Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

|  |  |
| --- | --- |
| Факультет | информационных технологий и управления |
| Кафедра | информационных технологий автоматизированных систем |

Лабораторная работа №2

**ТЕХНОЛОГИЯ АДМИНИСТРИРОВАНИЯ КОРПОРАТИВНЫХ СЕТЕЙ НА ОСНОВЕ WINDOWS POWERSHELL**

|  |  |
| --- | --- |
| Выполнили: | Проверил: |
| Студенты группы 820601  Бурак А.Ю.  Козловский Г.И.  Свидунович И.В. | Ярмолик В.И. |

Минск 2021

1. **Цель работы:**

Освоение технологии администрирования корпоративных сетей *Microsoft* *Windows* средствами *PowerShell*.

1. **Краткая характеристика *Windows PowerShell* и результаты выполнения индивидуального задания.**

**Общая характеристика *Windows* *PowerShell***

Недостатки традиционных средств автоматизации управления и администрирования *Microsoft Windows (cmd.exc, WMIC, WSH)*, усугубляемые усложнением сетевых корпоративных технологий, привели к созданию *Windows PowerShell (PS). Windows PowerShell* как технология достаточно проста и вместе с этим предоставляет широкие возможности системным инженерам и администраторам в деле автоматизации управления корпоративными сетями. *Windows PowerShell* как технология предоставляет практически неограниченные возможности по настройке серверов, виртуальных машин и сбору информации об их состоянии.

Истоки создания *PowerShell* (первоначальное название *Monad*) связаны с доработкой *WMIC (Windows Management Instrumentation Command)* для обеспечения доступа из командной строки к любым классам платформы *.NET Framework. Windows PowerShell* является и оболочкой командной строки и средой выполнения сценариев. Сценарии пишутся на новом языке. Язык сценариев *PowerShell* поддерживает управление объектами .*NET*, обладает совместимостью с языками, используемыми при программировании для .*NET* и имеет синтаксическое сходство с *C#* (язык *PowerShell* по существу представляет собой упрощённый *C#);* это обусловлено тем, что оболочка *PowerShell* основана на *.NET Framework*.

*Windows* *PowerShell* принципиально отличается от других оболочек тем, что PS обрабатывает не текст, а объекты платформы *.NET. Windows PowerShell* содержит встроенные команды (командлеты; *cmdlets*), которые имеют унифицированный интерфейс и обрабатываются одним синтаксическим анализатором. Командлет представляет собой команду, выполняющую одну единственную функцию.

Версия 1.0 *Windows* *PowerShell* была выпущена в 2006 г. Открытый исходный код был выпущен с началом разработки *PowerShell Core* в августе 2016 года.

*Microsoft* выпустила *PowerShell* 2.0 в составе *Windows* 7 и *Windows* *Server* 2008 *R2*. *Windows* *PowerShell* 2.0 предустановлен в этих системах. Исключением является режим установки *Windows* *Server* 2008 *R2* в режиме *Core*, где *PowerShell* 2.0 можно доустановить вручную. Для более старых платформ он доступен в составе *Windows* *Management* *Framework*.

*PowerShell* 2.0 вносит некоторые изменения в язык сценариев и *API* для встраивания, в дополнение к новым 240 командлетам.

**Команднаяоболочка *PowerShell***

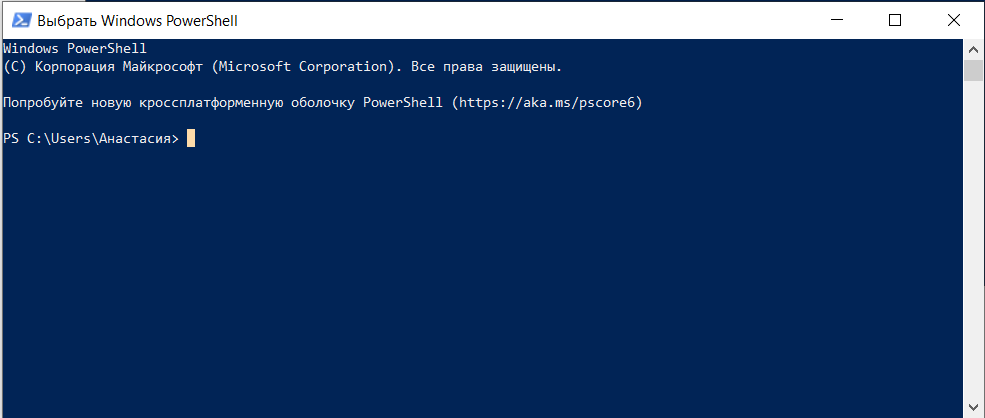


Рис. 1 – окно оболочки *Windows PowerShell*

В дополнение к традиционному копированию в окно классической *CMD* в *PowerShell* можно осуществлять копирование и вставку любой выделенной ("подсвеченной" левой кнопкой мыши) текстовой части самого окна *PowerShell* (необходимо выполнить двойной клик правой кнопкой мыши на подсвеченном блоке; это приведет к вставке данного блока на позицию мигающего курсора в окне *PS*).

Процесс копирования в *PowerShell:*

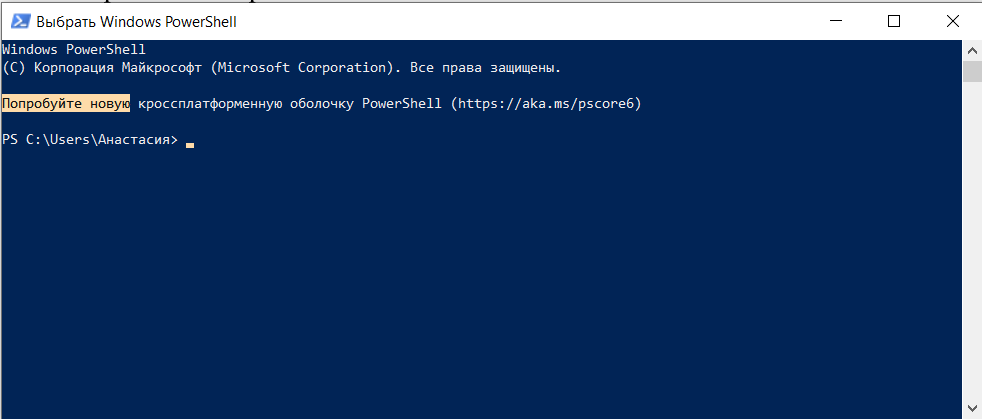
**

Рис. 2 – выделение текстовой части окна *PS*

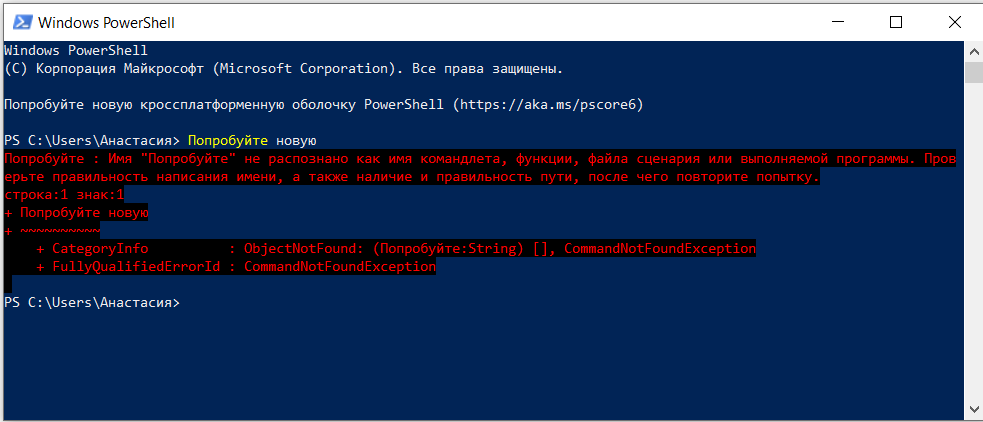
**

Рис. 3 – вставка выделенной текстовой части в *PS*

Другой особенностью, отличающей командную оболочку *PS* от классической оболочки *CMD*, является возможность использовать оболочку *PS* как калькулятор.

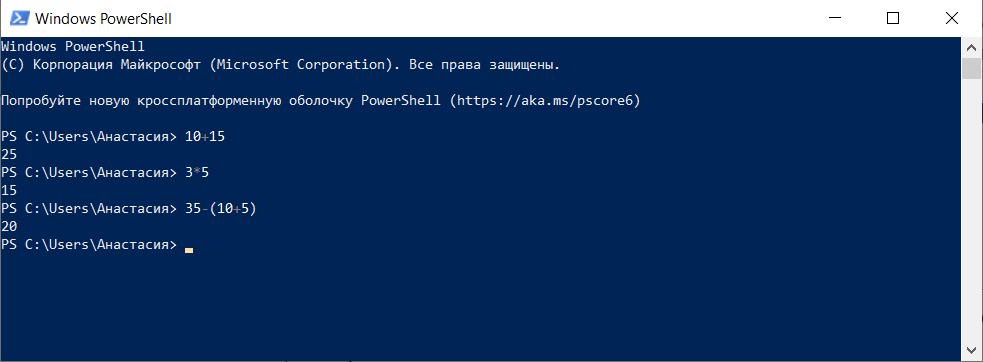


Рис. 4 – использование оболочки *PowerShell* как калькулятора

**Команды *PowerShell***

Главное же отличием *PowerShell* от предыдущих командных оболочек *Windows* состоит в том, что *Windows* *PowerShell* является объектно-ориентированной командной оболочкой, работающей на основе .*NET* *Framework*.

*Windows* *PowerShell* предоставляет значительно больше команд, чем классическая командная строка. *PowerShell* *v.* 1.0 содержит более 100 командлетов; (в *v.* 2.0 их более 200). Команды *PowerShell* состоят из глагола и существительного, разделенных тире. Например: *Get-Help* (вызов справки по синтаксису *PowerShell*). В *PowerShell* включены также псевдонимы многих известных команд; так, например, команды *help* (стиль Windows) и *man* (стиль Unix) работают аналогично команде *Get-Help*.

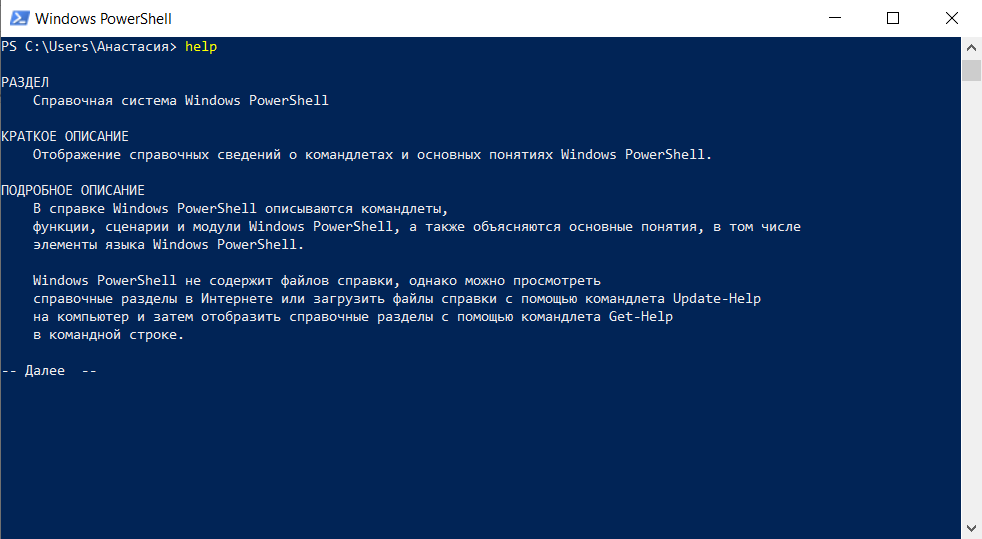


Рис. 5 – пример работы команды *help* в *PS*

Оболочка *PS* поддерживает следующие типы команд:

* внешние исполняемые файлы,
* командлеты,
* функции,
* сценарии.

**Внешние исполняемые файлы** – это обычные, выполняемые операционной системой файлы. Например, внешние команды интерпретатора *cmd.exe (cd, dir, copy* и др.). Для ознакомления с этими командами введите в окне CMD команду help. Начинать практическое освоение *PS* следует именно с этих команд.

**Командлет** используется внутри *PowerShell*. Командлеты является классами .*NET*, порожденными базовым классом *CMDLET*, который гарантирует их совместимый синтаксис. Командлеты компилируются в динамическую библиотеку *DLL* и подгружаются к процессу *PS* во время запуска оболочки *PS*. Каждый из командлетов (очень простой или сложный) выполняет узкую задачу. Посредством вертикальной черты ( | ) командлеты могут быть организованы в конвейер, в котором объекты предаются от одного командлета к другому.

**Функция** в *PowerShell* представляет собой блок кода на языке *PS*, имеющий название и хранящийся в памяти до завершения текущего сеанса командной строки.

Как и в других языках программирования в *PS* при описании функции можно задавать список формальных параметров, значения которых при выполнении функции будут заменены значениями переданных аргументов.

**Сценарии**. Сценарий представляет собой код на языке *PS*, хранящийся во внешнем файле с расширением *ps1*. Важной особенностью в плане сетевой безопасности является то обстоятельство, что скрипт с расширением *ps1* невозможно запустить на исполнение иначе как, запустив оболочку *PS*; так что для запуска сценария с расширением *ps1* необходимо запустив оболочку *PS*, ввести имя файла и нажать клавишу *Enter*.

Сценарии позволяют работать с *PowerShell* в пакетном режиме с заранее созданными командами на основе управляющих инструкций языка *PowerShell*. Сценарии в *PowerShell* в отличие от сценариев *WSH* и командных файлов *cmd.exe* можно писать непосредственно в самой оболочке, перенося затем готовый отлаженный код во внешний файл.

1. **Выводы по работе.**

Мы изучили основы оболочки *PowerShell* для администрирования корпоративных сетей *Microsoft* *Windows*, ее основные отличия от командной оболочки *CMD*, а также принципиальные различия сценариев *WSH* и *PS*.