

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра систем автоматизированного проектирования

ОТЧЕТ
по лабораторной работе №2-2
«УПРАВЛЕНИЕ WINDOWS С ПОМОЩЬЮ КОМАНДНОЙ СТРОКИ
И POWERSHELL»
по дисциплине «Операционные системы»

Студент гр. 3353

Преподаватель

Карпенко А.Ю.

Шинкарь К.Д.

Горячев А.В.

Санкт-Петербург

2025

Оглавление

Цель работы	3
Задачи	3
Ход работы.....	4
Упражнение 1 – Использование инструментов командной строки.....	4
Упражнение 2 – Использование внешних наборов инструментов (Sysinternals).....	9
Упражнение 3 – Запуск PowerShell. Получение справочной информации.	11
Выводы.....	23

Цель работы

Знакомство с управлением компонентами операционной системы Windows с помощью командной строки и PowerShell.

Задачи

1. **Запустить командную строку от имени администратора.** В заголовке окна открыть пункт «Свойства», объяснить назначение параметров. Настроить шрифт «на покрупнее».
2. **Просмотреть список известных ОС команд.** Найти способ получить информацию о каждой команде.
3. **Вывести список команд в файл на диске C:** `HELP >C:\CmdList.txt`. Просмотреть результат. Выполнить команду `HELP >>C:\CmdList.txt` и объяснить разницу между `>` и `>>`.
4. **Выполнить команду** `COPY C:\cmdList.txt CON.` Объяснить результат.
5. **Вывести список всех файлов в каталоге C:\Windows, имя которых начинается с буквы Т.**
6. **Создать файл с расширением CMD.** Написать в нём последовательность команд, которая:
 - Выводит на экран название файла.
 - Выведет список файлов из каталога C:\Windows, имя которых начинается с буквы, введённой в качестве первого параметра.
 - Остановит вывод до нажатия любой клавиши, после чего запустит текстовый редактор Notepad.exe.
7. **Скачать набор инструментов Sysinternals Suite.** Выбрать несколько утилит командной строки, скопировать их в виртуальную машину и запустить.
8. **Запустить PowerShell от имени администратора.** Ввести командлет `Get-Command` и оценить результат.
9. **Найти командлет, выводящий список всех активных сервисов.** Выполнить его.
10. **С помощью командлета Get-Process вывести информацию о текущих процессах в текстовый файл Proc-Info.txt.**
11. **Использовать механизм конвейера для сортировки информации, выдаваемой Get-Process, по имени процесса.** Подсчитать количество процессов, оставить те, имена которых начинаются на букву «М».
12. **Создать переменную \$I со значением 9.** Проверить её тип, изменить значение на строку и снова проверить тип.
13. **Создать массив \$m с пятью строками.** Добавить ещё один элемент, удалить второй элемент и вывести содержимое массива.
14. **Создать хэш-таблицу с тремя элементами.** Добавить новый элемент, удалить один из существующих и изменить значение оставшегося.
15. **Запустить скрипт в PowerShell ISE.** Создать переменную `$F = 8`, вывести её значение на экран и сохранить скрипт как `Test1.ps1`. Запустить скрипт через файловый менеджер и описать результат.

Ход работы

Упражнение 1 – Использование инструментов командной строки.

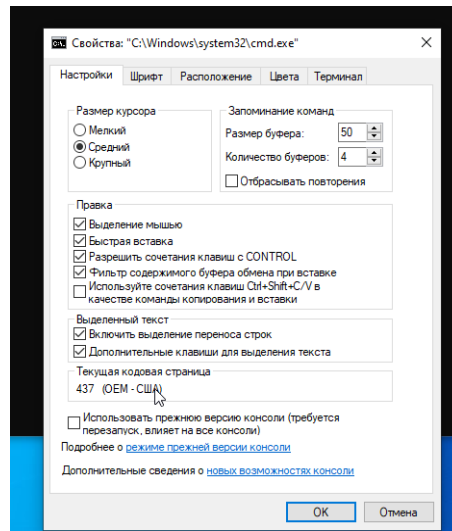


Рисунок 1 Свойства командной строки.

Настройка свойств командной строки, таких как размер шрифта, цвет фона и размер окна. Это делает работу с командной строкой более удобной.

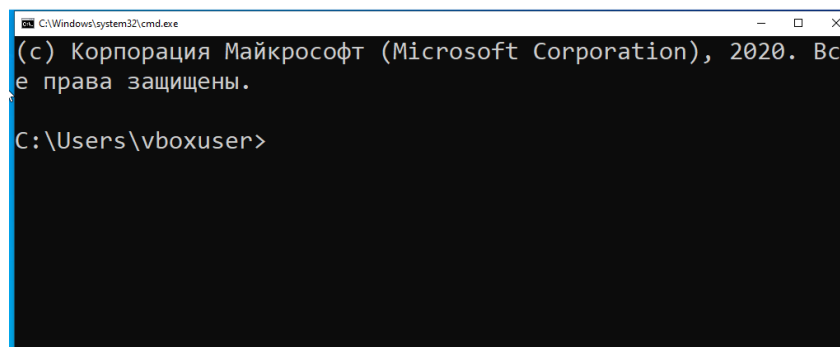


Рисунок 2 Открытие командной строки.

Запуск командной строки через меню "Пуск" или с помощью сочетания клавиш Win + R и ввода cmd. Командная строка может быть запущена как от имени пользователя, так и от имени администратора.

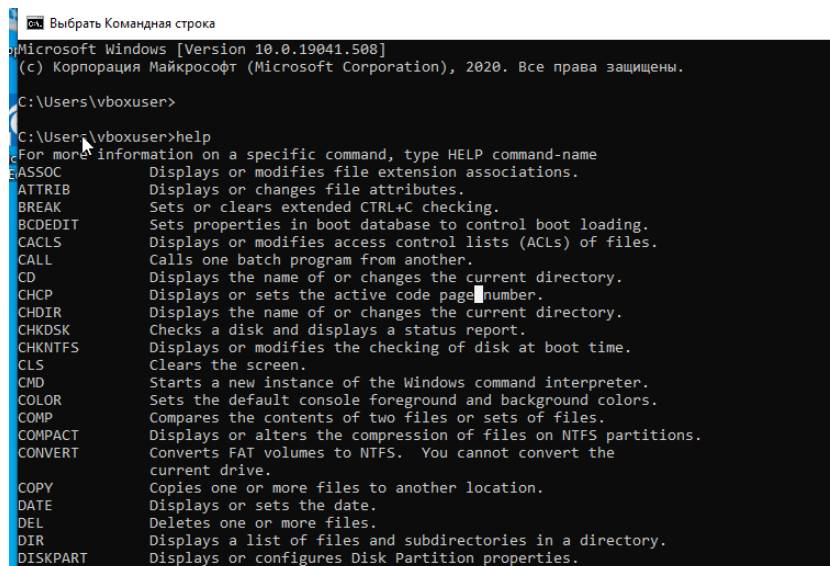


Рисунок 3

Команда **help** выводит список встроенных команд Windows, которые можно использовать в командной строке.

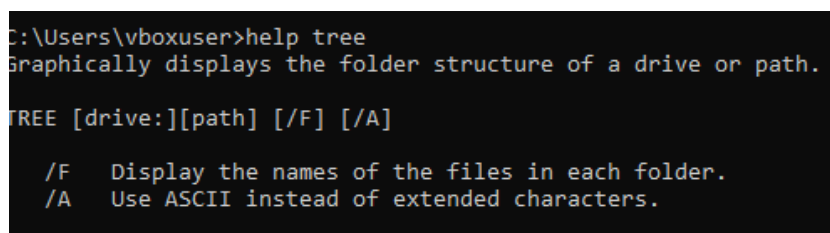


Рисунок 4

Проверили, являются ли некоторые команды идентичными или одна из них ссылается на другую, используя **help** для каждой команды.

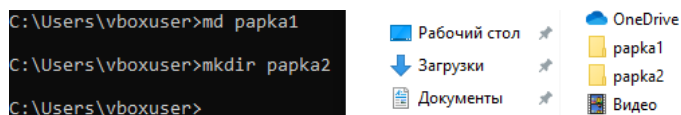


Рисунок 5

md и **mkdir**: Обе команды создают новые каталоги (папки). Они идентичны. **md** - сокращенная версия

```

C:\Users\vboxuser>help md
Creates a directory.

MKDIR [drive:]path
MD [drive:]path

If Command Extensions are enabled MKDIR changes as follows:

MKDIR creates any intermediate directories in the path, if needed.
For example, assume \a does not exist then:

    mkdir \a\b\c\d

is the same as:

    mkdir \a
    chdir \a
    mkdir b
    chdir b
    mkdir c
    chdir c
    mkdir d

which is what you would have to type if extensions were disabled.

```

Рисунок 6

Обе команды создают директории, и их можно использовать взаимозаменяемо.

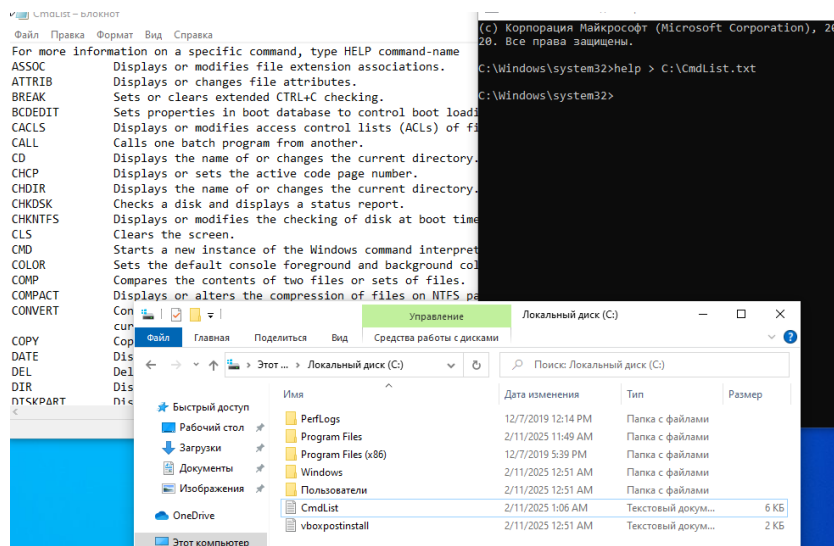


Рисунок 7

Команда > перенаправляет вывод команды help в файл CmdList.txt на диске C:. Файл будет содержать список всех доступных команд.

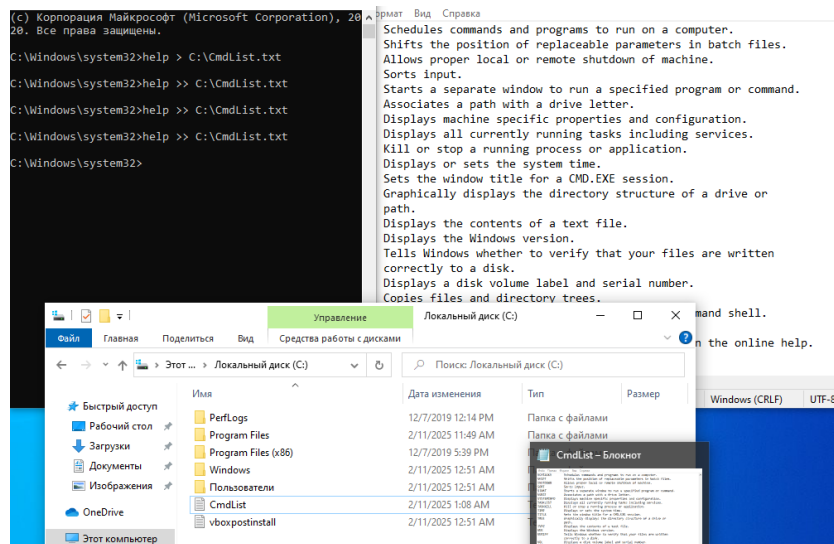


Рисунок 8

Команда >> добавляет вывод команды help в конец файла CmdList.txt, не перезаписывая его.

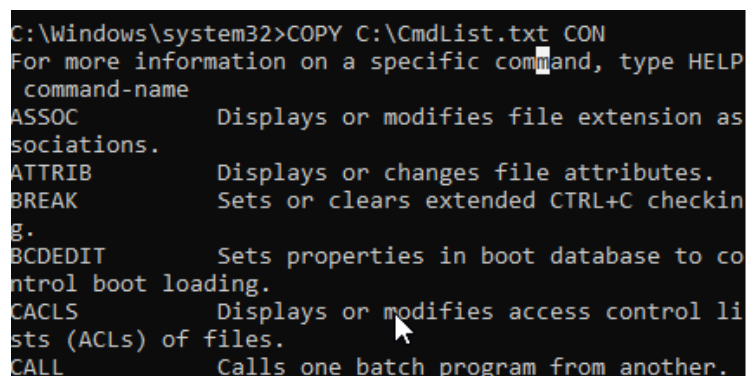


Рисунок 9

CON — это устройство, представляющее консоль (экран). Команда COPY копирует содержимое файла на экран.

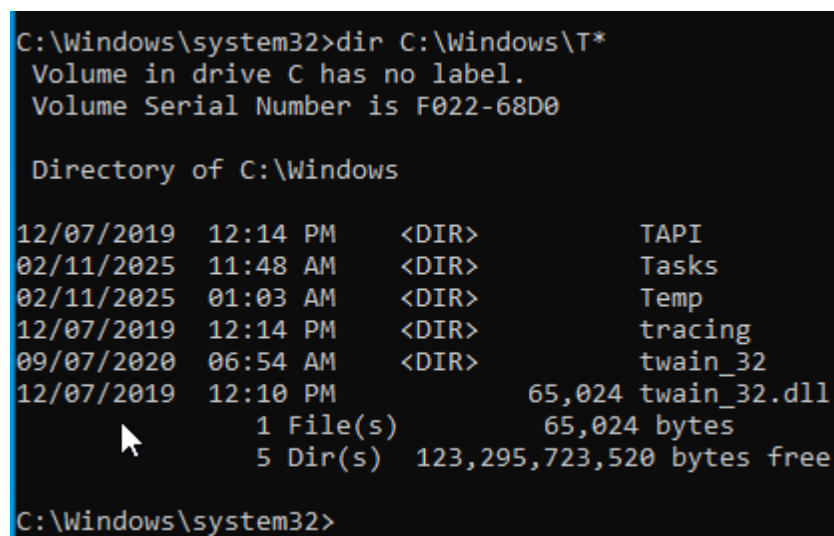
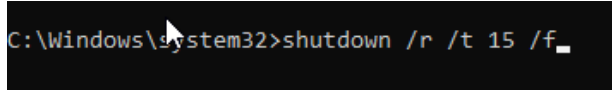


Рисунок 10

Символ * используется для поиска файлов по маске. В данном случае T* означает, что имена файлов должны начинаться с T.



```
C:\Windows\system32>shutdown /r /t 15 /f_
```

Рисунок 11 Компьютер перезагрузился через 15 секунд

Параметр /r указывает на перезагрузку, а /t 15 задает задержку в 15 секунд. Параметр /f (если добавить) закроет все открытые программы без предупреждения.

Далее в Notepad вписали скрипт

```
@echo off
```

```
echo Название файла: %0
```

```
echo Введенная буква: %1
```

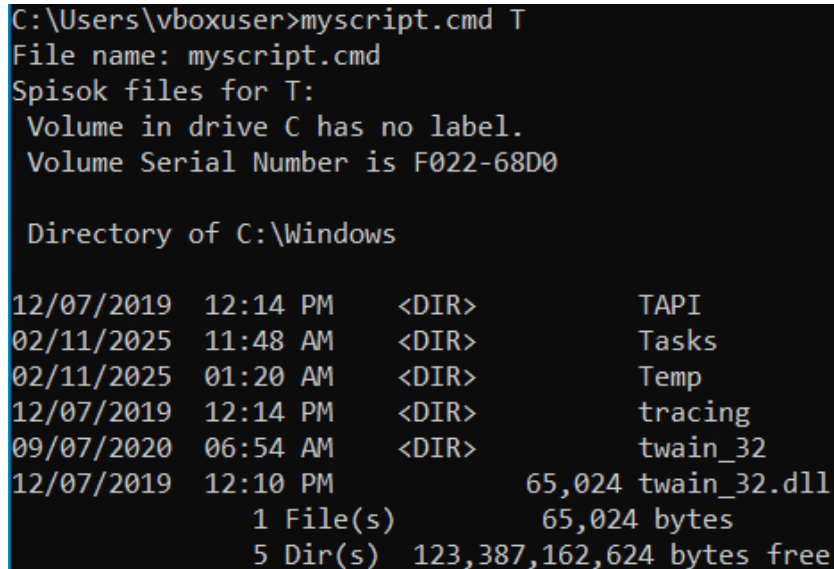
```
DIR C:\Windows\%1*
```

```
pause
```

```
notepad.exe
```

Сохранили файл MyScript с расширением .cmd.

Скрипты позволяют автоматизировать повторяющиеся задачи.



```
C:\Users\vboxuser>myscript.cmd T
File name: myscript.cmd
Spisok files for T:
Volume in drive C has no label.
Volume Serial Number is F022-68D0

Directory of C:\Windows

12/07/2019  12:14 PM    <DIR>          TAPI
02/11/2025  11:48 AM    <DIR>          Tasks
02/11/2025  01:20 AM    <DIR>          Temp
12/07/2019  12:14 PM    <DIR>          tracing
09/07/2020  06:54 AM    <DIR>          twain_32
12/07/2019  12:10 PM                65,024 twain_32.dll
                        1 File(s)        65,024 bytes
                        5 Dir(s)  123,387,162,624 bytes free
```

Рисунок 12 Создание командного файла (.CMD)

Это упражнение помогает освоить базовые навыки работы с командной строкой Windows, включая настройку, использование команд, перенаправление вывода и создание простых скриптов. Эти навыки полезны для автоматизации задач и администрирования системы.

Упражнение 2 – Использование внешних наборов инструментов (Sysinternals).

Sysinternals Suite — это набор мощных утилит для диагностики, мониторинга и управления системой Windows. Утилиты не требуют установки, они готовы к использованию сразу после распаковки.

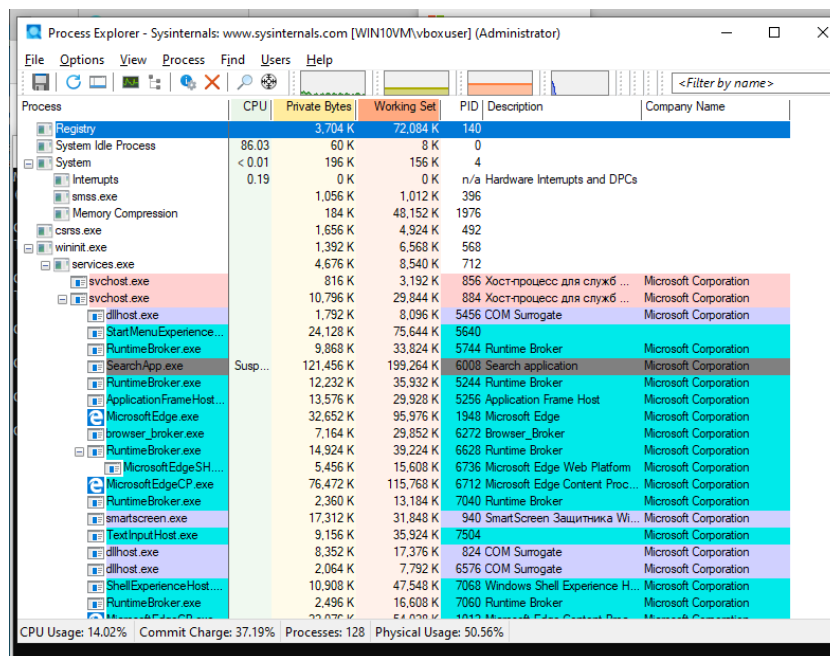


Рисунок 13

- `проsexp64.exe -t` — запускает Process Explorer в режиме дерева процессов, что полезно для анализа иерархии процессов.
- `handle.exe` — выводит список всех открытых дескрипторов в системе, что помогает диагностировать проблемы с блокировкой файлов или реестра.
- Обе утилиты являются мощными инструментами для анализа и диагностики системы, но они имеют разные области применения:
 - Process Explorer — для анализа процессов и их зависимостей.
 - Handle — для работы с дескрипторами и ресурсами.

```

C:\Users\ vboxuser\SysinternalsSuite>procexp64.exe -t

C:\Users\ vboxuser\SysinternalsSuite>handle.exe

Nthandle v5.0 - Handle viewer
Copyright (C) 1997-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

-----
System pid: 4 \NT AUTHORITY\SYSTEM
-----
smss.exe pid: 396 \<unable to open process>
-----
csrss.exe pid: 492 \<unable to open process>
-----
wininit.exe pid: 568 \<unable to open process>
-----
csrss.exe pid: 580 \<unable to open process>
-----
winlogon.exe pid: 664 NT AUTHORITY\???????
  40: File (RW-) C:\Windows\System32
 280: Section \Sessions\1\Windows\ThemeSection
 294: Section \Sessions\1\Windows\Theme4130013861
 2A8: Section \Windows\Theme1742877049
 3F0: File (R-D) C:\Windows\System32\ru-RU\user32.dll.mui
-----

```

Рисунок 14

Process Explorer показывает иерархию процессов, а Handle — открытые дескрипторы

Упражнение 3 – Запуск PowerShell. Получение справочной информации.

PowerShell — это мощный инструмент для автоматизации и управления системами. В этом упражнении мы изучим основы работы с PowerShell, включая получение справочной информации, использование командлетов, работу с переменными, массивами, хэш-таблицами и запуск скриптов.

Запуск от имени администратора необходим для выполнения команд, требующих повышенных привилегий.

Get-Command — это основной командлет для поиска доступных команд в PowerShell.

```
Выбрать Администратор: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Get-Command

CommandType      Name                                     Version      Source
-----
Alias             Add-AppPackage                         2.0.1.0      Appx
Alias             Add-AppPackageVolume                  2.0.1.0      Appx
Alias             Add-AppProvisionedPackage             3.0          Dism
Alias             Add-ProvisionedAppPackage             3.0          Dism
Alias             Add-ProvisionedAppxPackage            3.0          Dism
Alias             Add-ProvisioningPackage               3.0          Provisioning
Alias             Add-TrustedProvisioningCertificate     3.0          Provisioning
Alias             Apply-WindowsUnattend                3.0          Dism
Alias             Disable-PhysicalDiskIndication         2.0.0.0      Storage
Alias             Disable-StorageDiagnosticLog          2.0.0.0      Storage
Alias             Dismount-AppPackageVolume             2.0.1.0      Appx
Alias             Enable-PhysicalDiskIndication         2.0.0.0      Storage
Alias             Enable-StorageDiagnosticLog           2.0.0.0      Storage
Alias             Flush-Volume                         2.0.0.0      Storage
Alias             Get-AppPackage                       2.0.1.0      Appx
Alias             Get-AppPackageDefaultVolume           2.0.1.0      Appx
Alias             Get-AppPackageLastError               2.0.1.0      Appx
Alias             Get-AppPackageLog                    2.0.1.0      Appx
Alias             Get-AppPackageManifest               2.0.1.0      Appx
Alias             Get-AppPackageVolume                 2.0.1.0      Appx
Alias             Get-AppProvisionedPackage             3.0          Dism
Alias             Get-DiskSNV                          2.0.0.0      Storage
Alias             Get-PhysicalDiskSNV                  2.0.0.0      Storage
```

Рисунок 15

Шаблон *File* позволяет найти командлеты, в названии которых есть слово "File".

```
PS C:\Windows\system32> Get-Command -Name *File*

CommandType      Name                                     Version      Source
-----
Alias             Set-AppPackageProvisionedDataFile      3.0          Dism
Alias             Set-ProvisionedAppPackageDataFile     3.0          Dism
Alias             Set-ProvisionedAppxDataFile           3.0          Dism
Alias             Write-FileSystemCache                 2.0.0.0      Storage
Function          Block-FileShareAccess                 2.0.0.0      Storage
Function          Clear-FileStorageTier                 2.0.0.0      Storage
Function          Close-SmbOpenFile                     2.0.0.0      SmbShare
Function          Debug-FileShare                       2.0.0.0      Storage
Function          Disable-NetIPHttpsProfile             1.0.0.0      NetworkTransition
Function          Enable-NetIPHttpsProfile              1.0.0.0      NetworkTransition
Function          Get-FileHash                         3.1.0.0      Microsoft.PowerShell.Utility
Function          Get-FileIntegrity                    2.0.0.0      Storage
Function          Get-FileShare                        2.0.0.0      Storage
```

Рисунок 16

Параметры командлетов позволяют гибко настраивать их поведение.

```

PS C:\Windows\system32> Get-Content -Path "C:\CmdList.txt" -Encoding UTF8
For more information on a specific command, type HELP command-name
ASSOC      Displays or modifies file extension associations.
ATTRIB     Displays or changes file attributes.
BREAK      Sets or clears extended CTRL+C checking.
BCDEDIT    Sets properties in boot database to control boot loading.
CACLS      Displays or modifies access control lists (ACLs) of files.
CALL       Calls one batch program from another.
CD         Displays the name of or changes the current directory.
CHCP       Displays or sets the active code page number.
CHDIR      Displays the name of or changes the current directory.
CHKDSK     Checks a disk and displays a status report.
CHKNTFS    Displays or modifies the checking of disk at boot time.
CLS        Clears the screen.
CMD        Starts a new instance of the Windows command interpreter.
COLOR      Sets the default console foreground and background colors.
COMP       Compares the contents of two files or sets of files.
COMPACT    Displays or alters the compression of files on NTFS partitions.
CONVERT    Converts FAT volumes to NTFS. You cannot convert the
           current drive.

```

Рисунок 17

Пример выполнения командлета Get-Content с параметрами для вывода содержимого файла.

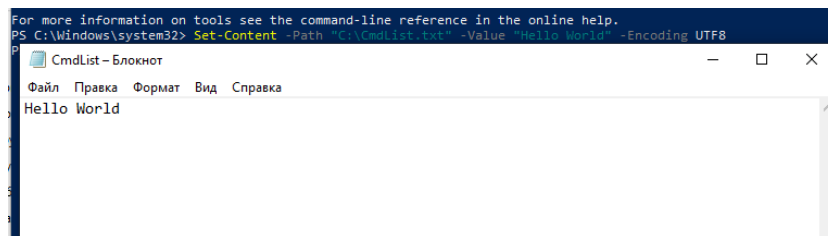


Рисунок 18

Get-Service выводит список всех сервисов на компьютере.

```

PS C:\Windows\system32> Get-Service

Status      Name                               DisplayName
-----
Stopped     AarSvc_3364c                      Agent Activation Runtime_3364c
Stopped     AJRouter                          Служба маршрутизатора AllJoyn
Stopped     ALG                                Служба шлюза уровня приложения
Stopped     AppIDSvc                          Удостоверение приложения
Running     AppInfo                           Сведения о приложении
Stopped     AppMgmt                           Управление приложениями
Stopped     AppReadiness                     Готовность приложений
Stopped     AppVClient                        Microsoft App-V Client
Running     AppXSvc                           Служба развертывания AppX (AppXSVC)
Stopped     AssignedAccessManager             Служба AssignedAccessManager
Running     AudioEndpointBuilder              Средство построения конечных точек ...
Running     Audiosrv                          Windows Audio
Stopped     autotimesvc                       Время в сети мобильной связи
Stopped     AxInstSV                          Установщик ActiveX (AxInstSV)
Stopped     BcastDVRUserService              Пользовательская служба DVR для игр...
Stopped     BDESVC                            Служба шифрования дисков BitLocker
Running     BFE                                Служба базовой фильтрации
Running     BITS                              Фоновая интеллектуальная служба пер...
Stopped     BluetoothUserService             Служба поддержки пользователей Blue...
Running     BrokerInfrastructure              Служба инфраструктуры фоновых задач
Stopped     BTAGService                       Служба звукового шлюза Bluetooth
Running     BthAvctpSvc                       Служба AVCTP
Stopped     bthserv                           Служба поддержки Bluetooth
Running     camsvc                            Служба диспетчера доступа к возможн...
Stopped     CaptureService_3364c             CaptureService_3364c
Running     cbdhsvc_3364c                    Пользовательская служба буфера обме...
Running     CDPSvc                            Служба платформы подключенных устро...

```

Рисунок 19

dir — это алиас для Get-ChildItem, который выводит содержимое текущей директории.

```
Администратор: Windows PowerShell
-a---- 12/7/2019 12:09 PM 88781 gatherNetworkInfo.vbs
-a---- 12/7/2019 12:09 PM 24006 gb2312.uce
-a---- 12/7/2019 12:08 PM 135168 gcdef.dll
-a---- 12/7/2019 12:08 PM 161880 gdi32.dll
-a---- 9/7/2020 6:50 AM 1068112 gdi32full.dll
-a---- 9/7/2020 6:50 AM 1711104 GdiPlus.dll
-a---- 12/7/2019 12:08 PM 828432 generaltel.dll
-a---- 12/7/2019 12:08 PM 673184 GenValObj.exe
-a---- 12/7/2019 12:08 PM 51200 Geocommon.dll
-a---- 9/7/2020 6:50 AM 487936 Geolocation.dll
-a---- 12/7/2019 12:09 PM 90112 getmac.exe
-a---- 12/7/2019 12:09 PM 11264 getuname.dll
-a---- 9/7/2020 6:50 AM 517120 glmf32.dll
-a---- 12/7/2019 12:08 PM 129024 globinpuhost.dll
-a---- 9/7/2020 6:50 AM 164352 glu32.dll
-a---- 12/7/2019 12:08 PM 39424 gmsaclient.dll
-a---- 9/7/2020 6:50 AM 133736 gpapi.dll
-a---- 9/7/2020 6:49 AM 78336 GPCSEWrapperCsp.dll
-a---- 12/7/2019 12:09 PM 584704 gpedit.dll
-a---- 12/7/2019 5:39 PM 147439 gpedit.msc
-a---- 9/7/2020 6:51 AM 704000 gpprefcl.dll
-a---- 12/7/2019 12:08 PM 40448 gpprnext.dll
-a---- 12/7/2019 12:09 PM 227840 gpresult.exe
-a---- 12/7/2019 5:39 PM 52736 gpscript.dll
-a---- 12/7/2019 5:39 PM 46592 gpscript.exe
-a---- 9/7/2020 6:50 AM 1296384 gpsvc.dll
-a---- 12/7/2019 12:08 PM 27136 gptext.dll
-a---- 12/7/2019 12:09 PM 30720 gpupdate.exe
-a---- 9/7/2020 6:50 AM 135168 GraphicsCapture.dll
-a---- 12/7/2019 12:08 PM 106496 GraphicsPerfSvc.dll
-a---- 12/7/2019 12:08 PM 19456 grb.rs
-a---- 12/7/2019 12:09 PM 77312 Groupinghc.dll
-a---- 12/7/2019 12:09 PM 52736 grpconv.exe
```

Рисунок 20

Get-ChildItem — это PowerShell-аналог команды dir.

```
Администратор: Windows PowerShell
-a---- 9/7/2020 6:50 AM 113664 enterpriseresourcemanager.dll
-a---- 9/7/2020 6:50 AM 148992 EoAExperiences.exe
-a---- 12/7/2019 12:09 PM 84992 eqossnap.dll
-a---- 9/7/2020 6:49 AM 202240 ErrorDetails.dll
-a---- 12/7/2019 12:08 PM 46080 ErrorDetailsCore.dll
-a---- 12/7/2019 12:08 PM 413696 es.dll
-a---- 12/7/2019 12:08 PM 20992 EsdSip.dll
-a---- 12/7/2019 12:08 PM 3280384 esent.dll
-a---- 12/7/2019 12:09 PM 65536 esentprf.dll
-a---- 12/7/2019 12:09 PM 409600 esentutl.exe
-a---- 12/7/2019 12:09 PM 38400 esevss.dll
-a---- 12/7/2019 12:08 PM 145920 eShims.dll
-a---- 12/7/2019 12:08 PM 33792 esrb.rs
-a---- 12/7/2019 12:08 PM 192000 EthernetMediaManager.dll
-a---- 12/7/2019 12:08 PM 2560 ETWCoreUIComponentsResources.dll
-a---- 12/7/2019 12:08 PM 88064 ETWSEProviderResources.dll
-a---- 12/7/2019 12:08 PM 51200 EtwRunDown.dll
-a---- 12/7/2019 12:09 PM 374272 eudcredit.exe
-a---- 12/7/2019 12:09 PM 137216 eUICCsCSP.dll
-a---- 12/7/2019 12:08 PM 80384 EventAggregation.dll
-a---- 12/7/2019 12:09 PM 17920 eventcls.dll
-a---- 12/7/2019 12:09 PM 44544 eventcreate.exe
-a---- 12/7/2019 12:09 PM 17935 EventViewer_EventDetails.xml
-a---- 12/7/2019 12:09 PM 84480 eventvwr.exe
-a---- 12/7/2019 12:09 PM 145127 eventvwr.msc
-a---- 12/7/2019 5:39 PM 773416 evr.dll
-a---- 9/7/2020 6:49 AM 353840 ExecModelClient.dll
-a---- 12/7/2019 12:08 PM 80896 execmodelproxy.dll
-a---- 12/7/2019 12:08 PM 67584 expand.exe
-a---- 9/7/2020 6:50 AM 2206208 ExplorerFrame.dll
-a---- 12/7/2019 12:08 PM 267776 ExSMime.dll
-a---- 12/7/2019 12:08 PM 35328 extrac32.exe
-a---- 12/7/2019 12:08 PM 24576 ExtrasXmlParser.dll
-a---- 12/7/2019 12:08 PM 8704 f3ahvoas.dll
-a---- 9/7/2020 6:50 AM 592896 facecredentialprovider.dll
-a---- 9/7/2020 6:50 AM 994616 Facilitator.dll
-a---- 9/7/2020 6:50 AM 103424 Family.Authentication.dll
-a---- 12/7/2019 12:09 PM 161792 Family.Cache.dll
-a---- 9/7/2020 6:50 AM 151552 Family.Client.dll
```

Рисунок 21

Запуск командлета Get-Process с указанием имени текущего компьютера.

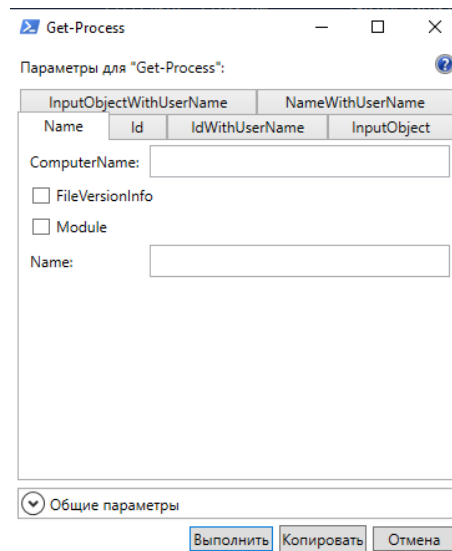


Рисунок 22

Эта команда выводит список процессов на текущем компьютере.

```
PS C:\Windows\system32> Get-Process -ComputerName $env:COMPUTERNAME
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
393	23	13036	30540	0.44	5256	1	ApplicationFrameHos
473	22	7044	29792	0.45	6272	1	browser_broker
260	14	5244	20060	9.16	3508	1	conhost
451	19	1744	4916	0.56	492	0	csrss
579	20	1944	6048	11.64	580	1	csrss
559	17	7652	22664	2.66	4144	1	ctfmon
309	25	6988	16484	0.36	824	1	dllhost
136	8	1772	8240	0.05	5456	1	dllhost
141	9	2144	7484	0.20	6576	1	dllhost
989	64	54916	81292	16.67	788	1	dwm
2388	103	55544	134468	41.42	4408	1	explorer
39	8	3524	7512	0.33	876	1	fontdrvhost
39	5	1272	3260	0.00	892	0	fontdrvhost
0	0	60	8		0	0	Idle
1570	25	8336	19944	2.16	720	0	lsass

Рисунок 23

Этот командлет выводит информацию о всех IP-адресах, назначенных компьютеру.

```

PS C:\Windows\system32> Get-NetIPAddress

IPAddress      : fe80::a4b5:96e0:750d:44b8%6
InterfaceIndex : 6
InterfaceAlias  : Ethernet
AddressFamily   : IPv6
Type            : Unicast
PrefixLength    : 64
PrefixOrigin    : WellKnown
SuffixOrigin    : Link
AddressState    : Preferred
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : ::1
InterfaceIndex : 1
InterfaceAlias  : Loopback Pseudo-Interface 1
AddressFamily   : IPv6
Type            : Unicast
PrefixLength    : 128
PrefixOrigin    : WellKnown
SuffixOrigin    : WellKnown
AddressState    : Preferred
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : 10.0.2.15
InterfaceIndex : 6
InterfaceAlias  : Ethernet
AddressFamily   : IPv4
Type            : Unicast
PrefixLength    : 24
PrefixOrigin    : Dhcp
SuffixOrigin    : Dhcp
AddressState    : Preferred
ValidLifetime   : 23:20:48
PreferredLifetime : 23:20:48
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : 127.0.0.1
InterfaceIndex : 1

```

Рисунок 24

Символ > перенаправляет вывод команды в файл.

Proc-info – Блокнот

Ид	Правка	Формат	Вид	Справка			
Idles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
393	23	13008	30528	0.44	5256	1	ApplicationFrameHost
473	22	7044	29792	0.45	6272	1	browser_broker
260	13	5212	20044	9.80	3508	1	conhost
447	19	1740	4912	0.56	492	0	csrss
580	21	1944	6048	11.94	580	1	csrss
559	17	7624	22652	2.66	4144	1	ctfmon
309	25	6932	16472	0.36	824	1	dllhost
136	8	1772	8240	0.05	5456	1	dllhost
141	9	2064	7460	0.20	6576	1	dllhost
991	64	55016	81296	16.95	788	1	dwm
2376	102	54936	134784	41.53	4408	1	explorer
39	8	3524	7512	0.33	876	1	fontdrvhost
39	5	1272	3260	0.00	892	0	fontdrvhost
0	0	60	8		0	0	Idle
1563	25	8336	19936	2.19	720	0	lsass
0	0	212	51256	1.14	1976	0	Memory Compression
1364	96	32012	95724	4.63	1948	1	MicrosoftEdge
867	41	21984	53912	0.59	1012	1	MicrosoftEdgeCP
1473	98	82464	132984	920.55	1300	1	MicrosoftEdgeCP
932	51	38672	83260	2.94	1360	1	MicrosoftEdgeCP
502	24	6032	27784	0.11	2616	1	MicrosoftEdgeCP

Рисунок 25

Результат выполнения команды Get-Process будет сохранен в файл Proc-Info.txt.

```
PS C:\Windows\system32> Get-Process | Sort-Object -property ProcessName
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
393	23	13008	30500	0.44	5256	1	ApplicationFrameHost
473	22	7044	29620	0.45	6272	1	browser_broker
260	13	5212	20044	9.98	3508	1	conhost
588	21	1944	6048	12.41	580	1	csrss
450	19	1740	4908	0.56	492	0	csrss
563	17	7640	21688	3.11	4144	1	ctfmon
141	9	2064	7448	0.20	6576	1	dllhost
136	8	1824	8252	0.05	5456	1	dllhost
309	25	6932	16000	0.36	824	1	dllhost
993	64	57552	82476	17.52	788	1	dwm
2451	105	57732	135868	42.47	4408	1	explorer
39	5	1272	3148	0.00	892	0	fontdrvhost

Рисунок 26

Конвейер (|) передает вывод одной команды на вход другой. В данном случае процессы сортируются по имени.

```
PS C:\Windows\system32> Get-Process | Measure-Object
```

Count	: 125
Average	:
Sum	:
Maximum	:
Minimum	:
Property	:

Рисунок 27


```
PS C:\Windows\system32> Get-Process | Where-Object {$_.ProcessName -like "M*"}
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
0 0 216 57356 1.38 1976 0 Memory Compression
1366 96 32044 95400 4.63 1948 1 MicrosoftEdge
867 41 21984 53912 0.59 1012 1 MicrosoftEdgeCP
1473 98 82684 117448 1,040.38 1300 1 MicrosoftEdgeCP
932 51 38672 68792 2.94 1360 1 MicrosoftEdgeCP
504 24 6060 26852 0.11 2616 1 MicrosoftEdgeCP
1214 71 76244 105892 6.77 6712 1 MicrosoftEdgeCP
1169 200 206396 250256 4.31 8160 1 MicrosoftEdgeCP
495 23 6028 26644 0.13 8252 1 MicrosoftEdgeCP
303 17 5348 15484 2.39 6736 1 MicrosoftEdgeSH
540 62 144576 113208 41.94 2020 0 MsMpEng
```

Рисунок 28

Эта команда выводит только процессы, имена которых начинаются на букву "М".

```
PS C:\Windows\system32> $i = Get-Process | Where-Object {$_.ProcessName -like "M*"}
PS C:\Windows\system32> $i
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
0 0 216 51924 1.39 1976 0 Memory Compression
1358 96 31972 95356 4.63 1948 1 MicrosoftEdge
867 41 21984 53912 0.59 1012 1 MicrosoftEdgeCP
1467 97 86600 121368 1,111.81 1300 1 MicrosoftEdgeCP
928 51 38640 68768 2.94 1360 1 MicrosoftEdgeCP
500 24 5968 26800 0.11 2616 1 MicrosoftEdgeCP
1210 70 76188 105884 6.77 6712 1 MicrosoftEdgeCP
1169 200 206396 250256 4.31 8160 1 MicrosoftEdgeCP
493 23 5964 26608 0.13 8252 1 MicrosoftEdgeCP
303 17 5348 15484 2.39 6736 1 MicrosoftEdgeSH
568 63 146664 126472 42.22 2020 0 MsMpEng
```

Рисунок 29

Результат выполнения команды сохраняется в переменную \$i. Далее вывели значение переменной

```
PS C:\Windows\system32> $i = 9
PS C:\Windows\system32> $i
9
PS C:\Windows\system32> $i.GetType()

IsPublic IsSerial Name BaseType
-----
True True Int32 System.ValueType
```

Рисунок 30

Создали переменную, проверили ее значение, проверили тип переменной.

```
PS C:\Windows\system32> $i = "stroka"
PS C:\Windows\system32> $i.GetType()

IsPublic IsSerial Name BaseType
-----
True True String System.Object

PS C:\Windows\system32> $i.lenght
PS C:\Windows\system32> $i.Lenght
PS C:\Windows\system32> $i.Length
6
```

Рисунок 31

Изменили тип и снова проверили

```
PS C:\Windows\system32> $i
stroka
PS C:\Windows\system32> $I
stroka
PS C:\Windows\system32> $i -eq $I
True
```

Рисунок 33

PowerShell автоматически определяет тип переменной на основе её значения.

```
PS C:\Windows\system32> [int]$k = 5
PS C:\Windows\system32> $k = "stroka"
Cannot convert value "stroka" to type "System.Int32". Error: "Input string was not in a correct format."
At line:1 char:1
+ $k = "stroka"
+ ~~~~~
+ CategoryInfo          : MetadataError: (:) [], ArgumentTransformationMetadataException
+ FullyQualifiedErrorId : RuntimeException
```

Рисунок 34

```
PS C:\Windows\system32> [string]$s = $k
PS C:\Windows\system32> $s
5
PS C:\Windows\system32> $s -eq $k
True
```

Рисунок 35

Типизированные переменные не позволяют присваивать значения другого типа.

```
PS C:\Windows\system32> $m = "Stroka1", "Stroka2", "sTROKA3", "Stroka4", "Stroka5"
PS C:\Windows\system32> $m[0]
Stroka1
PS C:\Windows\system32> $m += "Stroka6"
PS C:\Windows\system32> $m
Stroka1
Stroka2
sTROKA3
Stroka4
Stroka5
Stroka6
PS C:\Windows\system32> $m = $m | Where-Object {$_ -ne $m[1]}
PS C:\Windows\system32> $m
Stroka1
sTROKA3
Stroka4
Stroka5
Stroka6
```

Рисунок 36

Массивы в PowerShell динамические и могут изменяться.

Далее идет работа со списками

```

PS C:\Windows\system32> $ArrList = New-Object System.Collections.ArrayList
PS C:\Windows\system32> $ArrList.Add(@("sTROKA1", "Stroka2"))
0
PS C:\Windows\system32> $ArrList.Add(@("sTROKA3", "Stroka4"))
1
PS C:\Windows\system32> $ArrList.Add(@("sTROKA5", "Stroka6"))
2
PS C:\Windows\system32> $ArrList.Add(@("sTROKA7", "Stroka8"))
3
PS C:\Windows\system32> $ArrList[$ArrList.Count -1]
sTROKA7
Stroka8
PS C:\Windows\system32> $ArrList.RemoveAt(0)
PS C:\Windows\system32> $ArrList
sTROKA3
Stroka4
sTROKA5
Stroka6
sTROKA7
Stroka8
PS C:\Windows\system32> $ArrList.Remove(@("sTROKA3", "Stroka4"))
PS C:\Windows\system32> $ArrList
sTROKA5
Stroka6
sTROKA7
Stroka8

```

Рисунок 37

```

PS C:\Windows\system32> Get-Service | Where-Object {$_.Name -like "S*"} | ForeACH-Object {$ArrList.Add($_.Name)}
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

Рисунок 38

```

PS C:\Windows\system32> $ArrList
SamSs
SCardSvr
ScDeviceEnum
Schedule
SCPolicySvc
SDRSVC
seclogon
SecurityHealthService
SEMgrSvc
SENS
Sense
SensorDataService
SensorService
SensrSvc
SessionEnv
SgrmBroker
SharedAccess
SharedRealitySvc
ShellHWDetection
shpamsvc
smphost
SmsRouter
SNMPTRAP
spectrum
Spooler
sppsvc
SDPSRV
ssh-agent
SstpSvc
StateRepository
stisvc
StorSvc
svsvc
swprv
SysMain
SystemEventsBroker

```

Рисунок 39

Списки массивов позволяют работать с несколькими массивами одновременно.

Хэш-таблицы полезны для хранения пар ключ-значение.

```

PS C:\Windows\system32> $HashTable = @{
>> Name = "Ksenia";
>> Age = 19;
>> City = "SPB"
>> }
PS C:\Windows\system32> $HashTable.Occupation = "Programmer"
PS C:\Windows\system32> $HashTable.Remove("City")
PS C:\Windows\system32> $HashTable.Name = "Ksenia"
PS C:\Windows\system32> $HashTable

Name                Value
-----
Occupation           Programmer
Name                 Ksenia
Age                  19

```

Рисунок 40

Переменные, созданные в одном экземпляре PowerShell, не видны в другом.

```

PS C:\Windows\system32> Get-ExecutionPolicy
Restricted

```

Рисунок 41



Рисунок 42

При двойном клике открывается блокнот

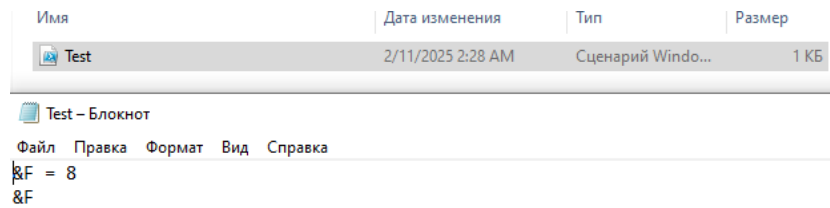


Рисунок 43

Если открывать через «Выполнить с помощью PowerShell», то открывает PowerShell, и там ничего не выводится.

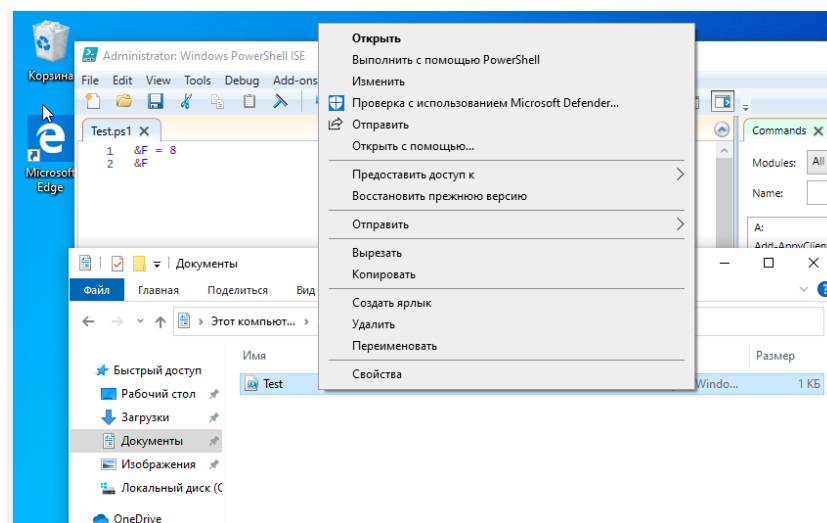


Рисунок 44

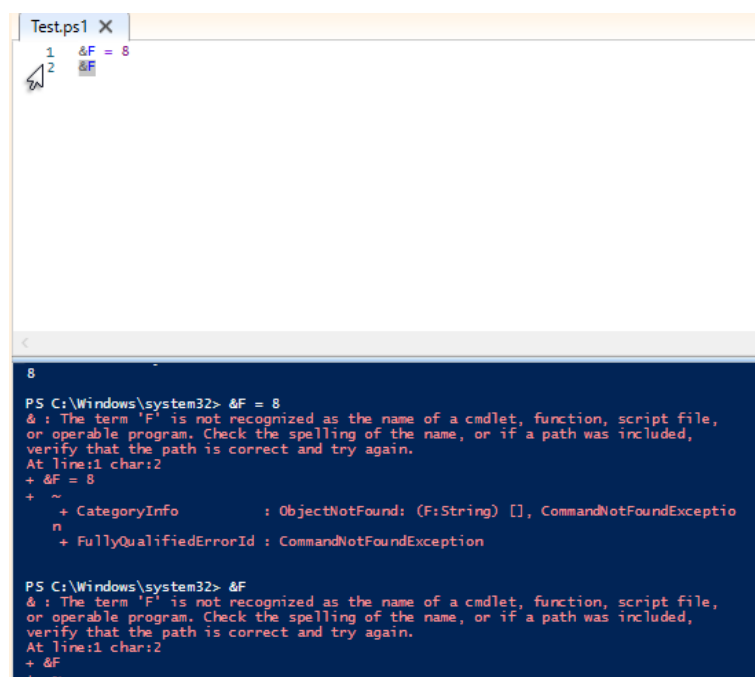


Рисунок 45

Выполнение построчно

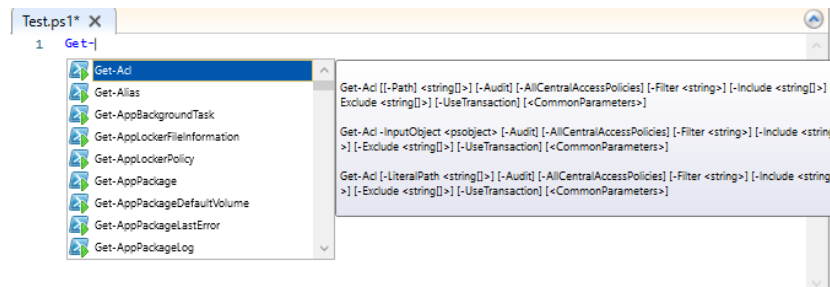


Рисунок 46 Автоподсказки

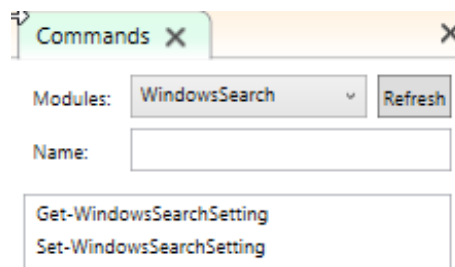


Рисунок 47 Поиск команд по типам

Выводы

- Командная строка Windows — это базовый, но мощный инструмент для выполнения задач управления системой, который остается актуальным даже с появлением более современных инструментов.
- PowerShell превосходит командную строку по функциональности, предоставляя более гибкие и мощные средства для автоматизации, управления системами и обработки данных.
- Sysinternals Suite — это незаменимый набор утилит для системных администраторов, который расширяет возможности стандартных инструментов Windows, позволяя решать сложные задачи диагностики и управления.
- Автоматизация задач — ключевое преимущество изученных инструментов. Командная строка, PowerShell и Sysinternals позволяют автоматизировать рутинные процессы, что экономит время и уменьшает вероятность ошибок.
- Конвейер (Pipeline) в PowerShell — это мощный механизм, который позволяет объединять командлеты для обработки данных, что делает выполнение задач более эффективным.
- Переменные, массивы и хэш-таблицы в PowerShell предоставляют гибкость для работы с данными, что особенно полезно при создании сложных скриптов.
- Интеллектуальная подсказка в PowerShell ISE ускоряет написание кода и уменьшает количество ошибок, делая процесс разработки более удобным.
- Отладка скриптов в PowerShell ISE позволяет находить и исправлять ошибки, что особенно важно при работе с большими и сложными скриптами.
- Документация и справка — важная часть работы с любым инструментом. Команды help в командной строке и Get-Help в PowerShell предоставляют подробную информацию о командах и их параметрах.
- Создание скриптов — это эффективный способ автоматизации задач. Скрипты можно сохранять, повторно использовать и передавать другим пользователям.
- Работа с файлами и процессами — одна из ключевых возможностей PowerShell. Командлеты, такие как Get-Content, Copy-Item, Get-Process, позволяют легко управлять файлами и процессами.
- Гибкость PowerShell позволяет адаптировать его для решения самых разных задач, от простых (например, вывод списка файлов) до сложных (например, генерация отчетов или управление сетевыми ресурсами).
- PowerShell ISE — это идеальная среда для разработки скриптов, которая сочетает в себе удобство графического интерфейса и мощь PowerShell.