Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра «Вычислительные методы и программирование»

ОТЧЕТ

По лабораторной работе №2

# Технология администрирования корпоративных сетей на основе *Windows PowerShell*

|  |  |  |
| --- | --- | --- |
| Проверил: |  | Выполнил: |
| Ярмолик В.И. |  | студенты ФИТиУ |
|  |  | Группы 820601 |
|  |  | Пенда В.Д.  Пальчик А.М. |

.

Минск 2021

**Цель работы:**

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

# Общая характеристика *Windows PowerShell*

# Недостатки традиционных средств автоматизации управления и администрирования *Microsoft Windows* (cmd.exe, *WMIC, WSH;* см лабораторную работу № 1), усугубляемые усложнением сетевых корпоративных технологий, привели к созданию *Windows PowerShell* (PS). *Windows* PS как технология достаточно проста и вместе с этим предоставляет широкие возможности системным инженерам и администраторам в деле автоматизации управления корпоративными сетями. *Windows PS* как технология предоставляет практически неорганичные возможности по настройке серверов, виртуальных машин и сбору информации об их состоянии

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

# *Windows PS* 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. Она доступна для *Windows XP SP2/SP3*, *Windows Server 2003, Windows Vista,* и встроена в *Windows Server 2008*. Версия 2.0 *Windows PowerShell* является компонентом *Windows 7 и Windows Server 2008 R2*.  Версия 2.0 PS доступна для *Windows XP SP3, Windows Server 2003 SP2, Windows Vista SP1 и Windows Server 2008.*

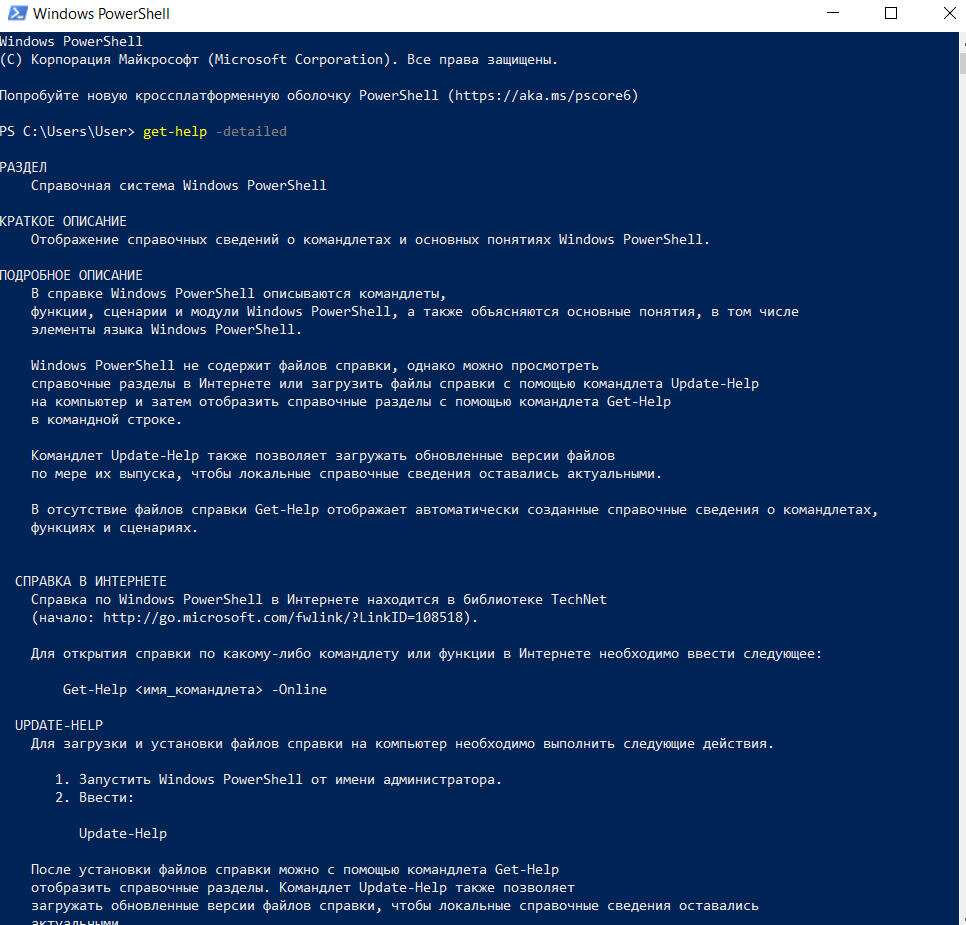


Рисунок 1. Фрагмент вывода команды *Get-Help.*

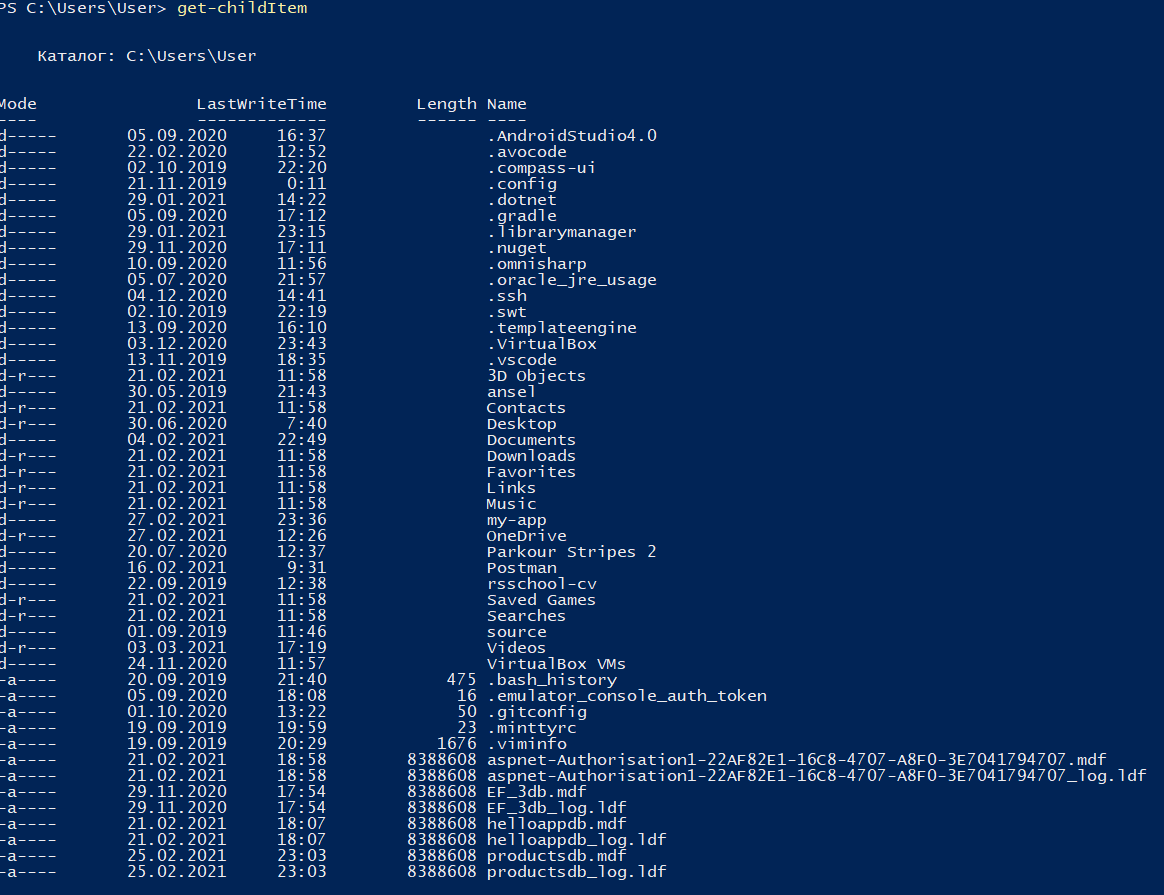


Рисунок 2. Фрагмент вывода команды *Get-ChildItem.*

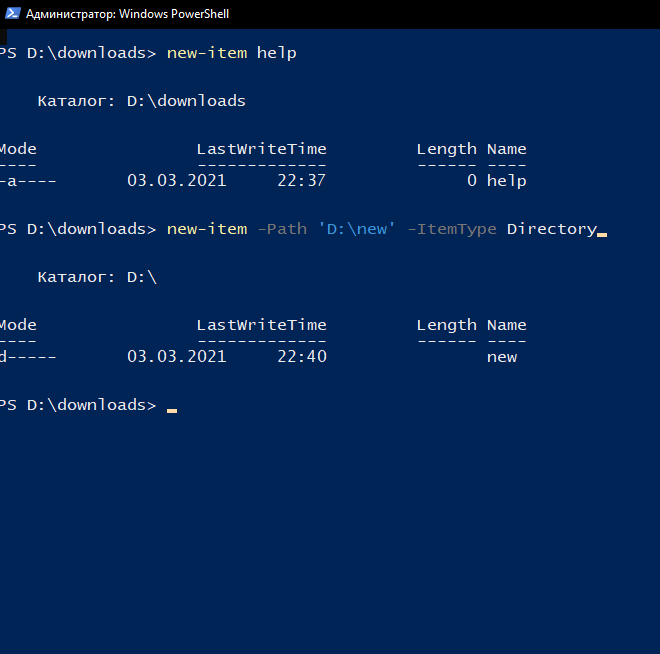


Рисунок 3. Фрагмент вывода команды *New-Item.*

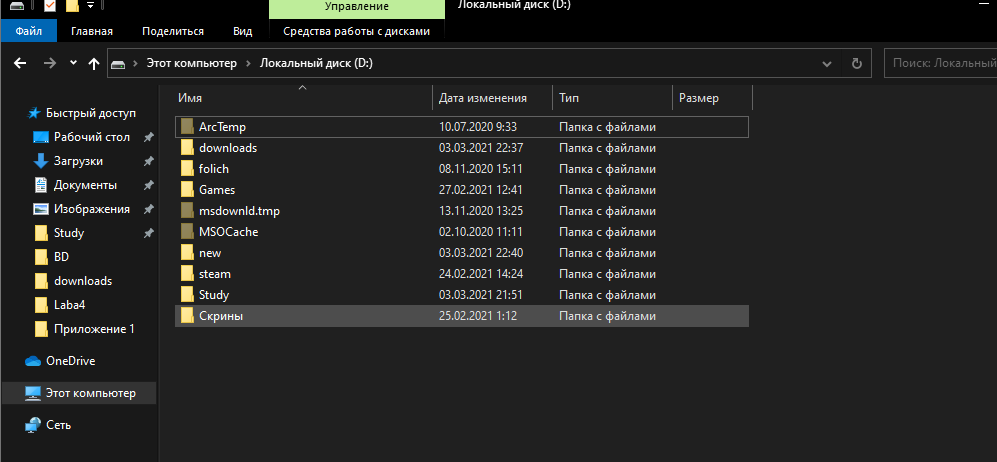


Рисунок 4. Результат выполнения команды *New-Item.*

**Индивидуальное задание:**

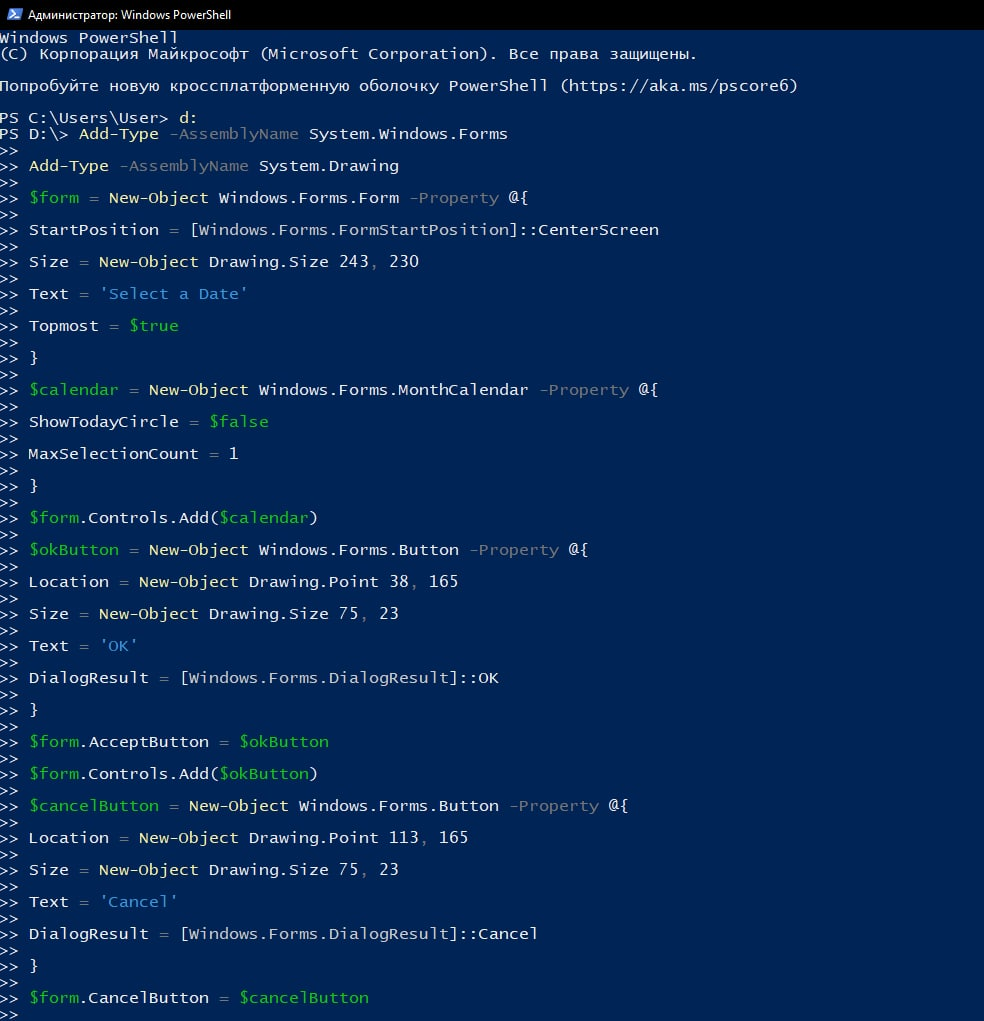


Рисунок 5. Фрагмент создания графического элемента управления «Выбор даты».

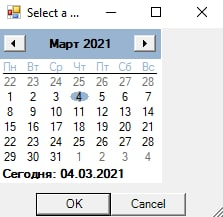


Рисунок 6. Результат созданного графического элемента управления «Выбор даты».

**Приложение**

Add-Type -AssemblyName System.Windows.Forms

Add-Type -AssemblyName System.Drawing

$form = New-Object Windows.Forms.Form -Property @{

StartPosition = [Windows.Forms.FormStartPosition]::CenterScreen

Size = New-Object Drawing.Size 243, 230

Text = 'Select a Date'

Topmost = $true

}

$calendar = New-Object Windows.Forms.MonthCalendar -Property @{

ShowTodayCircle = $false

MaxSelectionCount = 1

}

$form.Controls.Add($calendar)

$okButton = New-Object Windows.Forms.Button -Property @{

Location = New-Object Drawing.Point 38, 165

Size = New-Object Drawing.Size 75, 23

Text = 'OK'

DialogResult = [Windows.Forms.DialogResult]::OK

}

$form.AcceptButton = $okButton

$form.Controls.Add($okButton)

$cancelButton = New-Object Windows.Forms.Button -Property @{

Location = New-Object Drawing.Point 113, 165

Size = New-Object Drawing.Size 75, 23

Text = 'Cancel'

DialogResult = [Windows.Forms.DialogResult]::Cancel

}

$form.CancelButton = $cancelButton

$form.Controls.Add($cancelButton)

$result = $form.ShowDialog()

if ($result -eq [Windows.Forms.DialogResult]::OK) {

$date = $calendar.SelectionStart

Write-Host "Date selected: $($date.ToShortDateString())"

}

**Вывод**: научились применять технологии администрирования корпоративных сетей *Microsoft Windows* средствами *PowerShell,* изучили командлеты.