***Лабораторная работа 3***

**Тема :Работа с командной строкой. Сетевая активность. Пакетные файлы.**

***Цель работы:*** *получение практических навыков по работе с Командной строкой и по выявлению* *вредоносных программ на компьютере с Microsoft* *Windows XP с помощью Командной строки.*

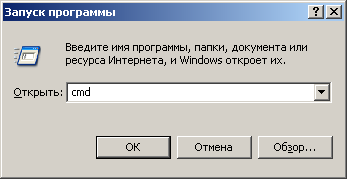
**Задание 1. Работа с Командной строкой**

В операционной системе Windows набираемые с клавиатуры и сразу же выполняемые команды выполняются с помощью так называемого командного интерпретатора, иначе называемого командным процессором или оболочкой командной строки **(command shell)**.

Таким образом, оболочка командной строки — 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 операционных системах Windows 9.x командный интерпретатор был представлен исполняемых файлом command.com (как и в MS-DOS). Начиная с Windows NT он реализован cmd.exe и обладает большими возможностями.

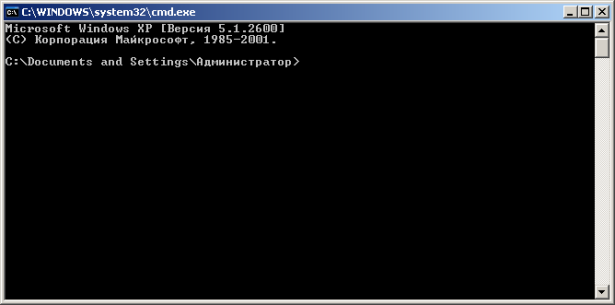
В Windows XP файл **Cmd.exe**, как и другие исполняемые файлы, соответствующие внешним командам операционной системы, находятся в каталоге SYSTEM32.

1. Для открытия сеанса работы с командной строкой выберите **Пуск->Выполнить**. В открывшемся окне наберите **cmd** и нажмите на ОК.



**Рис. 1.**

В результате откроется новое окно, в котором можно запускать команды и видеть результат их работы.



**Рис. 2.**

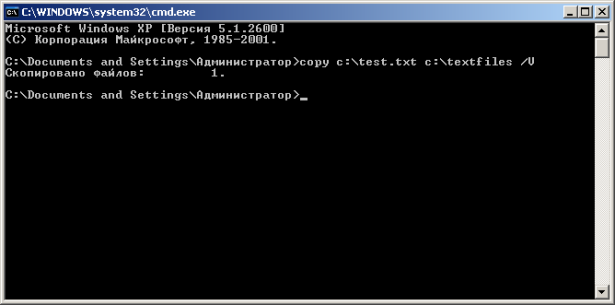
Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором — такие команды называются **внутренними** (например, copy или dir). Другие команды операционной системы представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и Cmd.exe, которые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

1. Для того чтобы выполнить команду введите имя этой команды (регистр не важен), ее параметры и ключи (если они необходимы) и нажмите клавишу Enter. 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]

Создайте на диске C файл **test.txt** и папку **Text files**. В Командной строке наберите

**copy C:\test.txt C:\Textfiles /V**



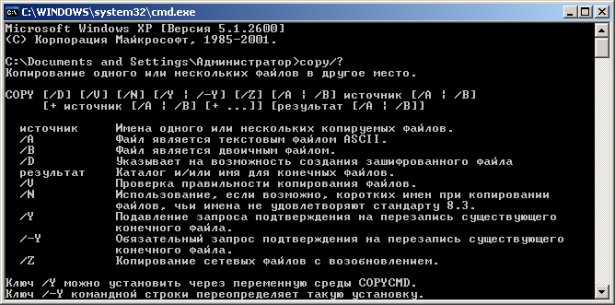
**Рис. 3.**

Имя команды здесь — **copy**, параметры — **C:\test.txt** и **C:\TextFiles**, а ключом является **/V**. Отметим, что в некоторых командах ключи могут начинаться не с символа /, а с символа – (минус), например, **-V**.

Следует обратить внимание, что если в имени папки есть пробел, в команде его быть не должно, как в указанном примере.

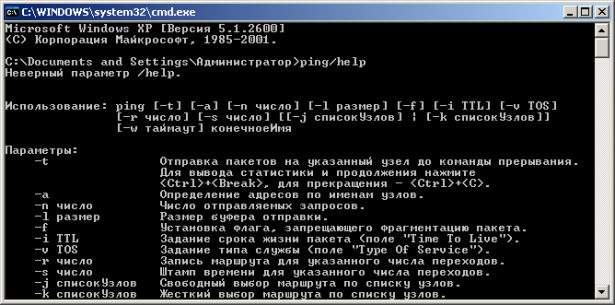
1. Многие команды Windows имеют большое количество дополнительных параметров и ключей, и запомнить их все невозможно или по крайней мере очень трудно. Большинство команд снабжено встроенной справкой, в которой кратко описываются назначение и синтаксис данной команды. Получить доступ к такой справке можно путем ввода команды с ключом **/?** или **/help**. В командной строке наберите

**copy /?**



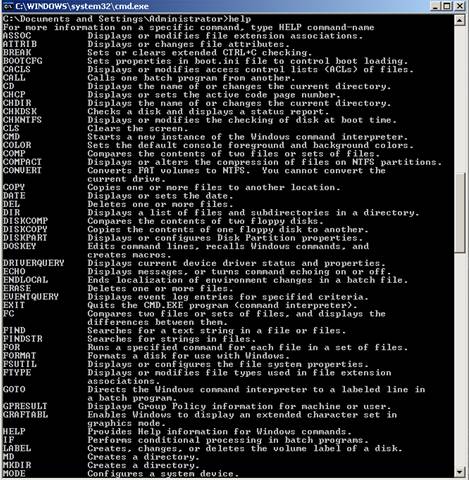
**Рис. 4.**

**ping /help**



**Рис. 5.**

Команда **help** выводит список основных команд Командной строки



**Рис. 6.**

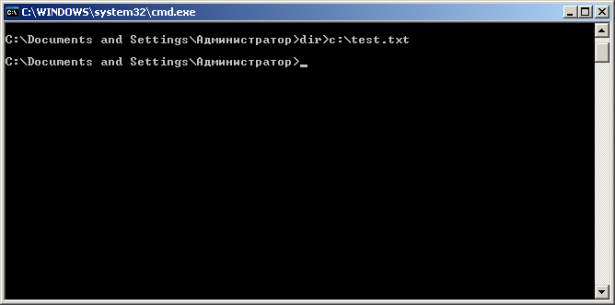
1. С помощью переназначения устройств ввода/вывода одна программа может направить свой вывод на вход другой или перехватить вывод другой программы, используя его в качестве своих входных данных. Таким образом, имеется возможность передавать информацию от процесса к процессу при минимальных программных издержках. Практически это означает, что для программ, которые используют стандартные входные и выходные устройства, операционная система позволяет: выводить сообщения программ не на экран, а в файл, читать входные данные не с клавиатуры , а из заранее подготовленного файла, передавать сообщения, выводимые одной программой, в качестве входных данных для другой программы.

Выходные данные практически всех команд высвечиваются в окне командной строки. Даже команды, выводящие данные на диск или принтер, выдают сообщения и запросы в окне командной строки.

Для перенаправления вывода команд из окна командной строки в файл или на устройство применяется оператор ">". Этот оператор используется с большинством команд. Например, для перенаправления вывода команды dir в файл test.txt введите в Командной строке:

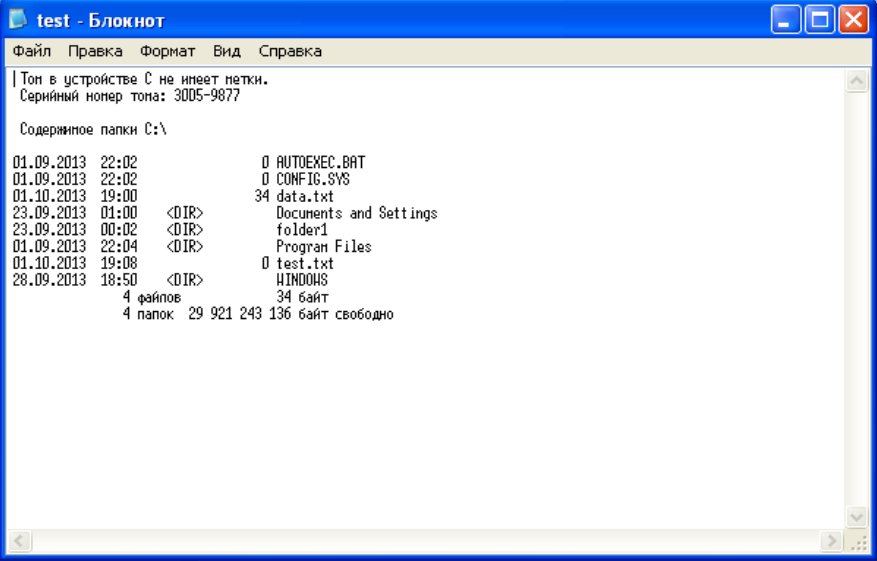
**dir>c:test.txt**

Если указанный файл не существует, интерпретатор команд Cmd.exe создаст его. Если файл существует, Cmd.exe заменит информацию в файле на данные, полученные от команды dir.



**Рис. 7.**

Откройте файл test.txt . В нем будет содержаться результат команды dir.



**Рис. 8.**

Следует отметить, что информация, которая была в файле до этого, будет стерта. Для того чтобы выходные файлы команды добавлялись в конец файла, необходимо использовать символ >>, а не > в синтаксисе команды.

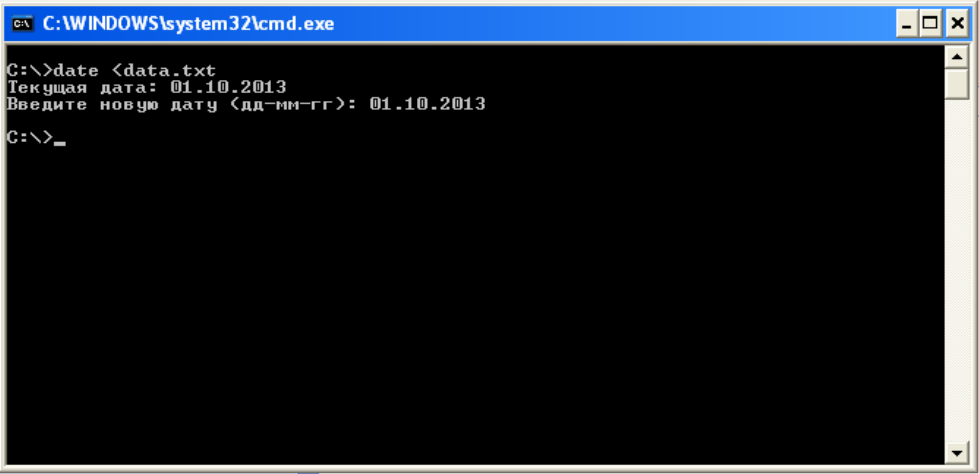
Местоположение потоков ввода и вывода называется дескриптор.

В таблице 1 описаны операторы перенаправления потоков ввода и вывода команд.

|  |  |
| --- | --- |
| Таблица 1. | |
| **Оператор перенаправления** | **Описание** |
| > | Записывает данные на выходе команды вместо командной строки в файл или на устройство, например, на принтер. |
| < | Читает поток входных данных команды из файла, а не с клавиатуры. |
| >> | Добавляет выходные данные команды в конец файла, не удаляя при этом существующей информации из файла. |
| >& | Считывает данные на выходе одного дескриптора как входные данные для другого дескриптора. |
| <& | Считывает входные данные одного дескриптора как выходные данные другого дескриптора. |
| | | Считывает выходные данные одной команды и записывает их на вход другой команды. Эта процедура известна под названием "канал". |

1. С помощью символа < можно прочитать входные данные для заданной команды не с клавиатуры, а из определенного (заранее подготовленного) файла. На диске С создайте файл data.txt и напишите в нем 01.10.2013. В Командной строке наберите

**Date < c:\data.txt**

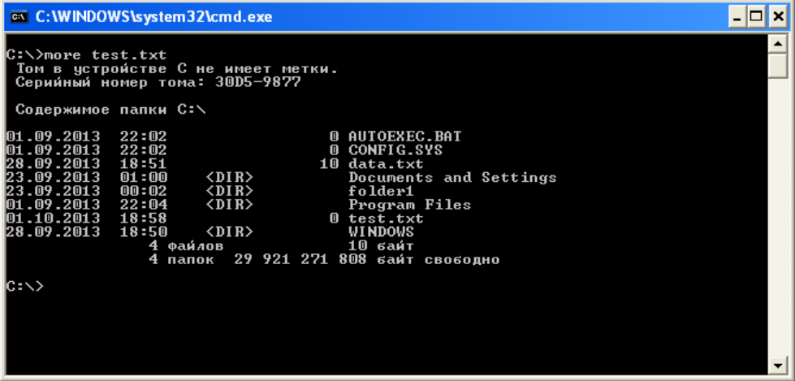


**Рис. 9.**

Проверьте дату на вашем компьютере – она изменилась на 01.10.2013

1. Команда **more** выводит содержимое файла или выхода команды в одном окне командной строки за раз. Например, чтобы отобразить содержимое файла test.txt в одном окне командной строки за раз, введите следующую команду:

**more c:\ test.txt**



**Рис. 10.**

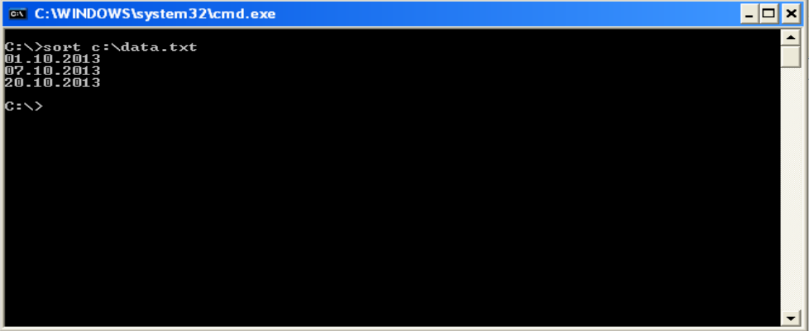
Если файл большой, то отображается одно окно командной строки со сведениями, а затем в нижней части окна командной строки отображается строка **-- More --**. Для остановки просмотра следует ввести комбинацию **CTRL+C**.

Команда **more** полезна при работе с командами, создающими выход более одного окна командной строки. Например, при выводе дерева каталогов жесткого диска.

1. Другой распространенной командой фильтрации является **sort** - она выполняет сортировку по алфавиту текстового файла или выхода команды.

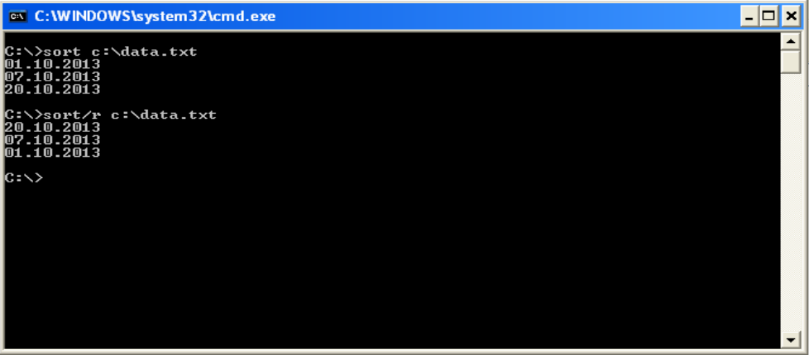
Отредактируйте файл data.txt, добавив туда две другие даты, например, 20.10.2013 и 07.10.2013.В командной строке наберите:

**sort < C:\data.txt**



**Рис. 11.**

**sort /r < C:\date.txt**



**Рис. 12.**

Ключ **/R** позволяет изменить порядок сортировки на обратный.

**Задание 2. Сетевая активность**

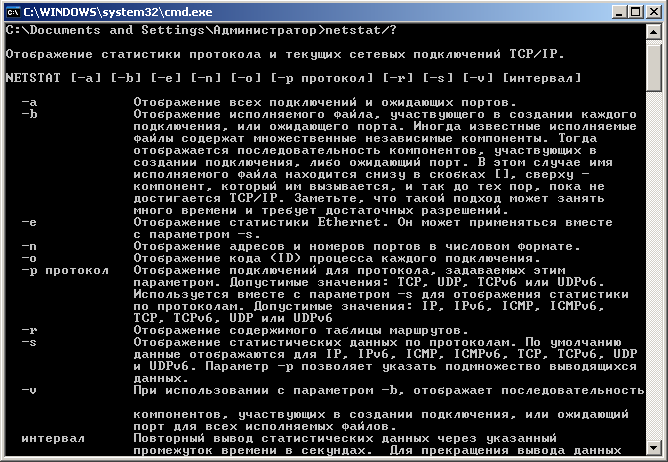
Одним из проявлений наличия вредоносной программы может быть возросшая сетевая активность. Вредоносная программа может отправлять письма, скачивать информацию из Интернета, передавать кому-то по сети конфиденциальную информацию и многое другое. При этом необходимо помнить, что легальные приложения также могут использовать Интернет без действий пользователя – например, антивирусная программа может скачивать обновления антивирусной базы данных.

Для получения полной информации о сетевой активности можно использовать команду **netstat**, которая выводит на экран мгновенную статистику сетевых соединений.

1. В Командной cтроке наберите:

**netstat /?**

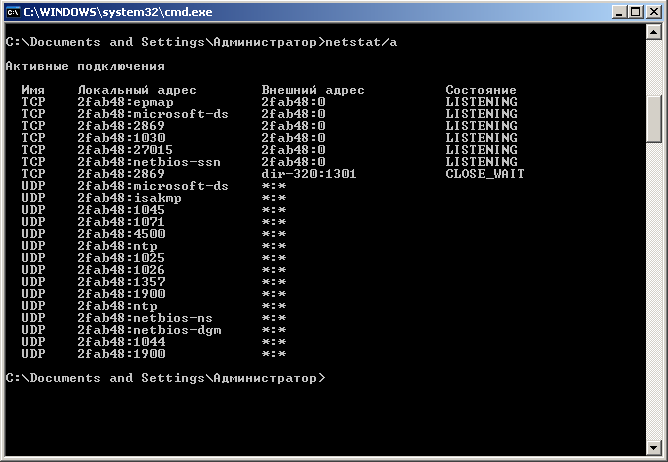
1. Прочитайте описание утилиты netstat. Убедитесь, что для вывода самой полной информации нужно использовать ключ **-a**



**Рис. 13.**

1. В Командной строке наберите:

**netstat /a**



**Рис. 14.**

Результатом выполнения команды является список активных подключений, в который входят установленные соединения и открытые порты.

**TCP(Transmission Control Protocol)-порты** обозначаются строкой "TCP" в колонке Имя. Открытые TCP-порты обозначаются строкой "LISTENING" в колонке состояние. Часть портов связана с системными службами Windows и отображается не по номеру, а по названию - epmap, microsoft-ds, netbios-ssn. Порты, не относящиеся к стандартным службам, отображаются по номерам.

**UDP(User Datagram Protocol)-порты** обозначаются строкой "UDP" в колонке Имя. Они не могут находиться в разных состояниях, поэтому специальная пометка "LISTENING" в их отношении не используется. Как и TCP-порты они могут отображаться по именам или по номерам.

Порты, используемые вредоносными программами, чаще всего являются нестандартными и поэтому отображаются согласно их номерам. Впрочем, могут встречаться троянские программы, использующие для маскировки стандартные для других приложений порты, например 80, 21, 443 - порты, используемые на файловых и веб-серверах.

Команда **netstat**, в отличие от Диспетчера задач Windows, не работает в режиме реального времени, а отображает мгновенную статистику. Следовательно, для просмотра активности соединений, скажем, через минуту, нужно заново выполнить команду.

**Задание 3. Командные файлы.**

Командные файлы принадлежат к кате­гории исполняемых файлов. Такие файлы снабжаются типом ВАТ (от английского слова batch — пачка), и в жаргонной речи их называют иногда "бэтч-файлами". Запуск командного файла осуществляется точно так же, как запуск программы в машинном коде (файла типа СОМ или ЕХЕ): указывается имя файла без расширения и, если требуется, пара­метры. Пример запуска командного файла START.BAT:

C:\start

Роль командных файлов особая. Они могут содержать целую группу команд ДОС или обращений к прикладным программам, ко­торые выполняются последовательно или в более сложном порядке, в зависимости от некоторых условий. Командный файл, в от­личие от исполняемых файлов остальных двух типов, содержит не машинный код программы, а текст, который интерпретируется команд­ным процессором. (Таким образом, по форме это обычный текстовый файл, содержимое которого можно вывести на принтер или эк­ран). Строки этого текста могут представлять собой:

— встроенные команды ДОС,

— обращения к исполняемым программам,

— вызовы других командных файлов,

— специальные команды для управления выдачей на экран,

— специальные команды для организации ветвлений и циклов,

— метки.

Все эти конструкции (за исключением меток) могут не только фи­гурировать в командных файлах, но и просто вводиться пользователем с клавиатуры. Однако в практической работе лишь команды первых трех типов вводятся пользователем, а остальные конструкции, вклю­чая метки, служат главным образом для создания нетривиальных ко­мандных файлов.

Рассмотрим несколько простых примеров применения командных файлов.

**Пример 1**.

Допустим, наша регулярная работа осуществляется в каталоге WORК, но время от времени мы хотим переходить в другой каталог — GRAPH, запускать там графический пакет PICASA, а по окончании работы с этой программой возвращаться в каталог WORK.

Последовательность команд при этом должна быть следующей:

cd \graph

picasa

cd \work

Эту последовательность можно разместить с помощью текстового редактора в командном файле. Каждая команда должна следовать на отдельной строке (регистр, на котором набирается текст, безразличен).

Дадим этому файлу имя V.BAT. Достаточно теперь задать в ка­честве одной команды имя этого файла — V, и содержащиеся в нем несколько команд начнут интерпретироваться одна за другой, избав­ляя нас от необходимости вводить их текст с клавиатуры. Первая команда сменит каталог, а вторая погрузит нас в операционную среду системы PICASA. Когда после выполнения необходимых действий мы, наконец, закончим работу с PICASA, управление вернется в командный файл. Выполнение третьей команды вернет нас в каталог WORK.

Заметим, что при выполнении каждой команды в данном примере на экране печатается приглашение, затем текст самой команды (эхо) и, наконец, результат ее работы. В следующем примере будет показано, как можно отключить выдачу приглашений и текстов команд, чтобы не загромождать экран лишней информацией.

**Пример 2**.

Допустим, мы хотим предельно сократить число манипуляций для создания необходимой рабочей обстановки при за­пуске системы. Мы можем заставить ДОС автоматически провести всю необходимую подготовку операционной среды с помощью командного файла следующего вида:

echo off

mode co80

path c:\;c:\exe;c:\turbo

cd \work

turbo

Дав этому файлу какое-либо простое запоминающееся имя, напри­мер, START. BAT, мы избавим себя от систематического набора ука­занных команд: достаточно набирать лишь слово START. Первая ко­манда этого файла ECHO OFF отключает "эхо", т. е. блокирует выдачу на экран приглашений ДОС и текстов последующих команд. Три сле­дующие команды осуществляют настройку необходимых параметров (установка режима дисплея, задание альтернативных маршрутов и рабочего каталога), а последняя команда производит запуск системы TURBO Pascal.

С помощью команды REM удобно блокировать исполнение некоторых команд, не удаляя их из текста командного файла.

Использование команд ECHO, PAUSE и REM полезно лишь в ко­мандных файлах; вводить их с клавиатуры для непосредственного ис­полнения не имеет особого смысла. Есть и другие команды, предназна­ченные главным образом для использования в командных файлах. К ним относятся:

GOTO — безусловный переход (на метку),

IF — проверка условия и ветвление,

FOR — управление повторным выполнением команд,

SHIFT — сдвиг списка формальных параметров.

Эти команды будут рассмотрены в следующем разделе.

Отметим еще одно важное свойство командных файлов — возмож­ность использования внутри них *формальных параметров.*

Для пояснения этой возможности обратимся еще раз к первому примеру. Допустим, нам хотелось бы применять одну и ту же последо­вательность команд, приведенную в примере 1, для запуска разных подсистем. Тогда можно видоизменить приведенный выше текст файла V. ВАТ, Заменив имя каталога GRAPH и имя вызываемой программы PICASA формальными параметрами. Командный файл при этом при­обретает вид:

cd \% 1

%2

cd \work

Символы %1 и %2 обозначают формальные параметры, вместо ко­торых при обращении к V.BAT будут подставлены в текстовом виде фактические параметры, указанные в командной строке. Обращение к V.BAT может иметь вид:

v graph picasa

Строки graph и picasa займут место соответственно параметров % 1 и %2, в результате чего получится точно такой же текст, как и в при­мере 1. Однако новый командный файл, в отличие от прежнего, можно использовать для запуска других подсистем, например:

v ss symphony или v ww word

Эффект таких обращений очевиден. В первом случае из каталога SS будет вызвана система SYMPHONY, во втором случае из ката­лога WW вызывается система WORD.

***Развитые командные файлы.***

Рассмотрим действие специальных команд, позволяющих управлять интерпретацией командных файлов, К ним, как указывалось, отно­сятся команды GOTO, IF, FOR, SHIFT, а также команда EXIT, иг­рающая особую роль при рекурсивном (вложенном) вызове командных файлов.

**Команда GOTO**.

Позволяет передавать управление на метку и тем самым осуществлять повторное исполнение участков командного файла или, наоборот, обходить некоторые участки (что обычно имеет смысл в сочетании с командой ветвления IF).

**Пример 1.**

Пусть командный файл С.ВАТ имеет вид;

echo off

:ml

echo Вывод на принтер файла % 1

echo Для остановки нажмите CtrlC

сору % 1 рrn

pause

goto ml

В данном примере вторая строка содержит метку ml (признаком метки является двоеточие в начале строчки), а последняя строка — команду перехода на эту метку. Четыре команды внутри повторяемого участка командного файла служат для выдачи на экран поясняющих сообщений (команды ECHO) и копирования на принтер (т. е. печати) файла, имя которого подставляется вместо формального параметра % 1.

Команда PAUSE приостанавливает исполнение, чтобы пользователь имел возможность осмотреться и либо прервать работу, либо продол­жить ее дальше.

Допустим, пользователь дает с терминала команду:

с spectr. doc

Тогда начинается циклическое выполнение файла С. ВАТ с фак­тическим параметром SPECTR.DOC. На принтер начнут выдаваться одна за другой копии этого текстового файла, прерываемые паузами после каждой очередной копии. Пользователь может прервать выполнение этого командного файла, нажав одновременно клавиши Ctrl и С.

**Команда IF.**

Позволяет проверять *условие* и выполнять команду в зависимости от результата его проверки. Что можно задавать в ка­честве условия? Для этого имеется три возможности:

1. Проверка *кода завершения* какой-либо программы, сработав­шей перед оператором IF. При этом в начале оператора IF исполь­зуется конструкция следующего вида:

IF ERRORLEVEL <N>

Любая программа может с помощью специального прерывания ДОС выработать в момент своего окончания - так называемый код за­вершения. Этот код сравнивается с числом N. Условие считается вы­полненным *(истинным),* если выработанный код завершения равен или больше указанного числа N.

2. Проверка наличия файла в каталоге. В этом случае начало опе­ратора IF имеет вид:

IF EXIST (имя файла или шаблон)

В данном случае команда IF проверяет, действительно ли сущест­вует файл с заданным именем в указанном или в текущем каталоге. При обнаружении файла условие считается выполненным.

3. Сравнение двух строк, которые, в частности, могут задаваться через формальные параметры. Соответствующая конструкция может иметь вид:

IF%(N)== (текстовая строка)

Здесь значение формального параметра % (N) сопоставляется с кон­кретной строкой. При абсолютном совпадении двух строк условие считается выполненным.

Любое из этих условий может задаваться со знаком логического отрицания NOT; отрицание условия считается выполненным, если само условие не выполнено (ложно).

**Пример 2.**

Опишем прием, который часто используется для организации ветвления по коду, введенному пользователем с терминала (настройка на пользователя). Будем предполагать наличие специаль­ной программы (назовем ее ASK-CODE), которая выполняет следующие операции:

— выдает на экран текст сообщения,

— принимает введенное пользователем с клавиатуры число,

* вырабатывает код завершения, совпадающий с этим числом.

Фрагмент командного файла, использующего программу ASK-CODE, может иметь следующий вид:

ASK-CODE "Ваш код?"

if errorlevel *5* goto USERS

if errorlevel 4 goto USER4

if errorlevel 1 goto USER I

cd \work

goto end

:USERS cd \user5

goto end

:USER4 cd \user4

goto end

:USER1 cd \userl

:end

В данном примере группа команд IF ERRORLEVEL ... осуществ­ляет анализ кода завершения, выработанного программой ASK-CODE (равного, напомним, числу, введенному пользователем с терминала в ответ на вопрос "Ваш код?"). В зависимости от значения происходит переход на одну из меток USERS, USER4, ... USER1, где затем происходит настройка операционной среды на конкретного пользова­теля. Если введенный пользователем код превышает 5*,* то переход все равно происходит на метку USER5; если же код равен 0, происходит переход на следующую команду под указанной группой операторов (cd \work). Таким образом, осуществляется ветвление внутри командного файла.

**Пример 3.**

Иногда в начале работы пользователю необходимо узнать, не оставил ли ему кто-либо из коллег какое-нибудь сообщение. Допустим, имеется договоренность, что такие сообщения заносятся в файл MAIL.(N) в каталоге USER(N), где (N) — код пользователя-адресата. Тогда проверка наличия сообщения и его печать для пользо­вателя с кодом 1 может осуществляться командой вида:

if exist MAIL.1 type MAIL.1

Такая команда может стоять внутри участка командного файла, рассмотренного выше, после строки cd \userl. Аналогичные команды могут находиться в других участках этого командного файла после меток USERS, USER4 ...

**Пример 4.**

Допустим, в общедоступном каталоге имеется про­грамма NOTES.EXE, реализующая работу с картотекой/записной книжкой. Такую программу пользователь должен запускать с разными параметрами — именами картотек, которые находятся в разных ката­логах:

— картотека TELEF в каталоге TEL,

— картотека PAPERS в каталоге PAPERS,

— картотека PERSON в каталоге ADMIN.

При обычном способе запуска пользователю пришлось бы набирать каждый раз по две команды с длинными именами, например:

cd \tel

notes telef

Можно создать в общедоступном каталоге командный файл, ко­торый позволит осуществлять необходимые вызовы с использованием простых однобуквенных мнемонических имен. Такой командный файл с именем N.BAT может иметь вид (выделение меток большими буквами здесь сделано лишь для наглядности):

echo off

if %1= goto exit

if %1= t goto TEL

if %1= p goto PAP

if %1= s goto PERS

goto exit

:TEL

cd \tel

notes telef

goto exit

:PAP

cd \papers

notes papers

goto exit

:PERS

cd \admln

notes person

:ехit

Обращение к указанному файлу при вызове разных картотек должно иметь вид:

n= t или n= р или n= s

При этом буквы t, p, s должны быть малыми, поскольку именно в таком виде они заданы в условиях операторов IF.

Отсутствие букв t, p или s-при вызове этого файла приведет к переходу на метку exit.

**Команда FOR.**

Обеспечивает циклическое выполнение команд ДОС. При этом можно задать формальный параметр и список фактических параметров (обычно—имен файлов), которые последовательно подставляются вместо формального параметра в текст исполняемой ко­манды.

**Пример 5.**

Пусть имеется необходимость систематически ко­пировать файлы PROQ.PAS, PROG.OBJ и PROG.EXE из рабочего каталога на виртуальный диск D:. С этой целью можно в соответствую­щем командном файле дать команду:

for %%A in (PAS OBJ EXE) do copy PROG.%%A d:

Здесь формальный параметр %%А последовательно сопоставляется со списком фактических параметров в круглых скобках и использу­ется в команде COPY, которая в данном случае сработает 3 раза подряд.

Подставив в качестве второго аргумента команды COPY вместо имени D: еще один формальный параметр %1, можно будет менять назначение копирования, задавая его с терминала.

**Команда SHIFT.**

Вызывает сдвиг списка формальных параметров относительно списка фактических параметров. Так, если в командном файле фигурируют формальные параметры %1 и %2, а в обращении к командному файлу — фактические параметры А, В, С, D, то сначала соответствие формальных и фактических параметров выглядит следую­щим образом:

%1=А %2=В Однократное применения команды SHIFT дает следующее соответствие:

%1=В %2=С Двукратное применение вызывает дальнейший сдвиг:

%1=С %2=D

**Пример 6.**

Пусть стоит задача, обратная той, которая рас­смотрена в примере 5; а именно, — нужно копировать на виртуальный диск D: файлы, имена которых пользователь будет вводить с терми­нала Эту задачу позволит решить командный файл следующего вида;

echo off

:loop

сору % 1 d:

shift

goto loop

Если дать этому файлу имя D.BAT и вызвать его с произвольным числом аргументов:

d f.l f.2 f.3 f.4 f.5 то будет происходить последовательное присваивание формальному параметру % 1 значений F.I, F.2, F.3, F.4, F.5 и циклическое исполнение команд COPY и SHIFT. После исчерпания фактических параметров команда COPY выдаст сообщение об ошибке, поскольку ее первый ар­гумент будет отсутствовать. В этот момент пользователь сможет пре­рвать работу командного файла, нажав клавиши Ctrl и С.

***Одним из методов защиты информации является работа с файлами, оптимизация которой может достигаться созданием пакетных файлов в среде Windows.***

**Задание 4. Создание пакетного файла, реализующего определенную последовательность действий в ОС Windows XP**

Порядок выполнения:   
I. Создайте новый пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберете самостоятельно.   
II. Изучите выбранный вариант задания табл. 1.   
III. Синтезируйте алгоритм работы пакетного файла.   
IV. Выберите необходимый набор команд для реализации алгоритма.   
V. С помощью выбранного набора команд запрограммируйте сценарий в виде пакетного файла, реализующего определенную последовательность действий в среде ОС Windows XP.   
VI. Сохраните текст пакетного файла.   
При выполнении задания используйте следующие инструкции:

* воспользовавшись командой Start и указав путь к пакетному файлу, запустите его на выполнение, нажав Enter для ввода,
* изучите полученный результат,

Дополнительную информацию по возможностям командной оболочки, а также все множество команд доступных при работе с ней наряду с параметрами и примерами применения можно получить в справке ОС Windows XP (Пуск | Справка и поддержка) в разделах «Общие сведения о командной оболочке», «Справочник по параметрам командной строки» и «Новые средства командной строки».

**Варианты для задания.**

**Таблица 1.** Варианты для задания .

|  |  |
| --- | --- |
| **Задание 4** | |
| **Вар. №** | **Описание пакетного файла** |
| **1** | Пакетный файл, предназначенный для резервного копирования файлов с определенным расширением из разных каталогов с возможностью создания резервного каталога, в случае его отсутствия в системе. Расширение файлов для копирования задается в качестве пакетного параметра. Резервное копирование осуществляется каждый четверг в 22:00. В течение 3 минут после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическое выключение системы с принудительным закрытием всех работающих приложений. |
| **2** | Пакетный файл, предназначенный для организации процесса поиска и отображения текстового файла. Поиск осуществляется по всем локальным дискам. Имя текстового файла задается пакетным параметром. После того как необходимый файл найден, в автоматическом режиме осуществляется его отображение в текстовом процессоре «Блокнот». |
| **3** | Пакетный файл, предназначенный для копирования каталога с его содержимым в заданное место назначения. Копируемый каталог и место назначения задаются в качестве пакетных параметров. После копирования каталога файл-отчет, содержащий информацию о количестве скопированных файлов и их месте расположения, в автоматическом режиме загружается в текстовый процессор «Блокнот». |
| **4** | Пакетный файл, предназначенный для перемещения каталога с его содержимым в заданное место назначения с запросом на удаление, перемещаемого каталога. Перемещаемый каталог и место назначения задаются в качестве пакетных параметров. После перемещения каталога в отдельный файл выводится отчет, содержащий два дерева каталогов тех мест, откуда и куда было осуществлено перемещение. В конце выводится сообщение вида «Отчет о перемещении находится в каталоге <путь>». |
| **5** | Пакетный файл, предназначенный для копирования каталога и включенных в него файлов, расположенных в месте, заданном определенным путем. Полный путь расположения и маска копируемых файлов задаются в качестве пакетных параметров. Если в результирующем каталоге уже находятся копирующиеся файлы, то повторное копирование должно сопровождаться выдачей предупреждающего сообщения о существовании файлов. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено». |
| **6** | Пакетный файл, предназначенный для создания отчета, содержащего «Software part» (программная часть), включающую информацию о содержимом корневых каталогов всех логических дисков в системе и «Hardware part» (аппаратная часть), включающую сведения о конфигурации компьютера и ОС, сведения о безопасности, параметры оборудования, такие как ОЗУ, дисковое пространство, сетевые карты и другие. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим имя компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>». |
| **7** | Пакетный файл, предназначенный для углубленной проверки жесткого диска с созданием файла отчета, путь к которому задается в качестве пакетного параметра. Проверка жесткого диска осуществляется ежедневно в 21:00. В течение 20 секунд по окончании проверки диска выводится сообщение «Проверка диска завершена. Файл-отчет находится в каталоге <путь >» и далее осуществляться автоматическая перезагрузка системы. |
| **8** | Пакетный файл, предназначенный для резервного копирования файлов системной папки Windows с возможностью создания резервного каталога, в случае его отсутствия в системе. Путь к резервному каталогу задается в качестве пакетного параметра. Резервное копирование осуществляется ежедневно в 23:00. В течение 2 минут после копирования выводится «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическое выключение системы с принудительным закрытием всех работающих приложений. |
| **9** | Пакетный файл, предназначенный для архивирования и шифрования указанного каталога с его содержимым. Архивируется каталог-источник с помощью существующего в системе архиватора (например, WinRar), вызов которого осуществляется непосредственно из пакетного файла. Архив в дальнейшем шифруется и сохраняется в определенном месте на жестком диске. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Шифрованный архив сохранен в каталог <путь>». |
| **10** | Пакетный файл, предназначенный для удаления файлов по маске, расположенных в месте, заданном определенным путем. Полный путь расположения и маска удаляемых файлов задаются в качестве пакетных параметров. В процессе необходимо осуществлять запрос на подтверждение удаления. В конце выводится сообщение вида «Стерто файлов: <количество> из каталога <путь>». |
| **11** | Пакетный файл, предназначенный для организации процесса поиска и сравнения оригинального и резервной копии (.bak) одного и того же файла. Если оригинальный файл найден, то осуществляется его сравнение с резервной копией. Отличия, найденные при сравнении, передаются в отчет, который сохраняется в определенном месте. Имя резервной копии файла и путь к месту назначения, где сохраняется отчет о сравнении, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Отчет сохранен в каталог <путь>». |
| **12** | Пакетный файл, предназначенный для создания отчета, содержащего «Software part» (программная часть), включающую информацию об присутствующих в системе загруженных драйверах и «Hardware part» (аппаратная часть), включающую сведения о конфигурации компьютера и ОС, сведения о безопасности, параметры оборудования, такие как ОЗУ, дисковое пространство и другие. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим IP-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в каталоге <путь>». |
| **13** | Пакетный файл, предназначенный для подсчета файлов в каталоге, заданном определенным путем. Полный путь расположения и расширение подсчитываемых файлов задаются в качестве пакетных параметров. Организовать отчет с возможностью дописывания в него информации вида «Каталог <путь> содержит <количество> файлов с <расширение> расширением» |
| **14** | Пакетный файл, предназначенный для резервного копирования каталога, заданного определенным путем, и содержащихся в нем файлов с возможностью создания резервного каталога, в случае его отсутствия в системе. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. Резервное копирование осуществляется ежедневно в 23:59. В течение 5 секунд после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическая перезагрузка системы. |
| **15** | Пакетный файл, предназначенный для отражения статистики по атрибутам файлов в каталоге, заданном определенным путем. В каталоге файлы с определенным атрибутом подсчитываются, а их количество передается в текстовый файл статистики с дописыванием в него информации вида «Файлов с атрибутом <атрибут>: <количество>». Пути к каталогу и месту назначения, где сохраняется файл статистики, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Отчет сохранен в каталог <путь>». |
| **16** | Пакетный файл, предназначенный для копирования файлов с определенным расширением и путем в заданное место назначения с их последующим шифрованием. Расширение файлов для копирования, полный путь расположения и путь к месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено. Шифрование скопированных данных завершено». |
| **17** | Пакетный файл, предназначенный для организации процесса поиска и сравнения оригинального и резервной копии (.bak) одного и того же файла. Имя резервной копии файла передается в качестве пакетного параметра. Если оригинальный файл найден, то осуществляется его сравнение с резервной копией. Отличия, найденные при сравнении, передаются в отчет, который, в автоматическом режиме загружается в текстовый процессор «Блокнот». |
| **18** | Пакетный файл, предназначенный для создания отчета, включающего информацию о присутствующих в системе загруженных драйверах и отображающего список приложений и служб, выполняющихся на компьютере. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим MAC-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>». |
| **19** | Пакетный файл, предназначенный для копирования файлов, определяемых маской и путем, в заданное место назначения с их последующим архивированием. Архивирование осуществляется с помощью доступного в системе архиватора (например, WinRar), вызов которого осуществляется непосредственно из пакетного файла. Маска файлов для копирования, полный путь расположения и путь к месту назначения задаются в качестве пакетных параметров. В конце выводится сообщение вида «Копирование файлов из каталога <путь> в каталог <путь> завершено. Архивирование скопированных данных завершено». |
| **20** | Пакетный файл, предназначенный для удаления файлов с определенным расширением, расположенных в месте, заданном определенным путем. Полный путь расположения и расширение удаляемых файлов задаются в качестве пакетных параметров. В процессе необходимо осуществлять запрос на подтверждение удаления. После удаления в отдельный файл выводится отчет, содержащий список удаленных файлов с их полным путем, который, в свою очередь, в автоматическом режиме загружается в текстовый процессор «Блокнот». |
| **21** | Пакетный файл, предназначенный для создания отчета событий и их свойств из журнала событий на локальном компьютере. Журналы событий: Application, System и Security. Файл-отчет копируется в некоторый сетевой каталог, задаваемый пакетным параметром, под именем, отражающим IP-адрес компьютера, с которого получен этот отчет. В конце выводится сообщение вида «Отчет находится в сетевом каталоге <путь>». Примечание: если необходимо, смените сервер сценариев ОС Windows XP. |
| **22** | Пакетный файл, предназначенный для перемещения файлов с определенным расширением из каталога-источника в заданное место назначения. Расширение файлов, каталог-источник и место назначения задаются в качестве пакетных параметров. После перемещения отчет, содержащий список перемещенных файлов с путем, загружается в текстовый процессор «Блокнот». |
| **23** | Пакетный файл, предназначенный для резервного копирования каталога, заданного определенным путем, и содержащихся в нем файлов с возможностью создания резервного каталога, в случае его отсутствия в системе. Пути к каталогу-источнику и месту назначения задаются в качестве пакетных параметров. Резервное копирование осуществляется ежедневно в 23:30. В течение 10 секунд после копирования выводится сообщение «Резервное копирование в каталог <путь> завершено» и далее происходит автоматическая перезагрузка системы. |
| **24** | Пакетный файл, предназначенный для копирования дерева каталогов из каталога-источника в заданное место назначения. Копируемый каталог-источник и место назначения задаются в качестве пакетных параметров. После копирования отчет, содержащий дерево каталогов с местом его расположения, в автоматическом режиме загружается в текстовый процессор «Блокнот». |
| **25** | Пакетный файл, предназначенный для организации процесса поиска и копирования файлов с определенным расширением. Поиск осуществляется по всем локальным дискам. Расширение файлов и место назначения, куда необходимо копировать файлы, задаются в качестве пакетных параметров. В конце выводится сообщение вида «Файлы найдены и скопированы в каталог <путь>». |
| **26** | Пакетный файл, предназначенный для копирования системных и скрытых файлов из каталога-источника в заданное место назначения. Каталог-источник и место назначения задаются в качестве пакетных параметров. После копирования отчет, содержащий список скопированных файлов с путем, в автоматическом режиме загружается в текстовый процессор «Блокнот». |