

Подробное руководство по работе с ВАТ-скриптом

Автор: Сечейко Николай

8 декабря 2024 г.

1 Введение

Представленный скрипт для Windows (ВАТ-файл) выполняет следующие задачи:

1. Запрашивает у пользователя имя папки.
2. Проверяет, существует ли указанная папка.
3. Если папка существует, скрипт находит в ней все текстовые файлы (*.txt).
4. Считывает содержимое каждого файла построчно, а затем разбивает строки на отдельные слова.
5. Анализирует каждое слово и, если оно начинается с буквы «v» или «V», записывает его в отдельный файл result.txt.
6. По окончании работы выводит сообщение о успешном завершении.

Ниже представлен исходный код скрипта, а также подробный разбор каждой его части. Все флаги и параметры, используемые в циклах и условиях, будут объяснены.

2 Исходный код скрипта

Листинг 1: Исходный код ВАТ-файла

```
1 @echo off
2 setlocal enabledelayedexpansion
```

```

3
4 set "BASEPATH=W:\pioivis_lab5"
5
6 set /p DIRNAME=Enter folder name:
7
8 if not exist "%BASEPATH%\%DIRNAME%" (
9     echo Folder "%DIRNAME%" not found %BASEPATH%.
10    pause
11    goto :EOF
12 )
13
14 echo Folder found!
15
16 if exist "%BASEPATH%\result.txt" del "%BASEPATH%\result.txt"
17
18 for %%f in ("%BASEPATH%\%DIRNAME%\*.txt") do (
19     for /f "usebackq delims=" %%L in ("%%f") do (
20         for %%W in (%%L) do (
21             set "word=%%W"
22             set "firstChar=!word:~0,1!"
23             if /i "!firstChar!"=="v" (
24                 echo %%W>>"%BASEPATH%\result.txt"
25             )
26         )
27     )
28 )
29
30 echo Words successfully written to result.txt file!
31 pause

```

3 Подробный разбор кода

3.1 Настройка среды и выключение эха

@echo off

setlocal enabledelayedexpansion

- @echo off — отключает отображение выполняемых команд в окне консоли, оставляя только их результат. Это делает вывод более аккуратным.
- setlocal enabledelayedexpansion — включает так называемую «отложенную подстановку переменных». Это позволяет использовать синтаксис !variable! внутри циклов for, чтобы немедленно обновлять и считывать значения переменных без выхода из цикла.

3.2 Определение базового пути и запрос имени папки

```
set "BASEPATH=W:\pioivis_lab5"  
set /p DIRNAME=Enter folder name:
```

- set "BASEPATH=..." задаёт переменную окружения BASEPATH, указывающую на корневой путь, где будет производиться обработка.
- set /p DIRNAME=... просит пользователя ввести имя папки. Введённое значение сохраняется в переменную DIRNAME. Флаг /p означает, что значение переменной задаётся с консоли.

3.3 Проверка существования папки

```
if not exist "%BASEPATH%\%DIRNAME%" (  
    echo Folder "%DIRNAME%" not found %BASEPATH%.  
    pause  
    goto :EOF  
)
```

- if not exist — проверяет, существует ли путь или файл. В нашем случае это проверка на существование директории. Если она не найдена, выводится сообщение об ошибке, скрипт приостанавливается (pause) и переходит к метке :EOF (конец файла), завершая работу.
- Если директория существует, проверка не срабатывает, и выполнение идёт дальше.

```
echo Folder found!
```

Если папка найдена, отображается подтверждающее сообщение.

3.4 Удаление предыдущего результата

```
if exist "%BASEPATH%\result.txt" del "%BASEPATH%\result.txt"
```

- if exist проверяет наличие файла result.txt.
- Если файл существует, команда del удаляет его, чтобы при новой обработке результат не дописывался к старым данным.

3.5 Обработка файлов

Основная логика расположена в цепочке вложенных циклов `for`:

```
for %%f in ("%BASEPATH%\%DIRNAME%\*.txt") do (  
    ...  
)
```

- `for %%f in (...) do` — цикл, который перебирает все элементы, соответствующие шаблону.
- Здесь используется шаблон `"%BASEPATH%\%DIRNAME%*.txt"`, то есть все текстовые файлы в указанной папке.
- `%%f` — переменная цикла, представляющая текущий файл.

Внутри этого цикла для каждого `%%f` выполняется следующий фрагмент кода.

3.5.1 Чтение файлов построчно

```
for /f "usebackq delims=" %%L in ("%%f") do (  
    ...  
)
```

Здесь используется цикл `for /f` для построчного чтения файлов.

- `/f` — указывает, что цикл будет работать с содержимым файла, чтением строки за строкой.
- `"usebackq delims="` — это набор опций для цикла:
 - `usebackq` — позволяет использовать кавычки для указания файлового имени. Без этого флага файл в кавычках может интерпретироваться неправильно. С данным флагом (`"%%f"`) однозначно понимается как файл, а не строка.
 - `delims=` — устанавливает разделители слов пустым. По умолчанию `for /f` разбивает строку по пробелам и табам, но здесь мы хотим получить строку целиком, поэтому убираем разделители. Это позволяет нам на следующем шаге самостоятельно разбить строку на слова.
- `%%L` — переменная цикла, хранящая текущую строку файла.

Таким образом, `for /f "usebackq delims=" %%L in ("%%f")` последовательно считывает каждую строку текстового файла, сохраняя её в `%%L`.

3.5.2 Разбиение строки на слова

```
for %%W in (%%L) do (  
    ...  
)
```

- Этот цикл без модификаторов /f разбивает строку %%L на отдельные слова по пробелам и табуляциям.
- %%W — переменная цикла, представляющая текущее слово.
- Важно отметить, что без флага /f цикл for просто разбивает содержимое в скобках на токены, разделённые пробелом. Это и есть наше разбиение строки, считанной ранее, на отдельные слова.

3.5.3 Анализ первого символа каждого слова

```
set "word=%%W"  
set "firstChar=!word:~0,1!"  
if /i "!firstChar!"=="v" (  
    echo %%W>>"%BASEPATH%\result.txt"  
)
```

- set "word=%%W" сохраняет текущее слово в переменную word.
- set "firstChar=!word:~0,1!" извлекает первый символ из word. Здесь :~0,1 означает «начиная с 0-го символа, длиной 1 символ».
- if /i "!firstChar!"=="v" — оператор if с флагом /i. Флаг /i делает сравнение нечувствительным к регистру, то есть «v» и «V» будут считаться равными. Если первый символ совпадает с «v» (в любом регистре), то условие истинно.
- Если условие истинно, выполняется команда echo %%W>>"%BASEPATH%\result.txt".
- >> означает дозапись в конец файла result.txt. Каждое совпавшее слово дописывается на новую строку.

Таким образом, на выходе мы получаем файл result.txt, содержащий все слова, начинающиеся на «v» или «V».

3.6 Завершение работы скрипта

echo Words successfully written to result.txt file!
pause

- По окончании обработки скрипт выводит сообщение о том, что все слова успешно записаны.
- Команда pause останавливает выполнение до нажатия любой клавиши, чтобы пользователь мог прочитать результат.

4 Вывод

Данный скрипт автоматически:

1. Запрашивает у пользователя папку.
2. Проверяет её существование.
3. Перебирает все текстовые файлы в этой папке.
4. Считывает каждый файл по строкам, затем разбивает строки на слова.
5. Выбирает только те слова, которые начинаются на заданный символ (в данном случае «v»), и записывает их в единый файл result.txt.

Используемые флаги и механизмы:

- @echo off — скрытие вывода команд.
- setlocal enabledelayedexpansion — позволяет обновлять переменные на лету внутри циклов.
- set /p — ввод данных от пользователя.
- if not exist — проверка наличия файлов/папок.
- del — удаление файлов.
- Циклы for и их флаги:
 - for %%f in (...) do — перебор файлов.
 - for /f "usebackq delims=" %%L in ("%%f") do — построчное чтение файлов с определёнными флагами.

- `for %%W in (%%L) do` — разбиение строк на слова.
- `if /i` — сравнение строк без учёта регистра.
- `echo ... >> file` — дозапись в конец файла.

В результате у вас есть универсальный пример, который можно модифицировать для решения других подобных задач текстовой фильтрации.