**Лабораторная работа №1**

**Командная оболочка ОС Windows, основные команды, пакетные файлы.**

Командная оболочка — это отдельный программный продукт, который обеспечивает прямую связь между пользователем и операционной системой (ОС). Текстовый пользовательский интерфейс в виде командной строки предоставляет среду, в которой выполняются команды, программы и служебные утилиты с текстовым интерфейсом. В командной оболочке и результат выполнения утилит и программ отображается на экране в виде, сходном с командным интерпретатором Command.com MS-DOS. Командная оболочка ОС Windows использует интерпретатор команд Cmd.exe, который осуществляет перевод введенной команды в понятный ОС вид, загружает приложения (утилиты) и управляет потоками данных между ними.

Имеется возможность использовать командную оболочку для создания и редактирования пакетных файлов (текстовых файлов с расширением **.bat**, также называемых сценариями), что позволяет 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 ОС Windows, Cscript.exe, для выполнения сложных сценариев посредством командной оболочки. Выполнение операций с помощью пакетных файлов является более эффективным, чем с помощью текстового интерфейса пользователя. Командные или пакетные файлы принимают все команды, доступные из командной строки.

К числу основных команд и служебных утилит, используемых при работе с файлами, дисками и томами в ОС Windows посредством командной оболочки, относятся, приведенные в списке 1:

Список 1:

**Assoc, Attrib, Cacls, Cd, Chdir, Chkdsk, Chkntfs, Comp, Compact, Convert, Copy, Date, Del, Dir, Diskcomp, Diskcopy, Erase, Fc, Find, Findstr, Format, Label, Md, Mkdir, Move, Print, Rd, Recover, Ren, Rename, Replace, Rmdir, Subst, Tree, Type, Vol, Xcopy.**

Справку также можно получить, набрав в окне командной оболочки строку Help и нажав Enter для ввода.

В пакетных файлах можно использовать практически все команды ОС Windows. Но есть несколько специальных команд, которые обретают смысл только в пакетных файлах. Таких специальных команд немного, но они играют важную роль при создании логики командного файла. В Таблице 1 представлен список этих команд и их краткое описание.

|  |  |
| --- | --- |
| **Команда** | **Краткое описание** |
| call | Вызов одного пакетного файла из другого. |
| echo | Вывод сообщений и переключение режима отображения команд на экране. |
| rem | Помещение комментариев в пакетные файлы |
| pause | Приостановка выполнения пакетного файла и вывод сообщения. |
| if | Оператор условного выполнения команд в пакетном файле. |
| goto | Передача управления в отмеченную строку пакетного файла. |
| choice | Получение от пользователя ответа на поставленный заранее или в самой команде вопрос |
| shift | Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла. |
| for | Запуск указанной команды для каждого из файлов в наборе. |

Таблица 1.

**Подсказку по каждой команде можно получить, указав после имени команды ключ /?**

Более подробное описание применения команд можно найти на сайте **https://ss64.com**

**Примеры пакетных файлов:**

1. Смысл этого файла в том, чтобы он создает новую папку BACKUP и копирует в нее содержимое текущей папки.

@**echo** off

cls

**echo** start backup

**mkdir** backup

**copy \***.**\*** backup

**echo** end backup

Для проверки работы этого файла можно создать папку BAT, создать в ней несколько файлов (например, текстовых), поместить в нее файл backup.bat, содержащий перечисленные выше команды, и запустить его на выполнение.

Использование символа @ перед командой отключения отображения команд ECHO OFF позволяет отключить вывод самой этой команды на экран. В результате на экране появляются только результаты выполнения команд и системные сообщения.

1. Для создания файла в процессе выполнения пакетного файла используется символ перенаправления. Он выглядит так: >

Например, можно создать файл-отчет о работе bat файла с именем, равным дате запуска bat файла:

**set datetemp=%date:~-10%**

**@echo .>%DATETEMP%.txt**

Сначала в памяти создаем переменную datetemp, которой присваиваем 10 символов справа налево от системной переменной DATE. Таким образом, теперь во временной переменной datetemp содержится только текущая дата. Следующей строкой перенаправляем вывод символа точка в файл, имя которого берем из переменной datetemp, а расширение txt указываем явно.

1. Использование команды *for* для последовательного вывода на экран монитора содержимого всех текстовых файлов *txt* текущего каталога:

**@echo off**

**cls**

**for %%f in (\*.txt) do type %%f**

**Задания 1.**

Для каждой из команд, приведенных в списке 1 выяснить её назначение и привести пример (скриншот) работы команды.

**Задание 2.**

Создать пакетный файл и выполнить по вариантам задание:

1. С помощью цикла FORсоздать 20 пустых текстовых файлов (txt) с нумерованными именамиFILE01.txt, FILE02.txt и т.д. и 10 пустых файлов с расширением inf. Перенести все файлы inf в директорию, название которой передать в скрипт в качестве аргумента.
2. Написать командный файл, получающий в качестве аргумента расширение файла (cmd или bat). В зависимости от аргумента все файлы указанного расширения из текущей директории переносятся в поддиректорию CMD\_ДАТА или BAT\_ДАТА соответственно (ДАТА – текущая календарная дата в формате дд.мм.гггг). Поддиректории также создаются в коде командного файла. Необходимые файлы cmd и bat создать вручную заранее, минимум по три для каждого расширения.
3. Написать командный файл, проверяющий наличие в текущей директории файла system\_ДАТА.log. В случае его отсутствия создать пустой файл с указанным именем. Если файл существует – предварительно удалить старый файл и создать новый. ДАТА – текущая календарная дата в формате дд.мм.гггг.
4. В произвольной директории вручную создать файлы с разными расширениями, не менее 10, и не менее 3 текстовых файлов. Во все файлы с расширением txt добавить в конец текст "ENDTEXTFILE" (без кавычек).
5. В качестве аргументов передать командному файлу количество файлов и расширение (любое текстовое). Через цикл FOR создать переданное в аргументах количество файлов указанного расширения с произвольным шаблоном имени.
6. Создать командный файл, производящий в текущей директории объединение всех файлов заданного расширения (любого текстового) в новый файл OUTPUT.txt. Расширение передать в качестве аргумента командного файла. Предварительно создать нужные файлы, заполнив их произвольной информацией.
7. Создать не менее четырёх произвольных директорий, заполнить их произвольным количеством файлов любого типа. Создать командный файл, осуществляющий перенос файлов из одной директории в другую (напр. из DIR1 вDIR2). Имена директорий передать в качестве аргументов командного файла.
8. Написать командный файл, создающий текстовый файл с пронумерованными строками. Количество строк и имя файла передать в качестве аргументов командного файла.
9. Написать командный файл создающий определенное количество текстовых файлов. Количество файлов передать в качестве аргумента. Имена файлов задать по шаблону file\_NUMBER.txt, где NUMBER – порядковый номер файла. Каждый текстовый файл заполняется пронумерованным количеством строк равным его порядковому номеру.
10. Написать командный файл создающий текстовый файл со списком всех файлов только в указанной директории. Путь к директории передать в качестве аргумента.
11. Написать командный файл создающий текстовый файл со списком всех файлов в указанной директории и во всех вложенных директориях. Путь к директории передать в качестве аргумента.