартно - 4 кб, от 512 б до 64 Кб

**Количество открытых файлов**

Зависит от ОС, в DOS 255, в win95-98 1024, winXP>16000, в Linux доступная память делить на 128 (inode)

**Количество запущенных процессов и потоков**

Количество процессов определяется числом строк в таблице процессов, максимум 32768, количество потоков то же

**Размер файла подкачки**

Определяется пользователем, по умолчанию 1,5 ОП, 4 Гб

**Размер ядра системы (в памяти и на диске)**

Примерно 3 Мб, в памяти ядро резервирует 2 Гб у каждого процесса

**Размер невыгружаемого страничного пула**

Максимум 4 Гб, но не стоит

**Количество файловых операций в час**

Ограничено объемом журнала транзакций (до 16 кластеров)

**Размер видеопамяти**

64 Кб - 4 Гб

**Количество клавиш на клавиатуре**

64 – 123 (у меня на ноутбуке 83, у компа 121)

**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)

страничка о First Draft by John von Neumann на википедии

**http://awards.acm.org/**

награды Asociation for Computing Machinery (виды наград, награжденные, информация по церемониям)

**http://www.oshistory.net/**

ошибка 500

**http://www.levenez.com/**

некто Эрик Левенец (на сайте выложена история Unix, Windows и языков программирования в виде диаграмм)

**http://www.microsoft.com/technet/sysinternals/**

веб-узел, на котором размещены усовершенствования сервисных программ Windows

**http://cs.usu.edu.ru/docs/pe/**

пособие по PE-файлам (Portable Executable Files, исполняемые файлы для любой архитектуры процессора)

**http://cs.usu.edu.ru/study/presentfs.html**

пособие по файловым системам

**http://web.archive.org/web/20000610180607/czyborra.com/**

некто Роман Чибора (Unicode в Unix)

**http://kernel.org/**

ядро linux (официальный сайт)

**http://freebsd.org/**

freebsd (официальный сайт)

**http://debian.org/**

debian (официальный сайт)

**http://ubuntu.com/**

ubuntu (официальный сайт)

**Почтовые адреса (+ФИО чьи):**

[**andrey.malets@gmail.com**](mailto:andrey.malets@gmail.com)

Андрей Малец

**avkhozov@gmail.com**

Андрей Хозов

**lupusy@inbox.ru**

???

**znick@hackerdom.ru**

Журавлев НН

[**leonid@volkanin.ru**](mailto:leonid@volkanin.ru)

Волканин Леонид Сергеевич