Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ОТЧЕТ

к лабораторной работе по дисциплине:

|  |
| --- |
| **Администрирование информационных систем** |
| Язык сценариев ОС Microsoft Windows «Power Shell» |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСМб-19-1 |  |  |  | Вовиков Д.Е. |
|  |  | подпись |  | Фамилия И.О |
| Проверил: |  |  |  |  | Хритова М.А. |
|  | должность |  | подпись |  | Фамилия И.О |

Иркутск 2022 г.

Содержание

[1 Введение 2](#_Toc119721895)

[2 Краткая информация о языке 4](#_Toc119721896)

[3 Описание всех действий и листинг скриптов 5](#_Toc119721897)

[3.1 Описание используемых команд 5](#_Toc119721898)

[3.1.1 Описание скрипта 1 5](#_Toc119721899)

[3.1.2 Описание скрипта 2 5](#_Toc119721900)

[3.1.3 Описание скрипта 3 5](#_Toc119721901)

[3.2 Листинг кода 5](#_Toc119721902)

[3.2.1 Скрипт 1 5](#_Toc119721903)

[3.2.2 Скрипт 2 6](#_Toc119721904)

[3.2.3 Скрипт 3 6](#_Toc119721905)

[4 Результаты работы скрипта 7](#_Toc119721906)

[4.1 Работа скрипта 1 7](#_Toc119721907)

[4.2 Работа скрипта 2 7](#_Toc119721908)

[4.3 Работа скрипта 3 8](#_Toc119721909)

[5 Заключение 10](#_Toc119721910)

1 Введение

**Цель работы:** Освоить язык сценариев «Power Shell».

**Задание:**

1. Изучить теоретическую информацию о языке сценариев Power Shell.
2. Написать 2-3 простейших сценария, оформив их в виде скрипта. Варианты сценариев студент придумывает сам.

Примеры сценариев:

* Копирование нескольких файлов на сменный носитель в определенное время (учитывать возможность изменения буквы диска).
* Изменения прав доступа по шаблону в папке студента.
* Отобразить список служб на компьютере пользователя.
* Отсортировать все службы на компьютере пользователя на запущенные и остановленные.
* Создать подкаталог текущего каталога и скопировать в него файлы с расширениями PAS, BAS, C и TXT, присвоив им атрибут «только для чтения»;
* В заданном каталоге выдать на экран список имен файлов с расширениями PAS,TXT, BAT.
* Отсортировать указанный файл текущего каталога по убыванию 2-го символа и выдать его на экран.

1. Написать отчет с приложением листинга скрипта.

**Требования к отчету:** Отчетный документ представляется в печатном или электронном виде, в форматах doc(x).

Отчет оформляется согласно действующему нормоконтролю и содержит следующие главы:

1. Введение

2. Краткая информация о языке.

3. Описание всех действий и листинг скрипта с комментариями.

4. Результаты работы скрипта, со скриншотами.

5. Заключение (с анализом возможностей Power Shell).

2 Краткая информация о языке

PowerShell — расширяемое средство автоматизации от Microsoft с открытым исходным кодом, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев. Впервые публично язык был продемонстрирован на Профессиональной конференции разработчиков в сентябре 2003 года под кодовым названием Monad. Изначально средство выпущено как компонент Windows под названием Windows PowerShell. Открытый исходный код был выпущен с началом разработки PowerShell Core в августе 2016 года.

Команды, исполняемые в Windows PowerShell, могут быть в форме командлетов, которые являются специализированными классами .NET, созданными с целью предоставления функциональности в PowerShell в виде сценариев PowerShell (.PS1) или являются обычными исполняемыми файлами. Если команда является исполняемым файлом, то PowerShell запускает её в отдельном процессе; если это команда, то он исполняется внутри процесса PowerShell. PowerShell предоставляет интерфейс командной строки, в котором можно вводить команды и отображать выводимые ими данные в текстовом виде. Этот пользовательский интерфейс, базирующийся на стандартном механизме консоли Windows, предоставляет настраиваемый механизм автозавершения команд, но не обладает возможностью подсветки синтаксиса, хотя при желании её можно обеспечить. В PowerShell также можно создавать псевдонимы (англ. alias) для командлетов, которые при вызове преобразуются в оригинальные команды. Кроме того, поддерживаются позиционные и именованные параметры для командлетов. При выполнении командлета работа по привязке значений аргументов к параметрам выполняется самим PowerShell, но при вызове внешних исполняемых файлов аргументы передаются им для самостоятельного разбора.

3 Описание всех действий и листинг скриптов

3.1 Описание используемых команд

3.1.1 Описание скрипта 1

Add-Type – добавляет класс Microsoft .NET в сеанс PowerShell.

-AssemblyName – ключ названия библиотеки.

System.Speech – класс для инициализации и настройки обработчика синтеза речи, создания сообщений, создания речи, реагирования на события и изменения характеристик голоса.

System.Speech.Synthesis.SpeechSynthesizer – классы для инициализации и настройки обработчика синтеза речи.

.SelectVoice("Microsoft Irina Desktop") – выбор голосового пакета для озвучивания текста.

.Speak($text) – озвучивание введенного текста.

Read-Host – считывает строку ввода из консоли.

3.1.2 Описание скрипта 2

Read-Host – считывает строку ввода из консоли.

Send-MailMessage – коммандлет отправки сообщений электронной почты.

-From – параметр указывает адрес электронной почты отправителя.

-To – параметр указывает адрес электронной почты получателя.

-Cc – Указывает адреса электронной почты, на которые отправляется точная копия сообщения электронной почты.

-Subject – этот параметр указывает тему сообщения электронной почты.

-Body – задает содержимое сообщения электронной почты.

-SmtpServer – указывает имя SMTP-сервера, который отправляет сообщение электронной почты.

-port – указывает альтернативный порт на SMTP-сервере. Значение по умолчанию — 25, это порт SMTP по умолчанию.

-UseSsl – протокол Secure Sockets Layer (SSL) используется для установки безопасного соединения с удаленным компьютером для отправки почты.

3.1.3 Описание скрипта 3

Get-Service – получает службы на компьютере.

-Name - указывает имена служб, которые необходимо получить.

Select – выбирает объекты или свойства объектов.

Out-File – отправляет вывод в файл.

3.2 Листинг кода

3.2.1 Скрипт 1

#Озвучивание введенного текста

function SpeechText($text){ #Метод озвучки текста

Add-Type -AssemblyName System.Speech #Подключаем библиотеку озвучки текста

$voice = New-Object System.Speech.Synthesis.SpeechSynthesizer #Предоставление доступа к функциям установленного модуля синтеза речи

$voice.SelectVoice("Microsoft Irina Desktop") #Выбор голосового модуля

$voice.Speak($text) #Озвучка текста

}

$text = Read-Host "Введите текст для озвучивания:" #Считываем текст из консоли

if (!$text) { #Проверка ввода на пустоту

SpeechText("Вы ничего не ввели! Я обиделась!")

}

else {

SpeechText($text)

}

3.2.2 Скрипт 2

#Отправка сообщений по почте (мини почтовый клиент)

$From = Read-Host "Введите адрес своей почты" #Считываем текст из консоли с адресом собственной почты

$To = Read-Host "Введите адрес почты получателя" #Считываем текст из консоли с адресом почты на которую отправляем сообщение

$Cc = Read-Host "Введите адрес почты получателя (копия)" #Считываем текст из консоли с запасным вариантом почты

$Subject = Read-Host "Введите тему сообщения" #Считываем текст из консоли с темой письма

$Body = Read-Host "Введите текст сообщения" #Считываем текст из консоли с текстом сообщения

$SMTPServer = "smtp.gmail.com" #Указываем smtp сервер Google

$SMTPPort = "587" #Указываем порт для отправки

#Используем командлет для отправки сообщения со следующими параметрами: собственный адрес, адрес получателя, запасной адрес получателя

#тему сообщения, текст сообщения, smtp сервер, порт smtp сервера, пустое свойство ssl-зашиты и получаем данные учетной записи (авторизируемся в аккаунте с почтой)

Send-MailMessage -From $From -to $To -Cc $Cc -Subject $Subject -Body $Body -SmtpServer $SMTPServer -port $SMTPPort -UseSsl -Credential (Get-Credential)

3.2.3 Скрипт 3

#Выгрузка информации о Сервисах в txt

#Получает службы на компьютере по ключу имени (\* для всех выдачи всех сервисов)

#Выбираем все данные, полученные из Get-Service

#Сохраняем данные в корневой директории со скриптом в файл response.txt

Get-Service -Name \* | Select \* | Out-File -FilePath response.txt

4 Результаты работы скрипта

4.1 Работа скрипта 1

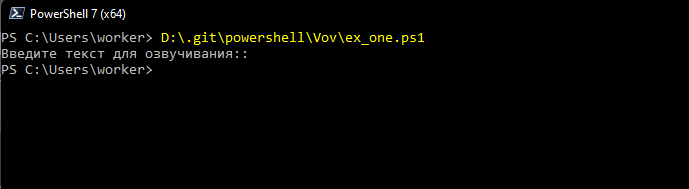


Рисунок 1 – Ввод пустой строки (озвучивание текста с информацией о вводе пустой строки)

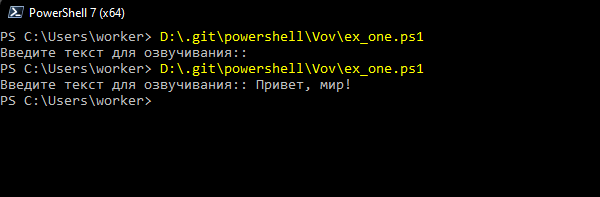


Рисунок 2 – Ввод текста (озвучивание текста)

4.2 Работа скрипта 2

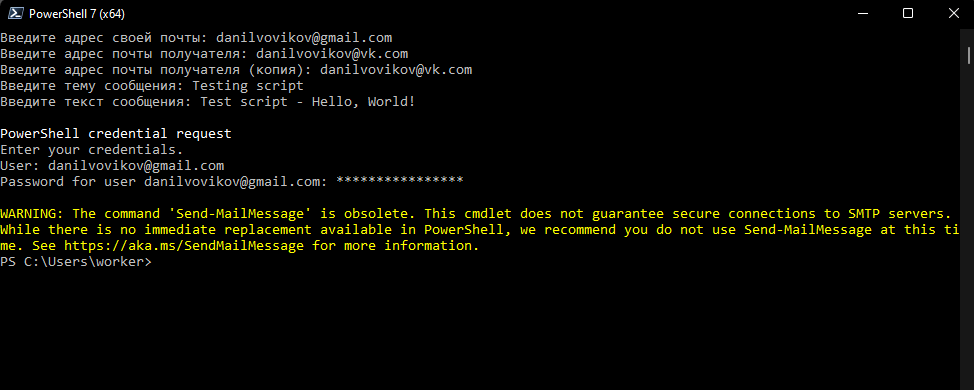


Рисунок 3 – Выполнение скрипта

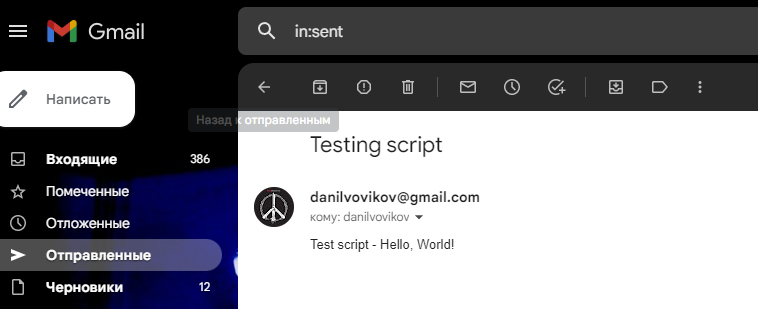


Рисунок 4 – Результат выполнения скрипта

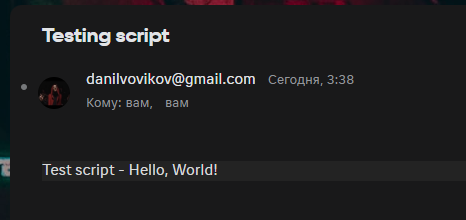


Рисунок 5 – Результат выполнения скрипта

4.3 Работа скрипта 3

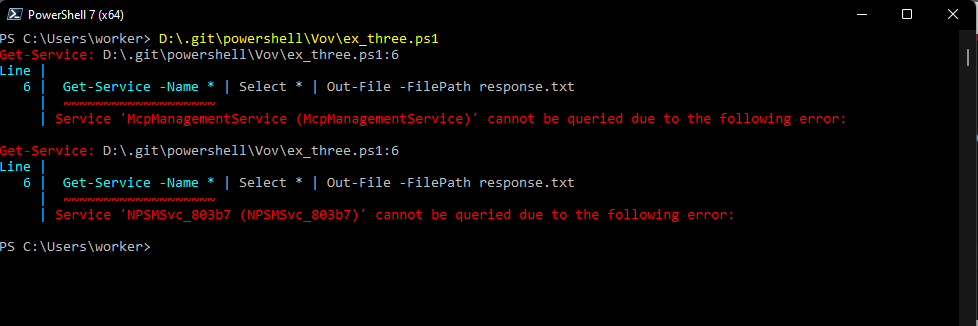


Рисунок 6 – Выполнение скрипта

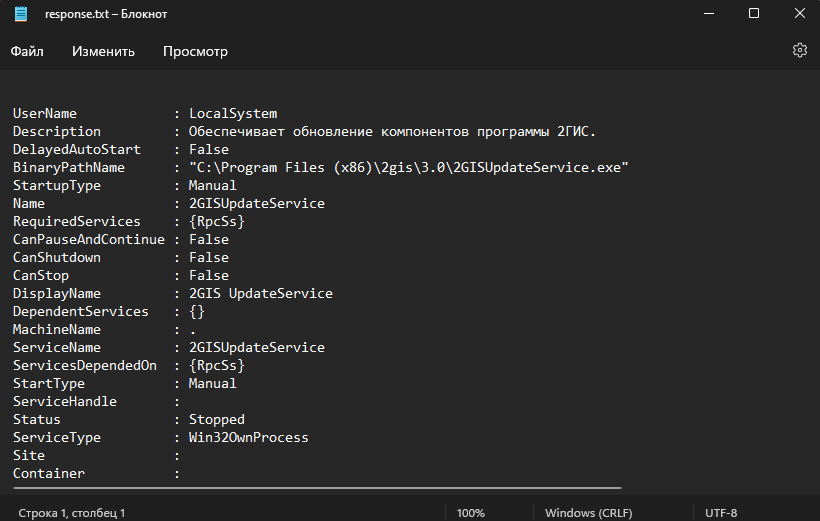


Рисунок 7 – Результат выполнения скрипта

5 Заключение

Несмотря на возможность выполнения большинства функций командной строки в Windows PowerShell, функционал приложений не совпадает.

В то время как Cmd использует команды DOS, PowerShell использует команды, известные как командлеты, которые представляют собой независимые программные объекты, позволяющие получить доступ к основным параметрам администрирования в Windows. Эти параметры недоступны из Cmd, что означает, что пользователи должны перемещаться по графическому интерфейсу, чтобы получить к ним доступ.

Cmd – это просто система оболочки, которая ограничивает пользователей выполнять только простые и базовые сценарии для выполнения командного файла. Однако PowerShell – это и оболочка, и среда сценариев, которая поддерживает создание больших и сложных файлов для управления системой Windows.

Командная строка не предназначена для системного администрирования, и пользователям требуются дополнительные утилиты для расширения ее функциональности. Напротив, в Windows PowerShell есть множество командлетов, которые упрощают жизнь системным администраторам. Он позволяет выполнять широкий спектр задач и автоматизацию, включая управление пользователями и разрешениями, конфигурации безопасности, извлечение данных и администрирование Active Directory.

Главной особенностью среды PowerShell, отличающей ее от всех других оболочек командной строки, является то, что единицей обработки и передачи информации здесь является объект, а не строка текста.