

# Лабораторная работа № 1

## Командная оболочка операционной системы WINDOWS.

[Введение](#)

[Общие сведения о командной оболочке](#)

[Использование синтаксиса команд](#)

[Использование нескольких команд и символов условной обработки](#)

[Вложенные командные оболочки](#)

[Использование переменных среды в Cmd.exe](#)

[Установка переменных среды](#)

[Команды ОС MS DOS](#)

[Пакетные \(процедурные\) файлы](#)

[Команды пакетной обработки](#)

[Описание основных команд пакетной обработки.](#)

[Задание лабораторной работы](#)

[Контрольные вопросы.](#)

[Требование к оформлению отчета.](#)

[Пример процедурного файла](#)

**Цель работы** - изучить основные команды командной оболочки операционной системы WINDOWS (в дальнейшем ОС). Освоить программирование и создание простых пакетных (процедурных) файлов.

Работа рассчитана на 4 ак. часа ( две пары).

### Введение

Ниже дается основные сведения о командной оболочке для системы WINDOWS XP. С целью совместимости с другими (ранними) версиями ОС и с ОС MS DOS в практическую часть лабораторной работы включаются только совместимые команды. Разработка пакетных (процедурных) файлов предполагается на базе простых команд оболочки (работающих в любых ОС и ОС MS DOS). В дальнейшем такое подмножество команд оболочки будет называться "[команды MS DOS](#)".

### Общие сведения о командной оболочке

Командная оболочка — это отдельный программный продукт, который обеспечивает прямую связь между пользователем и операционной системой. Текстовый пользовательский интерфейс командной строки предоставляет среду, в которой выполняются приложения и служебные программы с текстовым интерфейсом. В командной оболочке программы выполняются и результат выполнения отображается на экране в виде, сходном с интерпретатором Command.com MS-DOS. Командная оболочка Windows XP использует интерпретатор команд Cmd.exe, который загружает приложения и

направляет поток данных между приложениями, для перевода введенной команды в понятный системе вид.

Имеется возможность использовать командную оболочку для создания и редактирования пакетных файлов (также называемых сценариями), что позволит автоматизировать выполнение обычных задач. Например, можно использовать сценарии для автоматизации управления учетными записями пользователей и ежедневной архивацией в нерабочие часы. Также можно использовать сервер сценариев Windows, CScript.exe, для выполнения в командной оболочке сложных сценариев. Выполнение операций с помощью пакетных файлов является более эффективным, чем с помощью интерфейса пользователя. Пакетные файлы принимают все команды, доступные из командной строки. Дополнительные сведения о пакетных файлах и создании сценариев см. в разделе помощи OS "Использование пакетных файлов".

Имеется возможность настроить окно командной строки для облегчения просмотра и для увеличения контроля за выполнением программ. Дополнительные сведения о настройке окна командной строки см. в разделе помощи OS "Настройка командной строки".

## Использование синтаксиса команд

Синтаксическая структура выводится в том порядке, в котором следует вводить команду и следующие за ней параметры, если они есть. Следующий пример команды **xcopy** иллюстрирует разнообразие синтаксических форматов текста.

```
xcopy источник [результат] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d[:мм-дд-гггг]]
      [/u] [/i] [/s [/e]] [/t] [/k] [/r] [/h] [{/a|m}] [/n] [/o] [/x]
      [/exclude:файл1[+[файл2]][+[файл3]] [{/y|-y}] [/z]
```

В следующей таблице показано, как следует интерпретировать различные текстовые форматы.

### Форматирование

| Формат  | Значение   |
|---|--|
| <i>Курсив</i>   | Данные, которые должен ввести пользователь                     |
| <b>Полужирный шрифт</b>   | Элементы, которые следует вводить точно, как показано          |
| Пропуск (...)   | Параметры могут повторяться несколько раз в командной строке   |
| В квадратных скобках ([])   | Необязательные элементы  |
| В фигурных скобках ({}); варианты разделены вертикальной чертой (). Пример: {четные нечетные} | Набор значений, из которого можно выбрать только одно значение |
| Шрифт Courier   | Текст кода или выхода программы                                |

## Использование нескольких команд и символов условной обработки

Можно выполнять несколько команд из одной командной строки или сценария с помощью символов условной обработки. При использовании нескольких команд, содержащих символы условной обработки, выполнение команд, стоящих справа от символа условной обработки, будет проводиться в зависимости от результатов выполнения команды, стоящей слева от символа. Например, требуется, чтобы команда выполнялась, только если предыдущая команда не была выполнена успешно. Или требуется, чтобы команда выполнялась, только если предыдущая команда была выполнена успешно.

Для передачи нескольких команд можно использовать специальные символы, перечисленные в следующей таблице.

| Символ    | Синтаксис  | Определение   |
|-----------|--|---|
| & [...]   | команда1 & команда2  | Используется для разделения нескольких команд в одной командной строке. В Cmd.exe выполняется первая команда, затем вторая команда.   |
| && [...]  | команда1 && команда2   | Запускает команду, стоящую за символом &&, только если команда, стоящая перед этим символом была выполнена успешно. В Cmd.exe выполняется первая команда. Вторая команда выполняется, только если первая была выполнена успешно.                            |
| [...]     | команда1    команда2   | Запускает команду, стоящую за символом   , только если команда, стоящая перед символом    не была выполнена. В Cmd.exe выполняется первая команда. Вторая команда выполняется, только если первая не была выполнена (полученный код ошибки превышает ноль). |
| ( ) [...] | (команда1 & команда2)<br>; или ,    команда1 параметр1;параметр2 | Используется для группировки или вложения команд. Используется для разделения параметров команды.   |

### Примечания

- Амперсанд (&), вертикальная черта (|) и скобки ( ) являются специальными символами, которым должен предшествовать управляющий символ (^) или кавычки, если эти символы передаются в качестве аргументов.
- Если команда завершает операцию успешно, возвращается нулевой (0) код ошибки или не возвращается никакого кода. Дополнительные сведения о кодах завершения см. в разделе помощи OS "Пакеты Microsoft Windows Resource Kits".

## Вложенные командные оболочки

Имеется возможность вкладывать командные оболочки в Cmd.exe, открывая новый экземпляр Cmd.exe из командной строки. По умолчанию каждый экземпляр Cmd.exe наследует среду своего родительского приложения Cmd.exe. Вложение экземпляров Cmd.exe позволяет вносить в локальную среду изменения, которые не влияют на родительское приложение Cmd.exe. Это позволяет сохранять исходную среду Cmd.exe и возвращаться к ней после удаления вложенной командной оболочки. Изменения вложенной командной оболочки не сохраняются.

Чтобы создать вложенную командную оболочку, в командной строке введите:

**cmd**

Появится сообщение следующего вида:

**Microsoft (R) Windows XP (TM)**  
 (C) Copyright 1985-2001 Microsoft Corp.

Чтобы закрыть все вложенные командные оболочки, введите команду **exit**.

Можно еще уже локализовать изменения в экземпляре Cmd.exe (или в сценарии) с помощью команд **setlocal** и **endlocal**. Команда **setlocal** создает локальную область, а **endlocal** ее удаляет. Любые изменения, сделанные внутри области, созданной командами **setlocal** и **endlocal**, не учитываются; таким образом исходная среда остается без изменений. С помощью этих команд можно создать до 32 вложенных областей. Дополнительные сведения о командах **setlocal** и **endlocal** см. в разделах Setlocal и Endlocal.

## Использование переменных среды в Cmd.exe

Среда командной оболочки Cmd.exe определяется переменными, задающими поведение командной оболочки и операционной системы. Имеется возможность определить поведение среды командной оболочки или среды всей операционной системы с помощью двух типов переменных среды: системных и локальных. Системные переменные среды определяют поведение глобальной среды операционной системы. Локальные переменные среды определяют поведение среды в данном экземпляре Cmd.exe.

Системные переменные среды заданы заранее в операционной системе и доступны для всех процессов Windows XP. Только пользователи с привилегиями администратора могут изменять эти переменные. Эти переменные наиболее часто используются в сценариях входа в систему.

Локальные переменные среды доступны, только когда пользователь, для которого они были созданы, вошел в систему. Локальные переменные из куста **HKEY\_CURRENT\_USER** подходят только для текущего пользователя, но определяют поведение глобальной среды операционной системы.

В следующем списке представлены различные типы переменных в порядке убывания приоритета.

1. Встроенные системные переменные
2. Системные переменные куста **HKEY\_LOCAL\_MACHINE**
3. Локальные переменные куста **HKEY\_CURRENT\_USER**
4. Все переменные среды и пути указаны в файле Autoexec.bat.
5. Все переменные среды и пути указаны в сценарии входа в систему (если он имеется).
6. Переменные, используемые интерактивно в сценарии или пакетном файле

В командной оболочке каждый экземпляр Cmd.exe наследует среду своего родительского приложения. Поэтому можно изменять переменные в новой среде Cmd.exe, что не повлияет на среду родительского приложения.

В следующей таблице приведен список системных и локальных переменных среды для Windows XP.

| Переменная        | Тип       | Описание   |
|-------------------|-----------|--|
| %ALLUSERSPROFILE% | Локальная | Возвращает размещение профиля «All Users».                         |
| %APPDATA%         | Локальная | Возвращает используемое по умолчанию размещение данных приложений. |

|                          |           |  |
|--------------------------|-----------|--|
| %CD%                     | Локальная | Возвращает путь к текущей папке.   |
| %CMDCMDLINE%             | Локальная | Возвращает строку команд, с помощью которой был запущен данный экземпляр Cmd.exe.  |
| %CMDEXTVERSION%          | Системная | Возвращает номер версии текущих расширений обработчика команд.   |
| %COMPUTERNAME%           | Системная | Возвращает имя компьютера.   |
| %COMSPEC%                | Системная | Возвращает путь к исполняемой командной оболочке.  |
| %DATE%                   | Системная | Возвращает текущие данные. Использует тот же формат, что и команда <b>date /t</b> . Создается командой Cmd.exe. Дополнительные сведения о команде <b>date</b> см. в разделе Date.  |
| %ERRORLEVEL%             | Системная | Возвращает код ошибки последней использовавшейся команды. Значение, не равное нуля, обычно указывает на наличие ошибки.  |
| %HOMEDRIVE%              | Системная | Возвращает имя диска локальной рабочей станции, связанного с основным каталогом пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы». |
| %HOMEPATH%               | Системная | Возвращает полный путь к основному каталогу пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы».                                     |
| %HOMESHARE%              | Системная | Возвращает сетевой путь к общему основному каталогу пользователя. Задается на основании расположения основного каталога. Основной каталог пользователя указывается в оснастке «Локальные пользователи и группы».                             |
| %LOGONSEVER%             | Локальная | Возвращает имя контроллера домена, который проверял подлинность текущей сессии.  |
| %NUMBER_OF_PROCESSORS%   | Системная | Задает количество процессоров, установленных на компьютере.  |
| %OS%                     | Системная | Возвращает имя операционной системы. При использовании Windows 2000 имя операционной системы отображается как Windows_NT.  |
| %PATH%                   | Системная | Указывает путь поиска для исполняемых файлов.  |
| %PATHEXT%                | Системная | Возвращает список расширений файлов, которые рассматриваются операционной системой как исполняемые.  |
| %PROCESSOR_ARCHITECTURE% | Системная | Возвращает архитектуру процессора.<br>Значения: x86, IA64.   |
| %PROCESSOR_IDENTIFIER%   | Системная | Возвращает описание процессора.  |
| %PROCESSOR_LEVEL%        | Системная | Возвращает номер модели процессора, установленного на компьютере.  |
| %PROCESSOR_REVISION%     | Системная | Возвращает номер модификации процессора.   |

|                |                              |  |
|----------------|------------------------------|--|
| %PROMPT%       | Локальная                    | Возвращает параметры командной строки для текущего интерпретатора. Создается командой Cmd.exe.   |
| %RANDOM%       | Системная                    | Возвращает произвольное десятичное число от 0 до 32767. Создается командой Cmd.exe.  |
| %SYSTEMDRIVE%  | Системная                    | Возвращает имя диска, содержащего корневой каталог Windows XP (т. е. системный каталог).   |
| %SYSTEMROOT%   | Системная                    | Возвращает размещение системного каталога Windows XP.  |
| %TEMP% и %TMP% | Системная и пользовательская | Возвращает временные папки, по умолчанию используемые приложениями, которые доступны пользователям, выполнившим вход в систему. Некоторые приложения требуют переменную TEMP, другие — переменную TMP. |
| %TIME%         | Системная                    | Возвращает текущее время. Использует тот же формат, что и команда <b>time /t</b> . Создается командой Cmd.exe. Дополнительные сведения о команде <b>time</b> см. в разделе Time.                       |
| %USERDOMAIN%   | Локальная                    | Возвращает имя домена, содержащего список учетных записей пользователей.   |
| %USERNAME%     | Локальная                    | Возвращает имя пользователя, выполнившего вход в систему.  |
| %USERPROFILE%  | Локальная                    | Возвращает размещение профиля для текущего пользователя.   |
| %WINDIR%       | Системная                    | Возвращает размещение каталога операционной системы.   |

## Установка переменных среды

Используйте команду **set** для создания, удаления или отображения переменных среды. Команда **set** изменяет переменные только в среде текущей оболочки.

Чтобы отобразить переменную, в командной строке введите:

**set имя\_переменной**

Чтобы добавить переменную, в командной строке введите:

**set variablename=значение**

Чтобы удалить переменную, в командной строке введите:

**set имя\_переменной=**

Имеется возможность использовать большинство символов в качестве значений переменных, в том числе пробел. При использовании специальных символов, таких как <, >, &, или ^, перед ними следует помещать управляемый символ (^) или кавычки. При использовании кавычек они включаются в значение переменной, так как все символы, следующие после знака равенства принимаются за значение переменной. Учтите следующие примеры.

- Чтобы создать значение переменной **new&name**, введите:

```
set varname=new^&name
```

- Чтобы создать значение переменной "**"new&name"**", введите:

```
set varname="new&name"
```

- При вводе в командной строке **set varname=new&name** появляется следующее сообщение об ошибке:

```
"'name' is not recognized as an internal or  
external command, operable program or batch file."
```

В именах переменных регистр символов не учитывается. Однако, команда **set** отображает значение переменной так, как оно было введено. Можно комбинировать символы нижнего и верхнего регистра в именах переменных, чтобы облегчить восприятие кода (например, ИмяПользователя).

### Примечания

- Максимальный размер отдельной переменной среды составляет 8192 байта.
- Максимальный общий размер всех переменных среды, включая имена переменных и знак равенства, составляет 65 536 Кбайт.

## Подстановка значений в переменные среды

Чтобы иметь возможность подставлять значения в переменную среды из командной строки или из сценариев, следует заключить имя соответствующей переменной в символы процентов (**%имя\_переменной%**). Символы процентов указывают на то, что Cmd.exe должен обратиться к значениям переменных, а не делать посимвольное сравнение. После определения значения для имени переменной, заключите имя переменной в символы процентов. Cmd.exe проводит поиск всех вхождений имени переменной и заменяет его на определенное значение переменной. Например, требуется создать сценарий, содержащий различные значения (например, имена пользователей), и требуется определить соответствующее значение переменной среды USERNAME для каждого пользователя. Для этого следует написать сценарий с использованием переменной USERNAME, заключенной в кавычки. При выполнении сценария Cmd.exe заменит вхождения **%USERNAME%** соответствующими значениями, что избавит от необходимости делать это вручную для каждого пользователя. Подстановка значений не является рекурсивной. Cmd.exe проверяет переменные один раз. Дополнительные сведения о подстановке значений в переменные см. в разделах For и Call.

Более подробно с общим описанием можно познакомится в разделе помощи "Командная строка".

Получить весь список команд можно по команде HELP.

## Команды ОС MS DOS

Если в командной строке оболочки выполнить команду COMMAND или непосредственно выполнить указанную команду, то вызовется командная оболочка ОС MS DOS.

Для получения списка команд этой оболочки достаточно выполнить команду HELP. Ниже приводится полный список команд.

|          |   |
|----------|---|
| ASSOC    | Вывод либо изменение сопоставлений по расширениям имен файлов.                                      |
| AT       | Выполнение команд и запуск программ по расписанию.  |
| ATTRIB . | Отображение и изменение атрибутов файлов  |
| BREAK    | Включение/выключение режима обработки комбинации клавиш CTRL+C.                                     |
| CACLS    | Отображение/редактирование списков управления доступом (ACL) к файлам.                              |
| CALL     | Вызов одного пакетного файла из другого.  |
| CD       | Вывод имени либо смена текущей папки.   |
| CHCP     | Вывод либо установка активной кодовой страницы.   |
| CHDIR    | Вывод имени либо смена текущей папки.   |
| CHKDSK   | Проверка диска и вывод статистики.  |
| CHKNTFS  | Отображение или изменение выполнения проверки диска во время загрузки.                              |
| CLS      | Очистка экрана.   |
| CMD      | Запуск еще одного интерпретатора командных строк Windows.   |
| COLOR    | Установка цвета текста и фона, используемых по умолчанию.   |
| COMP     | Сравнение содержимого двух файлов или двух наборов файлов.  |
| COMPACT  | Отображение/изменение сжатия файлов в разделах NTFS.  |
| CONVERT  | Преобразование дисковых томов FAT в NTFS. Нельзя выполнить преобразование текущего активного диска. |
| COPY     | Копирование одного или нескольких файлов в другое место.  |
| DATE     | Вывод либо установка текущей даты.  |
| DEL      | Удаление одного или нескольких файлов.  |
| DIR      | Вывод списка файлов и подпапок из указанной папки.  |
| DISKCOMP | Сравнение содержимого двух гибких дисков.   |
| DISKCOPY | Копирование содержимого одного гибкого диска на другой.   |
| DOSKEY   | Редактирование и повторный вызов командных строк; создание макросов.                                |
| ECHO     | Вывод сообщений и переключение режима отображения команд на экране.                                 |
| ENDLOCAL | Конец локальных изменений среды для пакетного файла.  |
| ERASE    | Удаление одного или нескольких файлов.  |
| EXIT     | Завершение работы программы CMD.EXE (интерпретатора командных строк).                               |
| FC       | Сравнение двух файлов или двух наборов файлов и вывод различий между ними.                          |
| FIND     | Поиск текстовой строки в одном или нескольких файлах.   |
| FINDSTR  | Поиск строк в файлах.   |
| FOR      | Запуск указанной команды для каждого из файлов в наборе.  |
| FORMAT   | Форматирование диска для работы с Windows.  |
| FTYPE    | Вывод либо изменение типов файлов, используемых при сопоставлении по расширениям имен файлов.       |
| GOTO     | Передача управления в отмеченную строку пакетного файла.  |
| GRAFTABL | Позволяет Windows отображать расширенный набор символов в графическом режиме                        |
| HELP     | Выводит справочную информацию о командах Windows.   |
| IF       | Оператор условного выполнения команд в пакетном файле.  |

|          |   |
|----------|---|
| LABEL    | Создание, изменение и удаление меток тома для дисков.   |
| MD       | Создание папки.   |
| MKDIR    | Создание папки.   |
| MODE     | Конфигурирование системных устройств.   |
| MORE     | Последовательный вывод данных по частям размером в один экран.                                    |
| MOVE     | Перемещение одного или нескольких файлов из одной папки в другую.                                 |
| PATH     | Вывод либо установка пути поиска исполняемых файлов.  |
| PAUSE    | Приостановка выполнения пакетного файла и вывод сообщения.  |
| POPD     | Восстановление предыдущего значения текущей активной папки, сохраненного с помощью команды PUSHD. |
| PRINT    | Вывод на печать содержимого текстовых файлов.   |
| PROMPT   | Изменение приглашения в командной строке Windows.   |
| PUSHD    | Сохранение значения текущей активной папки и переход к другой папке.                              |
| RD       | Удаление папки.   |
| RECOVER  | Восстановление читаемой информации с плохого или поврежденного диска.                             |
| REM      | Помещение комментариев в пакетные файлы и файл CONFIG.SYS.  |
| REN      | Переименование файлов и папок.  |
| RENAME   | Переименование файлов и папок.  |
| REPLACE  | Замещение файлов.   |
| RMDIR    | Удаление папки.   |
| SET      | Вывод, установка и удаление переменных среды Windows.   |
| SETLOCAL | Начало локальных изменений среды для пакетного файла.   |
| SHIFT    | Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла.                       |
| SORT     | Сортировка ввода.   |
| START    | Запуск программы или команды в отдельном окне.  |
| SUBST    | Сопоставляет заданному пути имя диска.  |
| TIME     | Вывод и установка системного времени.   |
| TITLE    | Назначение заголовка окна для текущего сеанса интерпретатора командных строк CMD.EXE.             |
| TREE     | Графическое отображение структуры папок заданного диска или заданной папки.                       |
| TYPE     | Вывод на экран содержимого текстовых файлов.  |
| VER      | Вывод сведений о версии Windows.  |
| VERIFY   | Установка режима проверки правильности записи файлов на диск                                      |
| VOL      | Вывод метки и серийного номера тома для диска.  |
| XCOPY    | Копирование файлов и дерева папок.  |

Подробно со всеми командами можно ознакомится [здесь](#) ( [здесь архив файла в формате word](#) ).

Разработано большое количество разных учебных программ. Сохраните [этот архив](#) на Вашем локальном диске. После разархивации файлов - в режиме командной строки сделайте каталог с программными файлами текущим и выполните программу USERDOS.EXE.

## Пакетные (процедурные) файлы.

Чтобы избежать утомительного набора часто повторяющейся последовательности команд, вы можете поместить эти командные строки в файл и присвоить последнему расширение .bat. Теперь вам достаточно набрать имя этого файла и нажать ввод, чтобы содержащиеся в пакетном файле команды начались выполняться в той последовательности, в которой они следуют в пакетном файле (порядок следования можно сделать условным).

Создать пакетный файл можно с помощью редактора, например, EDIT.

В качестве последней командной строки файла можно указать имя другого пакетного файла. Используя команду CALL можно вызывать другие командные файлы как подпрограммы.

Вы можете помещать в начало командных строк символ @ для отмены вывода командной строки на экран (эхо) во время ее исполнения.

Разрешается использование таких средств, как <>>> и |.

Если существуют одновременно несколько внешних команд с одинаковым именем, то они выполняются в следующем порядке (в зависимости от расширения):

.com

.exe

.bat

Наиболее простой способ создания пакетного файла приводится ниже:

- Введите командную строку:

copy con имя\_пакетного\_файла.bat

- Нажмите ВВОД. Теперь вводимая с консоли (клавиатура) информация заносится в файл.

- Введите командную строку и нажмите ВВОД.

- Введите командную строку и нажмите ВВОД.

- Введите командную строку и нажмите ВВОД.

- ... и т.д.

- После введения последней командной строки нажмите одновременно клавиши Ctrl и Z (CONTROL+Z).

- Нажмите ВВОД.

- На экране появится сообщение:

1 File(s) copyied

(Скопирован 1 файл )

Теперь для запуска созданного пакетного файла введите его имя (расширение не обязательно) и нажмите ВВОД.

Дополнительную гибкость пакетному файлу придают формальные параметры, реальные значения которых подставляются в ходе выполнения пакетного файла. Эти параметры имеют вид %1 - %9, например dir %1 > %2.

Параметр %0 всегда заменяется системой на имя дисковода (если указано) и имя пакетного файла.

Если вы хотите использовать больше десяти параметров, то следует использовать команду shift.

Если знак процента является частью имени файла, то этот знак необходимо указывать дважды: abc%%d.exe.

Помимо десяти числовых формальных параметров вы можете использовать именованные формальные параметры, указываемые между двумя знаками процентов: %parameter%. Отличие их от числовых параметров заключается в том, что их реальные значения указываются вами не в командной строке вызова пакетного файла, а в среде (с помощью команды set), откуда MS-DOS будет выбирать их и подставлять вместо формальных параметров.

При вводе командной строки вызова пакетного файла необходимо помнить, что последовательность в ней реальных значений для числовых параметров этого пакетного файла должна соответствовать нумерации этих числовых параметров.

## Команды пакетной обработки

Ниже приведены основные команды пакетной обработки:

|                              |   |
|------------------------------|---|
| <a href="#"><u>call</u></a>  | Вызов одного пакетного файла из другого, без завершения последнего. |
| <a href="#"><u>echo</u></a>  | Вывод или отмена вывода эха или вывод текущего состояния.           |
| <a href="#"><u>for</u></a>   | Выполнение команды для группы файлов.                               |
| <a href="#"><u>goto</u></a>  | Переход на указанную метку.   |
| <a href="#"><u>if</u></a>    | Выполнение команды при наличии некоторого условия.                  |
| <a href="#"><u>pause</u></a> | Пауза в процессе выполнения файла.                                  |
| <a href="#"><u>rem</u></a>   | Вывод коментария.   |
| <a href="#"><u>shift</u></a> | Увеличение количества формальных параметров.                        |

Все команды пакетной обработки являются резидентными (внутренними).

Практически всегда при написании процедурных файлов удобно использовать [команды установки среды](#), команды [переадресации стандартного ввода-вывода](#) и пользовательские средства ведения диалога. Ниже дается краткое описание этих команд.

## SET

Цель: Формирование среды.

Синтаксис:

```
set [строка1]=[строка2]]
```

Если указываемая первая строка уже существует в среде, то ее параметр в среде заменяется на новый параметр (вторая строка). Если вы укажете только первую строку, то параметр этой строки в среде будет отменен. Если вы введете только саму команду без опций, то на экран будет выведены все текущие установки среды.

Описываемые возможности весьма удобны в пакетной обработке. Так например, если в пакетном файле есть команда

```
type %file%
```

то с помощью команды set вы можете, не модифицируя пакетного файла, задавать значение переменной file в среде. MS-DOS при выполнении пакетного файла будет выбирать из среды соответствующее значение, например moofile.txt:

```
set file=moofile.txt
```

## Переадресовка ввода/вывода

По умолчанию, стандартным выводом большинства программ является экран. Если вам необходимо переадресовать вывод программы в файл, то в командной строке после имени команды достаточно указать символ (>), после которого должно следовать имя файла приемника:

```
dir > myfile.doc
```

Если файл-приемник уже существовал, то старое содержимое файла будет уничтожено. Поэтому, если вам нужно, чтобы вывод какой-либо программы просто присоединялся к концу файла, то нужно указать два символа (>>):

```
dir a:\files >> myfile.doc
```

Переадресовка ввода указывается символом (<). Например, по следующей команде сортируется содержимое файла names, а результат сортировки записывается в файл namelist:

```
sort < names > namelist
```

MS-DOS включает специальные команды, называемые фильтрами. Они предназначены для преобразования вывода ваших программ, например - сортировки (фильтр sort в предыдущем примере), поиска текста в файле (find), вывода содержимого файла на экран постранично (more).

Кроме этого, у вас есть возможность адресовать вывод одной программы на вход другой с помощью символа () .

Например, следующая команда выводит содержимое каталога на экран в отсортированном по алфавиту порядке:

```
dir | sort
```

или

```
dir | sort | more
```

Вывод из предыдущего примера можно переадресовать в файл, в myfile.lst, например. Пусть он находится на диске в дисководе D:

```
dir | sort > d:myfile.lst
```

## **Описание основных команд пакетной обработки.**

### **CALL**

Цель: Вызов одного пакетного файла из другого, без завершения последнего.

Синтаксис:

```
call [дисковод:][маршрут] имя_пакетного_файла [команда]
```

где: команда - команда, выполняемая после выполнения пакетного файла.

Комментарий: При указании имени пакетного файла необходимо указывать расширение .bat. По выполнении пакетного файла возобновляется выполнение вызывавшего пакетного файла с команды указанной в командной строке вызова. Если опция "команда" опущена, то выполнение возобновляется с команды, непосредственно следующей за командой call.

Примечание: Нельзя использовать символы переадресации ввода/вывода с командой Call. Пакетный файл может выполнять рекурсивное обращение к самому себе, при условии, что предусмотрено условие окончания.

### **ECHO**

Цель: Вывод или отмена вывода эха или вывод текущего состояния.

Синтаксис:

```
echo [on]
```

или

```
echo [off]
```

или

`echo [сообщение]`

**Комментарий:** Обычно при выполнении команд соответствующие командные строки выводятся на экран (эхо). Отменить вывод эха можно с помощью опции off, а восстановить с помощью опции on. Если вы введете команду echo без опций, то на экране появится текущее значение (on или off).

Опцию "сообщение" можно применять для вывода сообщений на экран во время выполнения пакетного файла.

Символ `@`, помещенный в начало командной строки, отменяет эхо этой строки.

Используя переадресацию ввода-вывода возможно чтение-запись в файлы любой информации из процедурного файла и из параметров среды.

## FOR

**Цель:** Выполнение команды для группы файлов.

**Синтаксис:**

`for %%символ in группа do команда` (для пакетной обработки)

или

`for %символ in группа do команда` (для интерактивной обработки)

**Комментарий:** Описываемая команда присваивает каждому элементу группы переменную %%символ. "Символ" не должен быть 0-9, чтобы не совпадать с %0-%9. "Группа" имеет следующий формат:

`( элемент элемент ...)`

или

если элемент (имя файла) содержит символы заполнители, то все элементы, кроме этого элемента, игнорируются, а всем соответствующим файлам присваивается переменная %%символ.

**Пример:** Следующая команда присваивает переменную %%f всем файлам с расширением .txt и распечатывает их:

`for %%f in (*.txt) do print %%f`

Если вы используете команду for вне пакетного файла (интерактивная обработка), то вы должны указывать только один знак процента перед "символом".

## GOTO

**Цель:** Переход на указанную метку, т.е. переход к выполнению командной строки, "помеченной" указываемой меткой.

**Синтаксис:**

```
goto [:]метка
```

**Комментарий:** Метка может содержать пробелы, но не другие разделители, такие как точка с запятой или знак равно. Если пакетный файл не содержит метки, то выполнение пакетного файла прерывается.

**Примечание:** Любая строка пакетного файла, начинающаяся с символа (:), игнорируется во время выполнения пакетного файла:

```
:begin
echo off
format a: /s
if errorlevel 0 goto end
echo ОШИБКА ФОРМАТИРОВАНИЯ
:end
echo КОНЕЦ
```

## IF

**Цель:** Выполнение команды при наличия некоторого условия.

**Синтаксис:**

```
if [not] errorlevel код_завершения команда
```

или

```
if [not] строка1 == строка2 команда
```

или

```
if [not] exist имя_файла команда
```

**Комментарий:** Команда выполняется, а не игнорируется, если указываемое условие истинно. Описание условий приводится ниже:

код\_завершения - Число. При его совпадении с кодом завершения предыдущей команды, выполненной командным процессором, или при большем по значению коде завершения, условие считается истинным.

строка1 == строка2 - Условие считается истинным, только если строки идентичны после подстановки параметров. Строки не могут содержать такие разделители, как

запятая, точка с запятой, знак равно или пробел.

имя\_файла Условие считается истинным, только если указанный файл существует.

Если вы укажете опцию [not], то команда будет выполняться, только если условие ложно:

if not exist moofile.doc echo нет такого файла.

## PAUSE

Цель: Пауза в процессе выполнения файла.

Синтаксис:

pause [текст]

Комментарий: Когда командный процессор в ходе выполнения пакетного файла встречает команду pause, то выполнение пакетного файла прерывается до тех пор, пока вы не нажмете какую-либо клавишу. При этом на экран выводится текст (если был указан в командной строке). Кроме этого на экран выводится сообщение:

Strike a key when ready .(Когда будете готовы нажмите любую клавишу)

Если вы нажмете CONTROL+C, то на экране появится следующий вопрос:

Terminate batch job (Y,N)? (Прервать выполнение пакетного файла (Д/Н)? )

Примечание: Если эхо отменено, то ни строка pause ни строка текста не появятся на экране.

## REM

Цель: Вывод комментария.

Синтаксис:

rem [комментарий]

Комментарий: В строке комментария разрешены в качестве разделителей только пробелы, символы табуляции и запятые.

Примечание: Если эхо отменено, то строка комментария не выводится.

## SHIFT

Цель: Смещение формальных параметров пакетного файла.

Синтаксис:

**shift**

**Комментарий:** С помощью команды shift вы можете получить доступ к более чем десяти параметрам, т.е. те параметры, которые следуют за десятым (%9) сдвигаются по одному к %9.

Команду shift можно использовать, даже если количество параметров меньше десяти. Обратите внимание на то, что обратной команды shift не существует, так например, вы не сможете восстановить параметр %0, существовавший до сдвига.

**Пример:** Следующая пакетная программа копирует все перечисленные в командной строке файлы в указанный каталог:

```
set todir = %1
:one
shift
if "%1"=="" goto two
copy %1 %todir%
goto one
:two
set todir=
echo конец
```

Командная строка описанного выше примера будет иметь формат:

имя\_пакетного\_файла имя\_каталога имя\_файла имя\_файла.....

Пример обработки процедурой any\_file.bat всех файлов, находящихся в заданной директории:

Создаем «главный» файл all\_file.bat

echo on

for %%z in (%1\*.\* ) do call any\_file %%z %0 %1

При его выполнении для каждого файла из директории %1 по маске \*.\* будет выполнена процедура any\_file. Для этой процедуры первым параметром командной строки будет имя файла. Пример такой процедуры:

echo %1 %2 %3

## Задание лабораторной работы

Разработать командный (процедурный) файл согласно указанного ниже варианта задания. Выполнить его отладку и тестирование. В формулировке задания указание - заданный, указанный и т.п. относится к имени объекта, которое задается как параметр командной строки.

1. В указанной директории удалить все файлы с указанными типами. Типы файлов задаются в командной строке.
2. Вывести на экран списки файлов из указанных директорий согласно указанному шаблону.
3. Удалить из указанных директорий файлы, имена которых заданы шаблоном.
4. Скопировать из заданных директорий все файлы в указанную директорию. Если указанной директории не существует - создать ее.
5. Создать в заданной директории поддиректории, имена которых совпадают с типами файлов находящихся в заданной директории. (для совместимости - считать, что расширения имен файлов - заданы).
6. Скопировать из заданной директории все файлы в поддиректории так, что бы все файлы типа \*.exe были скопированы в поддиректорию EXE, а остальные файлы - в поддиректорию XXX.
7. Из заданной директории переписать указанные шаблоном файлы в заданную директорию так, что бы были переписаны только "новые" (не существующие в приемной директории) файлы.
8. Из заданной директории переписать указанные шаблоном файлы в заданную директорию так, что бы были переписаны только "старые" (существующие в приемной директории) файлы.
9. Из заданной директории переписать указанные шаблоном файлы в заданную директорию так, что бы в случае совпадения имен файлов, переписываемым файлам присвоить новые имена. Новые имена построить по шаблону N\_старое\_имя.
10. Вывести в указанный файл имена файлов из указанных директорий.
11. На заданном диске создать систему директорий: в главном каталоге - указанная. В указанной - поддиректории с именами EXE, ASM, LST.
12. Процедуру записи на дискету указанных шаблоном файлы. Дискету перед записью необходимо форматировать.
13. Удалить из заданной директории все файлы, если в этой директории существует указанный файл.
14. Удалить из заданной директории все файлы, если в этой директории не существует указанный файл.
15. Удалить заданную директорию. Предположить, что в указанная директория содержит только файлы.
16. Из указанной директории удалить все файлы, кроме тех, которые заданы шаблоном.
17. Всем файлам из указанной директории присвоить атрибуты "только чтение", кроме файлов, тип которых \*.txt.
18. Процедуру организующую формирования дискеты. Процедура должна показать пользователю список файлов и директорий, которые находятся на дискете и предоставить пользователю возможность отменить формирование.

19. Процедуру организующую копирование указанных шаблоном файлов на произвольное количество дисков. На каждую дискету копируются все указанные файлы.

20. Процедуру генерирующую файл с указанным именем и содержащий в каждой строке следующую информацию:

первый\_параметр\_ком.\_строки второй параметр\_ком.\_строки имя файла (из указанной директории по указанному шаблону)

## **Контрольные вопросы.**

1. Понятие текущей директории и текущего диска. Команды изменения текущей директории и текущего диска.
2. Стока приглашения командного режима. Команда смены строки приглашения.
3. Команда создания и удаления директорий.
4. Команды переименования и копирования файлов.
5. Команды организации "среды выполнения" (set).
6. Команды подготовки новых дисков к работе.
7. Процедурные файлы. Назначение и основные правила создания.
8. Основные команды для создания процедурных файлов.
9. Команда dir, more. Их параметры и варианты применения.
10. Атрибуты файлов. Их назначение и использование.
11. Принцип поиска внешней команды (программы) командной оболочкой. Команда set и path.

## **Требование к оформлению отчета.**

Отчет подготавливается на машинном носителе в формате Microsoft Word. Титульная страница оформляется по обще принятым правилам. На ней в обязательном порядке указывается тема работы. В текст отчета необходимо включить постановку задачи, листинг (снимок экрана, текст файла и т.п.) разработанного файла и результат тестирования.

Файл, содержащий текст отчета, именуется по следующему шаблону:

шифр\_группы-фамилия\_студента-дисциплина-номер\_работы.doc

где - дисциплина указывается **SP\_OS** (системное программирование и ОС)

## **Пример процедурного файла**

В указанной директории удалить все файлы с указанными типами. Типы файлов задаются в командной строке.

```
echo on
set deldir = %1
:one
shift
if "%1"=="" goto two
```

```
del %deldir%/*.%1  
goto one  
:two  
set deldir=
```

**Успешной работы.**