

Командная строка и сценарии Windows

Командная строка позволяет вводить и выполнять команды операционной системы и другие компьютерные команды. Вводя команды, можно выполнять на компьютере различные задачи, не пользуясь мышью или сенсорным вводом.

В разделе перечислены базовые приёмы работы в командной строке Windows.

Для запуска командной строки воспользуйтесь одним из следующих способов:

- «Пуск → Выполнить» (или клавиши Win+R) введите `cmd` и нажмите клавишу `Enter`;
- «Пуск → Все программы → Стандартные → Командная строка»;
- «Пуск → Поиск → Командная строка».

Внутренние и внешние команды. Структура команд

Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором — такие команды называются внутренними (например, `COPY` или `DIR`). Другие команды операционной системы представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и `Cmd.exe`, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, `MORE` или `XCOPY`).

Рассмотрим структуру самой командной строки и принцип работы с ней. Для того, чтобы выполнить команду, вы после приглашения командной строки (например, `C:\>`) вводите имя этой команды (регистр не важен), ее параметры и ключи (если они необходимы) и нажимаете клавишу `<Enter>`. Например:

```
C:\>COPY C:\myfile.txt A:\ /V
```

Имя команды здесь — `COPY`, параметры — `C:\myfile.txt` и `A:\`, а ключом является `/V`. Отметим, что в некоторых командах ключи могут начинаться не с символа `/`, а с символа `-` (минус), например, `-V`.

Многие команды Windows имеют большое количество дополнительных параметров и ключей, запомнить которые зачастую бывает трудно. Большинство команд снабжено встроенной справкой, в которой кратко описываются назначение и синтаксис данной команды. Получить доступ к такой справке можно путем ввода команды с ключом `/?`. Например, если выполнить команду `ATTRIB /?`, то в окне MS-DOS мы увидим следующий текст:

```
Отображение и изменение атрибутов файлов.
ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] [[диск:] [путь] имя_файла] [/S]
+   Установка атрибута.
-   Снятие атрибута.
R   Атрибут "Только чтение".
A   Атрибут "Архивный".
S   Атрибут "Системный".
H   Атрибут "Скрытый".
/S  Обработка файлов во всех вложенных папках указанного пути.
```

Для некоторых команд текст встроенной справки может быть довольно большим и не уместиться на одном экране. В этом случае помощь можно выводить последовательно по одному экрану с помощью команды `MORE` и символа конвейеризации `|`, например:

```
XCOPY /? | MORE
```

В этом случае после заполнения очередного экрана вывод помощи будет прерываться до нажатия любой клавиши. Кроме того, используя символы перенаправления вывода `>` и `>>`, можно текст, выводимый на экран, направить в текстовый файл для дальнейшего просмотра. Например, для

вывода текста справки к команде XCOPY в текстовый файл xcopy.txt, используется следующая команда:

```
XCOPY /? > XCOPY.TXT
```

Замечание

Вместо имени файла можно указывать обозначения устройств компьютера. В Windows поддерживаются следующие имена устройств: PRN (принтер), LPT1–LPT3 (соответствующие параллельные порты), AUX (устройство, присоединяемое к последовательному порту 1), COM1–COM3 (соответствующие последовательные порты), CON (терминал: при вводе это клавиатура, при выводе - монитор), NUL (пустое устройство, все операции ввода/вывода для него игнорируются).

Перенаправление ввода/вывода и конвейеризация (композиция) команд

Рассмотрим более подробно поддерживаемые в Windows UNIX-подобные концепции переназначения устройств стандартного ввода/вывода и конвейерного выполнения команд.

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

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

Из командной строки эти возможности реализуются следующим образом. Для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой, в текстовый файл, нужно использовать конструкцию

```
команда > имя_файла
```

Если при этом заданный для вывода файл уже существовал, то он перезаписывается (старое содержимое теряется), если не существовал — создается. Можно также не создавать файл заново, а дописывать информацию, выводимую командой, в конец существующего файла. Для этого команда перенаправления вывода должна быть задана так:

```
команда >> имя_файла
```

С помощью символа < можно прочитать входные данные для заданной команды не с клавиатуры, а из определенного (заранее подготовленного) файла:

```
команда < имя_файла
```

Приведем несколько примеров перенаправления ввода/вывода.

1. Вывод встроенной справки для команды COPY в файл copy.txt:

```
COPY /? > copy.txt
```

2. Добавление текста справки для команды XCOPY в файл xcopy.txt:

```
XCOPY /? >> copy.txt
```

3. Ввод новой даты из файла date.txt (DATE — это команда для просмотра и изменения системной даты):

```
DATE < date.txt
```

Если при выполнении определенной команды возникает ошибка, то сообщение об этом по умолчанию выводится на экран. В случае необходимости сообщения об ошибках (стандартный поток ошибок) можно перенаправить в текстовый файл с помощью конструкции

```
команда 2> имя_файла
```

В этом случае стандартный вывод будет производиться в файл. Также имеется возможность информационные сообщения и сообщения об ошибках выводить в один и тот же файл. Делается это следующим образом:

```
команда > имя_файла 2>&1
```

Например, в приведенной ниже команде стандартный выходной поток и стандартный поток ошибок перенаправляются в файл copy.txt:

```
XCOPY A:\1.txt C: > copy.txt 2>&1
```

Наконец, с помощью конструкции

```
команда1 | команда2
```

```
ECHO [сообщение]
```

Команды MORE и SORT

Эта команда считывает стандартный ввод из конвейера или перенаправленного файла и выводит информацию частями, размер каждой из которых не больше размера экрана. Используется MORE обычно для просмотра длинных файлов. Возможны три варианта синтаксиса этой команды:

```
MORE [диск:] [путь] имя_файла  
MORE < [диск:] [путь] имя_файла  
имя_команды | MORE
```

Параметр [диск:][путь]имя_файла определяет расположение и имя файла с просматриваемыми на экране данными. Параметр имя_команды задает команду, вывод которой отображается на экране (например, DIR или команда TYPE, используемая для вывода содержимого текстового файла на экран). Приведем два примера.

Для постраничного просмотра помощи команды DIR используется команда:

```
DIR /? | MORE
```

Для постраничного просмотра текстового файла news.txt возможны следующие варианты команд:

```
MORE news.txt  
MORE < news.txt  
TYPE news.txt | MORE
```

Примеры команд для работы с файловой системой

Рассмотрим некоторые наиболее часто используемые команды для работы с файловой системой. Отметим сначала несколько особенностей определения путей к файлам в Windows.

Пути к объектам файловой системы

Напомним, что файловая система логически имеет древовидную структуру и имена файлов задаются в формате [диск:][путь\]имя_файла, то есть обязательным параметром является только имя файла. При этом, если путь начинается с символа "\", то маршрут вычисляется от корневого каталога, иначе — от текущего каталога. Например, имя C:123.txt задает файл 123.txt в текущем каталоге на диске C:, имя C:\123.txt — файл 123.txt в корневом каталоге на диске C:, имя ABC\123.txt — файл 123.txt в подкаталоге ABC текущего каталога.

Существуют особые обозначения для текущего каталога и родительского каталогов. Текущий каталог обозначается символом . (точка), его родительский каталог — символами .. (две точки). Например, если текущим каталогом является C:\WINDOWS, то путь к файлу autoexec.bat в корневом каталоге диска C: может быть записан в виде ..\autoexec.bat.

В именах файлов (но не дисков или каталогов) можно применять так называемые групповые символы или шаблоны: ? (вопросительный знак) и * (звездочка). Символ * в имени файла означает произвольное количество любых допустимых символов, символ ? — один произвольный символ или его отсутствие. Скажем, под шаблон text??1.txt подходят, например, имена text121.txt и text911.txt, под шаблон text*.txt — имена text.txt, textab12.txt, а под шаблон text.* — все файлы с именем text и произвольным расширением.

Для того, чтобы использовать длинные имена файлов при работе с командной строкой, их нужно заключать в двойные кавычки. Например, чтобы запустить файл с именем 'Мое приложение.exe' из каталога 'Мои документы', нужно в командной строке набрать "C:\Мои документы\Мое приложение.exe" и нажать клавишу <Enter>.

Перейдем теперь непосредственно к командам для работы с файловой системой.

Команда CD

Текущий каталог можно изменить с помощью команды

```
CD [диск:] [путь\]
```

Путь к требуемому каталогу указывается с учетом приведенных выше замечаний. Например, команда CD \ выполняет переход в корневой каталог текущего диска. Если запустить команду CD без параметров, то на экран будут выведены имена текущего диска и каталога.

Команда COPY

Одной из наиболее часто повторяющихся задач при работе на компьютере является копирование и перемещение файлов из одного места в другое. Для копирования одного или нескольких файлов используется команда COPY.

Синтаксис этой команды:

```
COPY [/A|/B] источник [/A|/B] [+ источник [/A|/B] [+ ...]]  
[результат [/A|/B]] [/V] [/Y|/N]
```

Краткое описание параметров и ключей команды COPY приведено в [табл. 2.1](#).

Таблица 2.1. Параметры и ключи команды COPY

Параметр	Описание
----------	----------

источник	Имя копируемого файла или файлов
/A	Файл является текстовым файлом ASCII, то есть конец файла обозначается символом с кодом ASCII 26 (<Ctrl>+<Z>)
/B	Файл является двоичным. Этот ключ указывает на то, что интерпретатор команд должен при копировании считывать из источника число байт, заданное размером в каталоге копируемого файла
результат	Каталог для размещения результата копирования и/или имя создаваемого файла
/V	Проверка правильности копирования путем сравнения файлов после копирования
/Y	Отключение режима запроса подтверждения на замену файлов
/-Y	Включение режима запроса подтверждения на замену файлов

Приведем примеры использования команды COPY.

1. Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под тем же именем:

```
COPY abc.txt D:\PROGRAM
```

2. Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под новым именем def.txt:

```
COPY abc.txt D:\PROGRAM\def.txt
```

3. Копирование всех файлов с расширением txt с диска A: в каталог 'Мои документы' на диске C:

```
COPY A:\*.txt "C:\Мои документы"
```

Если не задать в команде целевой файл, то команда COPY создаст копию файла-источника с тем же именем, датой и временем создания, что и исходный файл, и поместит новую копию в текущий каталог на текущем диске. Например, для того, чтобы скопировать все файлы из корневого каталога диска A: в текущий каталог, достаточно выполнить такую краткую команду:

```
COPY A:\*.*
```

В качестве источника или результата при копировании можно указывать имена не только файлов, но и устройств компьютера. Например, для того, чтобы распечатать файл abc.txt на принтере, можно воспользоваться командой копирования этого файла на устройство PRN: COPY abc.txt PRN

Другой интересный пример: создадим новый текстовый файл и запишем в него информацию, без использования текстового редактора. Для этого достаточно ввести команду COPY CON my.txt, которая будет копировать то, что вы набираете на клавиатуре, в файл my.txt (если этот файл существовал, то он перезапишется, иначе — создастся). Для завершения ввода необходимо ввести символ конца файла, то есть нажать клавиши <Ctrl>+<Z>.

Команда COPY может также объединять (склеивать) несколько файлов в один. Для этого необходимо указать единственный результирующий файл и несколько исходных. Это достигается путем использования групповых знаков (? и *) или формата файл1 + файл2 + файл3. Например, для объединения файлов 1.txt и 2.txt в файл 3.txt можно задать следующую команду:

```
COPY 1.txt+2.txt 3.txt
```

Объединение всех файлов с расширением `dat` из текущего каталога в один файл `all.dat` может быть произведено так:

```
COPY /B *.dat all.dat
```

Ключ `/B` здесь используется для предотвращения усечения соединяемых файлов, так как при комбинировании файлов команда `COPY` по умолчанию считает файлы текстовыми.

Если имя целевого файла совпадает с именем одного из копируемых файлов (кроме первого), то исходное содержимое целевого файла теряется. Если имя целевого файла опущено, то в его качестве используется первый файл из списка. Например, команда `COPY 1.txt+2.txt` добавит к содержимому файла `1.txt` содержимое файла `2.txt`. Командой `COPY` можно воспользоваться и для присвоения какому-либо файлу текущей даты и времени без модификации его содержимого. Для этого нужно ввести команду типа

```
COPY /B 1.txt +,,
```

Здесь запятые указывают на пропуск параметра приемника, что и приводит к требуемому результату.

Команда `COPY` имеет и свои недостатки. Например, с ее помощью нельзя копировать скрытые и системные файлы, файлы нулевой длины, файлы из подкаталогов. Кроме того, если при копировании группы файлов `COPY` встретит файл, который в данный момент нельзя скопировать (например, он занят другим приложением), то процесс копирования полностью прервется, и остальные файлы не будут скопированы.

Команда DIR

Еще одной очень полезной командой является `DIR [диск:][путь][имя_файла] [ключи]`, которая используется для вывода информации о содержимом дисков и каталогов. Параметр `[диск:][путь]` задает диск и каталог, содержимое которого нужно вывести на экран. Параметр `[имя_файла]` задает файл или группу файлов, которые нужно включить в список. Например, команда

```
DIR C:\*.bat
```

выведет на экран все файлы с расширением `bat` в корневом каталоге диска `C:`. Если задать эту команду без параметров, то выводится метка диска и его серийный номер, имена (в коротком и длинном вариантах) файлов и подкаталогов, находящихся в текущем каталоге, а также дата и время их последней модификации. После этого выводится число файлов в каталоге, общий объем (в байтах), занимаемый файлами, и объем свободного пространства на диске. Например:

С помощью ключей команды `DIR` можно задать различные режимы расположения, фильтрации и сортировки. Например, при использовании ключа `/W` перечень файлов выводится в широком формате с максимально возможным числом имен файлов или каталогов на каждой строке. Например:

```
Том в устройстве C имеет метку PHYS1_PART2
Серийный номер тома: 366D-6107
Содержимое папки C:\aditor
[.]          [..]          TEMPLT02.DAT  UNINST1.000  HILITE.DAT
TEMPLT01.DAT UNINST0.000  TTABLE.DAT   ADITOR.EXE   README.TXT
ADITOR.NLP   ТЕКСТО~1.TXT
      11 файлов          533 647 байт
      2 папок           143 257 600 байт свободно
```

С помощью ключа `/A[:]атрибуты` можно вывести имена только тех каталогов и файлов, которые имеют заданные атрибуты (`R` — "Только чтение", `A` — "Архивный", `S` — "Системный", `H` — "Скрытый", префикс `"-"` имеет значение `HE`). Если ключ `/A` используется более чем с одним значением атрибута, будут выведены имена только тех файлов, у которых все атрибуты совпадают с

заданными. Например, для вывода имен всех файлов в корневом каталоге диска C:, которые одновременно являются скрытыми и системными, можно задать команду

```
DIR C:\ /A:HS
```

а для вывода всех файлов, кроме скрытых — команду

```
DIR C:\ /A:-H
```

Отметим здесь, что атрибуту каталога соответствует буква D, то есть для того, чтобы, например, вывести список всех каталогов диска C:, нужно задать команду

```
DIR C: /A:D
```

Ключ /O[[:]сортировка] задает порядок сортировки содержимого каталога при выводе его командой DIR. Если этот ключ опущен, DIR печатает имена файлов и каталогов в том порядке, в котором они содержатся в каталоге. Если ключ /O задан, а параметр сортировка не указан, то DIR выводит имена в алфавитном порядке. В параметре сортировка можно использовать следующие значения: N — по имени (алфавитная), S — по размеру (начиная с меньших), E — по расширению (алфавитная), D — по дате (начиная с более старых), A — по дате загрузки (начиная с более старых), G — начать список с каталогов. Префикс "-" означает обратный порядок. Если задается более одного значения порядка сортировки, файлы сортируются по первому критерию, затем по второму и т.д.

Ключ /S означает вывод списка файлов из заданного каталога и его подкаталогов.

Ключ /B перечисляет только названия каталогов и имена файлов (в длинном формате) по одному на строку, включая расширение. При этом выводится только основная информация, без итоговой.

Например:

```
templt02.dat
UNINST1.000
hilite.dat
templt01.dat
UNINST0.000
ttable.dat
aditor.exe
readme.txt
aditor.hlp
Текстовый файл.txt
```

Команды MKDIR и RMDIR

Для создания нового каталога и удаления уже существующего пустого каталога используются команды MKDIR [диск:]путь и RMDIR [диск:]путь [ключи] соответственно (или их короткие аналоги MD и RD). Например:

```
MKDIR "C:\Примеры"
RMDIR "C:\Примеры"
```

Команда MKDIR не может быть выполнена, если каталог или файл с заданным именем уже существует. Команда RMDIR не будет выполнена, если удаляемый каталог не пустой.

Команда DEL

Удалить один или несколько файлов можно с помощью команды

```
DEL [диск:] [путь]имя_файла [ключи]
```

Для удаления сразу нескольких файлов используются групповые знаки ? и *. Ключ /S позволяет удалить указанные файлы из всех подкаталогов, ключ /F – принудительно удалить файлы, доступные только для чтения, ключ /A[:]атрибуты – отбирать файлы для удаления по атрибутам (аналогично ключу /A[:]атрибуты] в команде DIR).

Команда REN

Переименовать файлы и каталоги можно с помощью команды RENAME (REN). Синтаксис этой команды имеет следующий вид:

```
REN [диск:] [путь] [каталог1|файл1] [каталог2|файл2]
```

Здесь параметр каталог1|файл1 определяет название каталога/файла, которое нужно изменить, а каталог2|файл2 задает новое название каталога/файла. В любом параметре команды REN можно использовать групповые символы ? и *. При этом представленные шаблонами символы в параметре файл2 будут идентичны соответствующим символам в параметре файл1. Например, чтобы изменить у всех файлов с расширением txt в текущей директории расширение на doc, нужно ввести такую команду:

```
REN *.txt *.doc
```

Если файл с именем файл2 уже существует, то команда REN прекратит выполнение, и произойдет вывод сообщения, что файл уже существует или занят. Кроме того, в команде REN нельзя указать другой диск или каталог для создания результирующих каталога и файла. Для этой цели нужно использовать команду MOVE, предназначенную для переименования и перемещения файлов и каталогов.

Команда MOVE

Синтаксис команды для перемещения одного или более файлов имеет вид:

```
MOVE [/Y|/Y] [диск:] [путь] имя_файла1 [, ...] результирующий_файл
```

Синтаксис команды для переименования папки имеет вид:

```
MOVE [/Y|/Y] [диск:] [путь] каталог1 каталог2
```

Здесь параметр результирующий_файл задает новое размещение файла и может включать имя диска, двоеточие, имя каталога, либо их сочетание. Если перемещается только один файл, допускается указать его новое имя. Это позволяет сразу переместить и переименовать файл. Например,

```
MOVE "C:\Мои документы\список.txt" D:\list.txt
```

Если указан ключ /-Y, то при создании каталогов и замене файлов будет выдаваться запрос на подтверждение. Ключ /Y отменяет выдачу такого запроса.