

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и
управления»



Сёмкин П.С., Сёмкин А.П.

Методические указания по выполнению лабораторных работ
по дисциплине

«Операционные системы»

Лабораторная работа № 2

«ОС Windows. Командная строка CMD. Командные файлы.»

Москва

2017 г.

Оглавление

1	Цель работы.	2
2	Теоретическая часть.	2
2.1	Команды ОС Windows.....	2
2.1.1	Внутренние и внешние команды. Структура команд	2
2.1.2	Перенаправление ввода/вывода и конвейеризация (композиция) команд	4
2.1.3	Команды MORE и SORT	6
2.1.4	Условное выполнение и группировка команд.....	8
2.2	Команды для работы с файловой системой	10
2.2.1	Команда CD.....	10
2.2.2	Команда COPY.....	11
2.2.3	Команда XCOPY.....	13
2.2.4	Команда DIR	16
2.2.5	Команды MKDIR и RMDIR.....	17
2.2.6	Команда DEL	18
2.2.7	Команда REN	18
2.2.8	Команда MOVE	18
2.3	Командные файлы.....	19
2.3.1	Назначение	19
2.3.2	Вывод сообщений и дублирование команд	20
2.3.3	Использование параметров командной строки	20
3	Задание на выполнение работы	22
4	Контрольные вопросы	24
5	ЛИТЕРАТУРА	24

1 Цель работы.

Целью работы является изучение интерфейса командной строки операционной системы Windows 7 и работа с командами и командными файлами.

Продолжительность работы – 2 часа

2 Теоретическая часть.

2.1 Команды ОС Windows

2.1.1 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, такие команды называются внешними (например, MORE или XCOPY).

Для выполнения команды после приглашения командной строки надо ввести имя этой команды (регистр не важен), ее параметры и ключи (если они необходимы) и нажать клавишу <Enter>.

Например:

C:\>COPY C:\myfile.txt D:\ /V

здесь **COPY** - имя команды -, **C:\myfile.txt** и **D** – параметры, **/V** ключ. 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 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 **/?**.

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 **xcory.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).

2.1.2 Перенаправление ввода/вывода и конвейеризация (композиция) команд

Windows поддерживает UNIX-концепции переназначения устройств стандартного ввода/вывода и конвейерного выполнения команд. помощью переназначения 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. 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).

В командной строке для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой, в текстовый файл, необходимо 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

Примеры перенаправления ввода/вывода.

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

COPY /? > copy.txt

Добавление текста справки для команды XCOPY в файл copy.txt:

XCOPY /? >> copy.txt

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

DATE < date.txt

Если при выполнении 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

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 C:\1.txt F: > copy.txt 2>&1

С помощью конструкции

команда1 | команда2

можно использовать сообщения, выводимые первой командой, в качестве входных данных для второй команды (конвейер команд).

Используя механизмы перенаправления ввода/вывода и конвейеризации, можно из командной строки посылать информацию на различные устройства и автоматизировать ответы на запросы, выдаваемые командами или программами, использующими стандартный ввод.

Для решения таких задач предназначена команда **ECHO [сообщение]**, которая выводит сообщение на экран.

Примеры использования этой команды.

Посылка символа прогона на принтер:

ECHO ^L > PRN

Соединение по телефону из командной строки (модем связан с портом COM2):

ECHO ATDT 1(123)555-1234 > COM2

2.1.3 Команды **MORE** и **SORT**

Одной из наиболее часто использующихся команд, применяющей перенаправление ввода/вывода и конвейеризацию, является команда **MORE**. 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).

Примеры.

Для поэкранного просмотра помощи команды **DIR** используется команда:

DIR /? | MORE

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

MORE news.txt

MORE < news.txt

TYPE news.txt | MORE

Другой распространенной командой, использующей перенаправление ввода/вывода и конвейеризацию, является команда **SORT**. Эта команда работает как фильтр: она считывает символы в заданном столбце, упорядочивает их в возрастающем или убывающем порядке и выводит отсортированную информацию в файл, на экран или другое устройство. Возможны два варианта синтаксиса этой команды:

SORT [/R] [/+n] [[диск1:][путь1]файл1] [> [диск2:][путь2]файл2]

или

[команда] | SORT [/R] [/+n] [> [диск2:][путь2]файл2]

В первом случае параметр **[диск1:][путь1]файл1** определяет имя файла, который нужно отсортировать. Во втором случае будут отсортированы выходные данные указанной команды. Если параметры **файл1** или команда не заданы, то **SORT** будет считывать данные с устройства стандартного ввода.

Параметр **[диск2:][путь2]файл2** задает файл, в который будет направляться сортированный вывод; если этот параметр не задан, то вывод будет направлен на устройство стандартного вывода.

По умолчанию сортировка выполняется в порядке возрастания. Ключ **/R** позволяет изменить порядок сортировки на обратный (от Z к A и затем от 9 до 0). Например, для поэкранного просмотра отсортированного в обратном порядке файла **price.txt**, нужно задать следующую команду:

`SORT /R < price.txt |MORE`

Ключ `/+n` задает сортировку в файле по символам n-го столбца. Например, `/+10` означает, что сортировка должна осуществляться, начиная с 10-й позиции в каждой строке. По умолчанию файл сортируется по первому столбцу.

2.1.4 Условное выполнение и группировка команд

В командной строке Windows можно использовать специальные символы, которые позволяют вводить несколько команд одновременно и управлять работой команд в зависимости от результатов их выполнения. С помощью таких символов условной обработки можно содержание небольшого пакетного файла записать в одной строке и выполнить полученную составную команду.

Используя символ амперсанда **&**, можно разделить несколько утилит в одной командной строке, при этом они будут выполняться друг за другом. Например, если набрать команду **`DIR & PAUSE & COPY /?`** и нажать клавишу `<Enter>`, то вначале на экран будет выведено содержимое текущего каталога, а после нажатия любой клавиши – выведена встроенная справка команды **`COPY`**.

Символ **^** позволяет использовать командные символы как текст, то есть при этом происходит игнорирование значения специальных символов.

Например, если ввести в командной строке

`ECHO Абв & COPY /?`

и нажать клавишу `<Enter>`, то произойдет выполнение подряд двух команд:

`ECHO Абв` и **`COPY /?`** (команда **`ECHO`** выводит на экран символы, указанные в командной строке после нее).

После выполнения команды

`ECHO Абв ^& COPY /?`

на экран будет выведено **`Абв & COPY /?`**

В этом случае просто выполняется одна команда ECHO с соответствующими параметрами.

Условная обработка команд в Windows осуществляется с помощью символов **&&** и **||** следующим образом. Двойной амперсанд **&&** запускает команду, стоящую за ним в командной строке, только в том случае, если команда, стоящая перед амперсандами была выполнена успешно. Например, если в корневом каталоге диска C: есть файл **plan.txt**, то выполнение команды **TYPE C:\plan.txt && DIR** приведет к выводу на экран этого файла и содержимого текущего каталога. Если же файл C:\plan.txt не существует, то команда DIR выполняться не будет.

Два символа **||** осуществляют в командной строке обратное действие, т.е. запускают команду, стоящую за этими символами, только в том случае, если команда, идущая перед ними, не была успешно выполнена. Таким образом, если в предыдущем примере файл C:\plan.txt будет отсутствовать, то в результате выполнения строки **TYPE C:\plan.txt || DIR** на экран выведется содержимое текущего каталога.

Условная обработка действует только на ближайшую команду, то есть в команде

TYPE C:\plan.txt && DIR & COPY /? команда **COPY /?** запустится в любом случае, независимо от результата выполнения команды **TYPE C:\plan.txt**.

Несколько команд можно сгруппировать в командной строке с помощью скобок.

Например, есть две команды:

TYPE C:\plan.txt && DIR & COPY /?

TYPE C:\plan.txt && (DIR & COPY /?)

В первой из них символ условной обработки **&&** действует только на команду **DIR**, во второй - одновременно на две команды: **DIR** и **COPY**.

2.2 Команды для работы с файловой системой

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 каталога. Текущий каталог обозначается символом **.** (точка), его родительский каталог - символами **..** (две точки). Например, если текущим каталогом является C:\WINDOWS, то путь к файлу **explorer.exe** в корневом каталоге диска C: может быть записан в виде **..\explorer.exe**.

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>.

2.2.1 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.

2.2.2 Команда COPY

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

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

Таблица 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

Примеры использования команды COPY.

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

COPY abc.txt D:\PROGRAM

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

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

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 C: в текущий каталог, достаточно выполнить команду:

COPY C:*.*

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**

Пример: создания нового текстовый файла и записи в него информации 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.- 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 с помощью команды: **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 **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.

2.2.3 Команда XCOPY

Указанные при описании команды `COPY` проблемы можно решить с помощью команды `XCOPY`, которая предоставляет намного больше возможностей при копировании. Необходимо отметить, что `XCOPY` может работать только с файлами и каталогами, но не с устройствами.

Синтаксис команды:

XCOPY источник [результат] [ключи]

Команда XCOPY имеет множество ключей, ниже дано описание некоторых из них.

Ключ /D[:[дата]] позволяет копировать только файлы, измененные не ранее указанной даты. Если параметр дата не указан, то копирование будет производиться только если источник новее результата.

Например, команда

XCOPY "C:\Мои документы*.*" "D:\BACKUP\Мои документы" /D

скопирует в каталог 'D:\BACKUP\Мои документы' только те файлы из каталога 'C:\Мои документы', которые были изменены со времени последнего подобного копирования или которых вообще не было в 'D:\BACKUP\Мои документы'.

Ключ /S позволяет копировать все непустые подкаталоги в каталоге-источнике.

С помощью ключа /E можно копировать все подкаталоги, включая и пустые.

Если указан ключ /C, то копирование будет продолжаться даже в случае возникновения ошибок. Это бывает очень полезным при операциях копирования, производимых над группами файлов, например, при резервном копировании данных.

Ключ /I важен для случая, когда копируются несколько файлов, а файл назначения отсутствует. При задании этого ключа команда XCOPY считает, что файл назначения должен быть каталогом. Например, если задать ключ /I в команде копирования всех файлов с расширением txt из текущего каталога в несуществующий еще подкаталог TEXT,

XCOPY *.txt TEXT /I

то подкаталог TEXT будет создан без дополнительных запросов.

Ключи /Q, /F и /L отвечают за режим отображения при копировании. При задании ключа /Q имена файлов при копировании не отображаются, ключа /F - отображаются полные пути источника и результата. Ключ /L

обозначает, что отображаются только файлы, которые должны быть скопированы (при этом само копирование не производится).

С помощью ключа **/H** можно копировать скрытые и системные файлы, а с помощью ключа **/R** — заменять файлы с атрибутом "Только для чтения".

Например, для копирования всех файлов из корневого каталога диска C: (включая системные и скрытые) в каталог SYS на диске D:, нужно ввести команду:

```
XCOPY C:\*.* D:\SYS /H
```

Ключ **/T** позволяет применять XCOPY для копирования только структуры каталогов источника, без дублирования находящихся в этих каталогах файлов, причем пустые каталоги и подкаталоги не включаются. Для того, чтобы все же включить пустые каталоги и подкаталоги, нужно использовать комбинацию ключей **/T /E**.

Используя XCOPY можно при копировании обновлять только уже существующие файлы (новые файлы при этом не записываются). Для этого применяется ключ **/U**. Например, если в каталоге C:\2 находились файлы a.txt и b.txt, а в каталоге C:\1 — файлы a.txt, b.txt, c.txt и d.txt, то после выполнения команды

```
XCOPY C:\1 C:\2 /U
```

в каталоге C:\2 по-прежнему останутся лишь два файла a.txt и b.txt, содержимое которых будет заменено содержимым соответствующих файлов из каталога C:\1.

Если с помощью XCOPY копировался файл с атрибутом "Только для чтения", то по умолчанию у файла-копии этот атрибут снимется. Для того, чтобы копировать не только данные, но и полностью атрибуты файла, необходимо использовать ключ **/K**.

Ключи **/Y** и **/-Y** определяют, нужно ли запрашивать подтверждение перед заменой файлов при копировании. **/Y** означает, что такой запрос нужен, **/-Y** - не нужен.

2.2.4 Команда DIR

Команда используется для вывода информации о содержимом дисков и каталогов.

DIR [диск:][путь][имя_файла] [ключи],

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.

Команда

DIR C:*.bat

выведет на экран список файлов 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.

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.

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 НЕ). 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

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.

2.2.5 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**).

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.

2.2.6 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).

2.2.7 Команда REN

Переименовать файлы и каталоги можно с помощью команды **RENAME (REN)**.

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 каталога 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.

2.2.8 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.

2.3 Командные файлы

2.3.1 Назначение

Командных файл содержат в себе последовательность команд Windows. Командные файлы в ряде случаев представляют удобный и простой способ работы с файлами, папками и процессами в операционной системе Windows. Часто такие файлы используются для автоматизации рутинных процессов, например, для очистки временных папок, для создания резервных копий и тому подобного.

Кроме команд командные файлы содержат также некоторые дополнительные инструкции и ключевые слова, придающие командным файлам некоторое сходство с алгоритмическими языками программирования.

Например, если записать в файл **deltmp.bat** следующие команды:

C:

CD %TEMP%

DEL /F *.tmp

и запустить его на выполнение (аналогично исполняемым файлам с расширением com или exe), то мы удалим все файлы во временной папке каталога Windows. Таким образом, исполнение командного файла приводит к тому же результату, что и последовательный ввод записанных в нем команд. При этом не проводится никакой предварительной компиляции или проверки

синтаксиса кода; если встречается строка с ошибочной командой, то она игнорируется. Очевидно, что если приходится часто выполнять одни и те же действия, то использование командных файлов может сэкономить много времени.

2.3.2 Вывод сообщений и дублирование команд

По умолчанию команды пакетного файла перед исполнением выводятся на экран. С помощью команды **ECHO OFF** можно отключить дублирование команд, идущих после нее (сама команда **ECHO OFF** при этом все же дублируется).

Для восстановления режима дублирования используется команда **ECHO ON**. Кроме этого, можно отключить дублирование любой отдельной строки в командном файле, написав в начале этой строки символ **@**, например: **@DIR D:**

В пакетном файле можно выводить на экран строки с сообщениями. Делается это с помощью команды

ECHO сообщение

Команда **ECHO**. (точка должна следовать непосредственно за словом "ECHO") выводит на экран пустую строку.

Например,

Часто бывает удобно для просмотра сообщений, выводимых из пакетного файла, предварительно полностью очистить экран командой **CLS**.

Используя механизм перенаправления ввода/вывода (символы **>** и **>>**), можно направить сообщения, выводимые командой **ECHO**, в определенный текстовый файл.

2.3.3 Использование параметров командной строки

При запуске пакетных файлов в командной строке можно указывать произвольное число параметров, значения которых можно использовать

внутри файла. Это позволяет, например, применять один и тот же командный файл для выполнения команд с различными параметрами.

Для доступа из командного файла к параметрам командной строки применяются символы %0, %1, ..., %9 или %*. При этом вместо %0 подставляется имя выполняемого пакетного файла, вместо %1, %2, ..., %9 - значения первых девяти параметров командной строки соответственно, а вместо %* - все аргументы. Если в командной строке при вызове пакетного файла задано меньше девяти параметров, то "лишние" переменные из %1 – %9 замещаются пустыми строками.

При необходимости можно использовать более девяти параметров командной строки. Это достигается с помощью команды SHIFT, которая изменяет значения замещаемых параметров с %0 по %9, копируя каждый параметр в предыдущий, то есть значение %1 копируется в %0, значение %2 – в %1 и т.д. Замещаемому параметру %9 присваивается значение параметра, следующего в командной строке за старым значением %9. Если же такой параметр не задан, то новое значение %9 — пустая строка.

При включении расширенной обработки команд SHIFT поддерживает ключ /n, задающий начало сдвига параметров с номера n, где n может быть числом от 0 до 9.

Команда, обратная SHIFT (обратный сдвиг), отсутствует. После выполнения SHIFT уже нельзя восстановить параметр (%0), который был первым перед сдвигом. Если в командной строке задано больше десяти параметров, то команду SHIFT можно использовать несколько раз.

В командных файлах имеются некоторые возможности синтаксического анализа заменяемых параметров. Для параметра с номером n (%n) допустимы синтаксические конструкции (операторы), представленные в табл. 2.

Таблица 2. Операторы для заменяемых параметров

Операторы	Описание
%~Fn	Переменная %n расширяется до полного имени файла

%~Dn	Из переменной %n выделяется только имя диска
%~Pn	Из переменной %n выделяется только путь к файлу
%~Nn	Из переменной %n выделяется только имя файла
%~Xn	Из переменной %n выделяется расширение имени файла
%~Sn	Значение операторов N и X для переменной %n изменяется так, что они работают с кратким именем файла
%~\$PATH:n	Проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %n заменяется на полное имя первого найденного файла. Если переменная PATH не определена или в результате поиска не найден ни один файл, эта конструкция заменяется на пустую строку. Естественно, здесь переменную PATH можно заменить на любое другое допустимое значение

Данные синтаксические конструкции можно объединять друг с другом, например:

%~DPn — из переменной %n выделяется имя диска и путь,

%~NXn — из переменной %n выделяется имя файла и расширение.

3 Задание на выполнение работы

1. Войти в систему под учётной записью **StudXX**, где **XX** - индекс группы.

Запустить виртуальную машину **Oracle VM VirtualBox**. Запустить гостевую операционную систему **Windows 7**.

2. Запустить командный интерпретатор **cmd**

Интерфейс командной строки в операционных системах Windows реализован с помощью командного интерпретатора cmd.exe. Интерпретатор позволяет последовательно вводить команды в интерактивном режиме и отображать результаты их выполнения.

- Закрепить окно командной строки на панели задач

Пуск - Все программы ⇒ Стандартные ⇒ Командная строка

Щёлкнуть правой клавишей мыши на команде – выбрать «**Закрепить на панели задач**»

- **Запустить командную строку с правами администратора**
 - Щёлкнуть правой клавишей мыши на значке командной строки в панели задач
 - Щёлкнуть правой клавишей мыши на пункте «Командная строка»
 - Выбрать пункт «Запуск от имени администратора»
- 3. Сделать текущим корневым каталог диска **C:**
- 4. Просмотреть содержимое корневого каталога, и, если в нём нет каталога **LabOS**, то создать его
- 5. Из корневого каталога диска **C:** перейти в каталог **LabOS**
- 6. В каталоге **LabOS** создать каталог **ИУ5-XX**(заменить **XX** на индекс группы)
- 7. В каталоге **ИУ5-XX** создать каталог **Work1** и подчиненный ему каталог **DOC1**.
- 8. В каталоге **ИУ5-XX** создать каталог **Work2** и подчиненный ему каталог **DOC2**
- 9. В каталоге **DOC1** создать файлы:
 - **группа.txt**, содержащий индекс группы,
 - **дисциплина.txt**, содержащий текст «**операционные системы**»
 - **студент.txt**, содержащий фамилию, имя, отчество студента
- 10. В каталоге **Work1** создать командный файл **run.bat**, , выполняющий следующие операции:
 - просмотр оглавления каталога **DOC1**
 - просмотр содержимого файлов **группа.txt**, **дисциплина.txt**, **студент.txt**,
 - создание каталога **Tools**
- 11. Запустить данный командный файл на исполнение.
- 12. Перенести файл **run.bat** в каталог **Tools**
- 13. Склеить файлы **группа.txt**, **дисциплина.txt**, **студент.txt**, в файл **анкета.txt**

14. Поместить файл **анкета.txt**. в каталог **DOC2**

15. Написать командный файл **Просмотр.bat**, который выводит на экран содержимое файла. Имя выводимого файла должно задаваться в параметре командного файла. Перед выводом содержимого файла необходимо напечатать заголовок, содержащий **имя выводимого файла, текущую дату и время.**

Примечание: Команды для выдачи даты и времени **date /T** и **time /T**

16. С помощью командного файла **Просмотр.bat** вывести на экран содержимое файла **анкета.txt**

4 Контрольные вопросы

1. В чём различие внутренних и внешних команд Windows
2. Как вызываются команды в командной строке?
3. Как можно получить справку по команде?
4. В чём суть конвейерного выполнения команд?
5. Что такое командный файл?
6. Как используются параметры командных файлов?
7. Объясните назначение переменных среды и правила их использования.
8. Как можно приостановить выполнение командного файла?

5 ЛИТЕРАТУРА

1. . В. Гордеев, Операционные системы: Учебник для вузов. 2-е издание
СПб.: «Питер», 2007. – 416 с.: ил.
2. Книттель Б., Windows 7. Скрипты, автоматизация и командная строка. –
СПб.: Питер, 2012 – 784 с. : ил.