

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего профессионального образования
«Томский государственный университет систем управления и
радиоэлектроники»

Кафедра электронных приборов

Локальные компьютерные сети

ИСПОЛЬЗОВАНИЕ КОМАНДНЫХ ФАЙЛОВ

Методические указания к лабораторной работе
для студентов направлений «Электроника и микроэлектроника»
(специальность «Электронные приборы и устройства»)

Агеев Евгений Юрьевич

Использование командных файлов = Локальные компьютерные сети: Методические указания к лабораторной работе для студентов направлений «Электроника и микроэлектроника» (специальность «Электронные приборы и устройства» / Е.Ю. Агеев; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Томский государственный университет систем управления и радиоэлектроники, Кафедра электронных приборов. - Томск : ТУСУР, 2012. – 14 с.

Настройкой сетевых подключений обычно занимается администратор локальной компьютерной сети. При этом часто возникают задачи, требующие выполнения рутинных, повторяющихся операций. Для автоматизации выполнения таких задач в операционной системе Windows достаточно широко используются командные BAT-файлы. Такие файлы иногда называют файлами сценариев (англ. script) или просто «скриптами».

Предназначено для студентов очной и заочной форм, обучающихся по направлению «Электроника и микроэлектроника» (специальность «Электронные приборы и устройства») по курсу «Локальные компьютерные сети»

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Томский государственный университет систем управления и
радиоэлектроники»

Кафедра электронных приборов

УТВЕРЖДАЮ

Зав.кафедрой ЭП

_____ С.М. Шандаров

« ___ » _____ 2012 г.

Локальные компьютерные сети

ИСПОЛЬЗОВАНИЕ КОМАНДНЫХ ФАЙЛОВ

Методические указания к лабораторной работе
для студентов направлений «Электроника и микроэлектроника»
(специальность «Электронные приборы и устройства»)

Разработчик

_____ Е.Ю. Агеев

_____ 2012 г

СОДЕРЖАНИЕ

1 Введение.....	5
2 Теоретическая часть.....	5
2.1 Команды ВАТ-файлов	5
2.2 Контрольные вопросы	11
3 Экспериментальная часть.....	12
3.1 Задание на работу.....	12
3.2 Содержание отчета	13
Рекомендуемая литература	13

1 Введение

Настройкой сетевых подключений обычно занимается администратор локальной компьютерной сети. При этом часто возникают задачи, требующие выполнения рутинных, повторяющихся операций. Для автоматизации выполнения таких задач в операционной системе Windows достаточно широко используются командные BAT-файлы. Такие файлы иногда называют файлами сценариев (англ. script) или просто «скриптами». Современные версии Windows поддерживают и более развитые средства управления различными (в том числе и сетевыми) настройками операционной системы, такие как WSH (Windows Scripting Host - сервер сценариев Windows), служба каталогов Active Directory, обеспечивающая администраторов интуитивным иерархическим представлением сети и единым инструментом администрирования всех сетевых объектов, наконец, инструментарий управления Windows - WMI (Windows Management Instrumentation), но, несмотря на это, командные файлы, благодаря достаточной простоте и легкости создания, по-прежнему применяются администраторами локальных сетей.

2 Теоретическая часть

2.1 Команды BAT-файлов

Перенаправление потоков ввода/вывода

Перенаправление потоков ввода/вывода - одна из самых полезных возможностей используемых в командных файлах. Такое перенаправление позволяет вывести результат работы команды или программы не на экран, а, например, в файл или на принтер. Для перенаправления потоков ввода/вывода используются символы ">", ">>", "<", "<<" и "|". С помощью символов перенаправления вывода результат работы одной команды можно передать на вход другой команды.

Например, команда

```
DIR *.* >abc.txt
```

выведет результаты не на экран, а в файл **abc.txt**, предварительно создав или перезаписав его (если такой файл уже существовал). Команда.

```
DIR *.* >>abc.txt
```

также запишет результат в файл **abc.txt**, но она не удалит его содержимое, а добавит свой вывод в конец файла. Если такой файл не существует, он будет создан. Следующий пример показывает применение фильтра **MORE** совместно с командой **DIR**:

```
DIR *.* | MORE
```

В этом случае после заполнения экрана выводом команды **DIR** вывод будет останавливаться до нажатия любой клавиши.

Пример использования команды ECHO, совместно с командой DIR:

ECHO Y | DEL *.*

Сама команда `DEL *.*` должна удалить все файлы в текущем каталоге. Однако, она запрашивает подтверждение на подобное действие. Для того, чтобы использовать команду в составе пакетного файла, мы используем передачу результатов выполнения команды `ECHO Y`, которая должна вывести символ `Y`, в команду `DEL *.*`, которой требуется один символ: `Y` или `N`. Похожее действие выполнит команда

```
program.exe <file.in
```

которая передаст программе **program.exe** данные из файла **file.in** таким образом, как будто они были введены с клавиатуры.

При перенаправлении ввода/вывода полезно использовать специальные команды-фильтры. Фильтры пропускают только часть получаемой информации, соответствующей запросу. Это команды **FIND** и **SORT**.

```
FIND [/V] [/C] [/N] [/I] "строка" [[диск:][путь]
        имя_файла[...]]
```

Поиск текстовой строки в одном или нескольких файлах.

Параметры команды:

`/V` - Вывод всех строк, НЕ содержащих заданную строку.

`/C` - Вывод только общего числа строк, содержащих заданную строку.

`/N` - Снабжение отображаемых строк номерами.

`/I` - Поиск без учета регистра символов.

"строка" - Искомая текстовая строка.

`[диск:][путь] имя_файла` - файл либо файлы, в которых производится поиск. Если путь не задан, **FIND** производит поиск в тексте, введенном с клавиатуры, либо переданном по конвейеру другой командой.

Пример: `DIR A: | FIND /V "компьютер"`

Выведет имена всех файлов, не содержащих слово "компьютер".

Примечание: Можно использовать параметр `/V` вместе с `/C` или `/N`, но `/N` не будет работать с `/C`.

Команда сортировки **SORT**, вариант записи для сортировки информации в файле:

```
SORT [/R] [/+п] [[диск1:][путь1]имя_файла1] [> [диск2:]
[путь2]имя_файла2], вариант для обработки вывода команды: [команда |]
SORT [/R] [/+п] [> [диск2:][путь2]имя_файла2]
```

Сортировка ввода с выводом результатов в файл, на экран или другое устройство

Параметры:

`/R` - Изменение порядка сортировки на обратный, от **Z** к **A** и затем от 9 до 0.

`/+п` - Сортировка файла по символам, начиная с `п`-го символа.

`[диск1:][путь1] имя_файла1` - Сортируемые файлы

`[диск2:][путь2] имя_файла2` - Файл, в который направляется

сортированный ввод.

команда - Команда, вывод которой следует отсортировать.

Пример:

```
SORT /R < BOOK1 > REV
```

Сортирует файл **BOOK1** в обратном порядке и записывает результат в файл **REV**.

Специальные команды

REM	Команда REM служит для ввода комментариев в командный файл. Любой текст, расположенный в строке после нее, считается комментарием и не выполняется.
ECHO	Команда ECHO используется для управления эхо-режимом, то есть режимом отображения на экране служебной информации (текста программы) во время ее выполнения. ECHO OFF запрещает этот вывод, а ECHO ON – разрешает. Команда ECHO <Текст> используется для вывода текста на устройство (в частности, на экран). Например, команда: ECHO Привет! выведет на экран слово Привет! .
@	Это аналог оператора ECHO OFF только для одной строки. Часто используется следующим образом в начале многих пакетных файлов: @ECHO OFF , чтобы скрыть вывод на экран самой команды ECHO OFF при выполнении пакетных файлов.
GOTO <метка >	Оператор GOTO осуществляет безусловный переход на метку внутри командного файла. Метка обозначается знаком двоеточия в начале строки – : label (в самом операторе указывается только имя метки без двоеточия). Метка может включать пробелы, но не может включать другие разделители, такие как точка с запятой или знак равенства. В команде GOTO используются только первые восемь знаков метки. Примечание: символ двоеточия может также служить комментарием.
%	Указание на переменную (формальный параметр). Всего может быть до 10 независимых переменных. При использовании формальных параметров переменная %0 заменяется на имя пакетного файла, а

	<p>переменные с %1 по %9 – на соответствующие аргументы, напечатанные в командной строке. Например, если в командной строке записать: <code>abc.bat a bc def</code>, то переменная %1 будет содержать значение a, %2 будет содержать bc, а %3 – def. Задание переменных широко используется для создания универсальных пакетных файлов при работе с повторяющимися операциями. Запись в файле ABC.BAT обратного порядка параметров: <code>ABC.BAT %2 %1</code> вызовет прекращение выполнения файла, запущенного с набором параметров и автоматический повторный запуск этого же файла с параметрами, записанными в обратном порядке. Эта манипуляция иногда используется для исправления неправильного порядка параметров, переданных пакетному файлу.</p>
SHIFT	<p>Команда SHIFT сдвигает список параметров на один влево. То есть, переменная %1 будет содержать значение переменной %2 до сдвига. Использование команды SHIFT позволяет задавать более 10 параметров для командного файла (Не «внутри» файла, а в командной строке запуска). Применение команды с ключом: SHIFT /n исключает из списка сдвигаемых первые n переменных, если n=2, то не изменятся переменные %0 и %1. Значение n не может быть больше 8.</p>
CALL <имя bat- файла>	<p>Команда CALL позволяет вызвать один пакетный файл из другого, не прекращая выполнения предыдущего, то есть происходит действие, аналогичное вызову подпрограммы. Этим способом можно организовать рекурсивное выполнение пакетных файлов. Примечание: вызвать подпрограмму с последующим возвратом можно и без использования команды CALL. Для этого используется команда: <code>cmd.com /c abc.bat</code>.</p>
IF	<p>IF <условие> <следствие>. Оператор IF проверяет истинность какого-либо <условия> и в зависимости от этого выполняет или не выполняет команду <следствие>. Возможны следующие условные</p>

	<p>конструкции:</p> <pre>IF EXIST ABC.BAT GOTO work</pre> <p>если существует файл ABC.BAT, то перейти на метку work.</p> <pre>IF NOT EXIST ABC.BAT GOTO exit</pre> <p>если не существует файл ABC.BAT, то перейти на метку exit.</p> <pre>IF %1= =ASDF GOTO label</pre> <p>если переменная %1 равна ASDF, то – переход на метку label.</p> <pre>IF ERRORLEVEL 1 GOTO err1</pre> <p>если ошибка (возвращаемое значение) выполнения предыдущей команды равно 1, то перейти на метку err1.</p>
PAUSE	<p>Команда PAUSE приостанавливает выполнение пакетного файла и выдает на экран приглашение нажать на любую клавишу для продолжения работы.</p>
FOR	<p>Обеспечивает циклическое выполнение команд. При этом можно использовать как формальные, так и фактические параметры. Например:</p> <pre>FOR %%P in (PAS OBJ EXE) DO copy PROG.%%P d:</pre> <p>Здесь формальный параметр %%P последовательно сопоставляется со значениями переменных в списке фактических параметров, заключенных в скобки. После каждого сопоставления выполняется команда copy. Знак процента используется в операторе FOR удвоенным для обозначения пустой переменной в пакетном файле. В качестве команды можно указать и вызов какой-либо подпрограммы. Замечание: В строке цикла FOR нельзя использовать переадресацию и перенаправление потоков.</p>
SET	<p>Команда SET задает переменную, значение которой подставляется вместо ее имени при любом использовании этого имени между знаками процента. Так, если задано: SET B=220, то при использовании в пакетном файле следующей конструкции: ECHO %B% на</p>

экран будет выведено "220". Переменные, определенные с помощью оператора **SET** называются переменными окружения среды (environment) и являются видимыми после выполнения команды до перезапуска операционной системы. Другими словами, такую переменную можно использовать из одного пакетного файла или программы после задания в другом.

Особенности использования возвращаемых значений

Любая программа при завершении своей работы возвращает операционной системе код своего завершения. При успешном завершении возвращается ноль, иначе - код ошибки. Для получения этого значения предусмотрена ключевая переменная **ERRORLEVEL**.

Пример пакетного файла:

```

OESHO OFF
REM Запускаем программу prg1.exe
PRG1.EXE
REM Анализ кода завершения
IF ERRORLEVEL 2 GOTO FILE_NOT_FOUND
IF ERRORLEVEL 1 GOTO WRITE_ERROR
IF ERRORLEVEL 0 GOTO EXIT_OK
GOTO ON_EXIT
:FILE_NOT_FOUND
ECHO Ошибка! Файл не найден!
GOTO ON_EXIT
:WRITE_ERROR
ECHO Ошибка записи!
GOTO ON_EXIT
:EXIT_OK
ECHO Программа завершена благополучно.
:ONEXIT

```

Анализ кода завершения начинается не с нуля, а с максимально возможного значения. Так делается потому, что подобная проверка означает: «если **errorlevel** *больше или равен* значению, то...». То есть, если проверка начнется с нуля, то любое значение будет истинным на первой же строке.

Передача параметров запускаемой программе и возврат значений из нее позволяет организовать двустороннюю связь между вызывающей и вызываемой программой.

Пример командного файла с формальными переменными, использованием команд SET, SHIFT, IF, GOTO.

```

@echo off
rem MYCOPY.BAT копирует произвольное число файлов rem в
каталог.
rem Используется следующий синтаксис:
rem mycopy dir файл1 файл2 ...
set todir=%1
:getfile
shift
if "%1"==" " goto end
copy %1 %todir%
goto getfile
:end
set todir=
echo Работа закончена

```

2.2 Контрольные вопросы

1. Почему командные файлы используются администраторами локальных компьютерных сетей.

2. Каковы особенности задания формального параметра в команде FOR, как в этой команде задаются фактические параметры.

3. Что такое переменные окружения, можно ли задать значения собственным переменным и каким образом. Как долго хранятся значения переменных окружения. Как получить из переменной окружения ее значение.

4. Какие условия в командных файлах можно проверить с помощью команды IF.

5. Что такое перенаправление вывода в командных файлах, как оно осуществляется, какие при этом применяются управляющие символы.

6. Команда FIND осуществляет поиск текстовой строки внутри файла или находит совпадение/несовпадение с именем файла.

7. Какое значение хранит формальный параметр %0.

8. В каких случаях в командных файлах используется команда SHIFT, как она работает, приведите пример использования этой команды в командном файле.

9. Как в командных файлах используется команда CALL, поясните, что такое рекурсия, приведите пример рекурсивной работы командного файла.

10. Что такое метки в командном файле, для чего они используются, совместно с какой командой применяются. Может ли метка быть строкой текста.

3 Экспериментальная часть

3.1 Задание на работу

Варианты заданий

1. Создать командный файл, выполняющий операцию проверки канала связи до заданного (в качестве параметра командного файла) компьютера в сети с выводом текстового сообщения о результате проверки.

2. Создать командный файл, проверяющий какие порты на локальной машине используются протоколом TCP и выводящий эту информацию в файл `working_ports.txt`.

3. Создать командный файл, записывающий в текстовый файл `wst_users.txt` информацию обо всех имеющихся на локальном компьютере пользователях, принадлежности их к имеющимся на компьютере группам, дате и времени последнего входа на локальный компьютер для каждого пользователя, может ли он изменять свой пароль и дата очередной смены пароля.

4. Создать командный файл, создающий на локальном компьютере 25 новых пользователей по списку в текстовом файле. Всем пользователям устанавливается одинаковый пароль и требование сменить его при первом входе.

5. Создать командный файл, проверяющий наличие статических записей в ARP-кэше, при обнаружении таких строк, они выводятся на экран в сопровождении соответствующего комментария.

6. Создать командный файл, считывающий информацию о конфигурации сетевых настроек локального компьютера при использовании его в качестве сервера и в качестве рабочей станции, вся информация сохраняется в текстовый файл `config.txt`.

7. Создать командный файл, кратковременно периодически запускающийся и выполняющий контроль объема полученного трафика. При превышении порогового значения трафика выполняется остановка службы WORKSTATION.

8. Создать командный файл, получающий список процессов, запущенных на удаленном компьютере и принудительно останавливающий процесс с номером 1024, если он есть в списке.

9. С помощью командного файла получить информацию о разрешениях доступа к папке на удаленном компьютере. Имя папки вводится в командной строке, как параметр командного файла.

10. Создать командный файл, выполняющий копирование всех файлов, перечисленных в текстовом файле, содержащем список указанных файлов, в директорию, введенную в командной строке, как параметр командного файла. В эту же директорию копируется в том числе и сам файл-список, при этом он переименовывается в файл `List.txt`.

3.2 Содержание отчета

Каждый студент должен представить преподавателю отчет по лабораторной работе, которой должен содержать:

1. Титульный лист.
2. Введение.
3. Результаты выполнения задания.
4. Список используемой литературы.

Рекомендуемая литература

1. Аппаратные средства локальных сетей: энциклопедия / М. Гук. - СПб. : Питер, 2005. – 572 с. - ISBN 5-8046-0113-X :
2. Компьютерные сети: Принципы, технологии, протоколы : учебное пособие для вузов / В. Г. Олифер, Н. А. Олифер. - 3-е изд. - СПб. : Питер, 2006. - 960 с. - ISBN 5-469-00504-6
3. Сетевые операционные системы : Учебное пособие для вузов / В. Г. Олифер, Н. А. Олифер. - СПб. : Питер, 2007. - 538[6] с. : - ISBN 5-272-00120-6 :
4. Локальные сети, модемы, интернет: ответы и советы / сост. : И. Грень. - Минск : Новое знание, 2004. – 350 с. : ил. - ISBN 985-475-059-0 :

Учебное пособие

Агеев Е.Ю.

Использование командных файлов

Методические указания к лабораторной работе
по дисциплине «Локальные компьютерные сети»

Усл. печ. л. _____ Препринт
Томский государственный университет
систем управления и радиоэлектроники
634050, г.Томск, пр.Ленина, 40