Лабораторная работа №4

Изучение основных возможностей WindowsPowerShell

Цель работы:научиться использованию возможностей WindowsPowerShell для выполнения задач администрирования в ОС Windows.

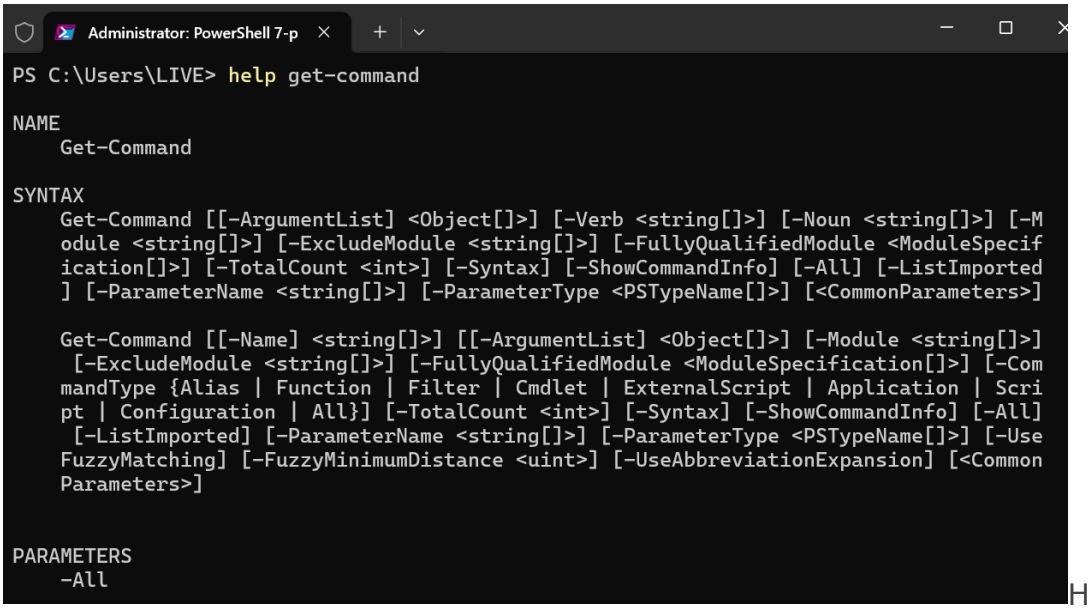
2. Практическое задание

2.1.Запуск среды WindowsPowerShell

Запуск среды WindowsPowerShell осуществляется следующим образом **Пуск ->Все Программы** и выбрать элемент **WindowsPowerShell**. Другой вариант запуска оболочки - пункт **Выполнить** в меню Пуск, ввести имя файла PowerShell и нажать кнопку ОК.

1. Выполним первую команду в PowerShell. Для первого знакомства с WindowsPowerShell вполне подойдет команда help. Внимательно изучите выведенную на экран информацию.

   Исполните из командной строки:



HelpGet-Command

   В результате выполнения этой команды мы получим полное описание команды Get-Command, включая ее назначение, синтаксис, опции и т.п.

2. Выполните команду:

   Get-Command

На экран будет выведен список всех встроенных команд.

```
PS C:\Users\LIVE> 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-MsixPackage                                    2.0.1.0    Appx
Alias           Add-MsixPackageVolume                              2.0.1.0    Appx
Alias           Add-MsixVolume                                     2.0.1.0    Appx
Alias           Add-ProvisionedAppPackage                          3.0        Dism
Alias           Add-ProvisionedAppSharedPackageContainer           3.0        Dism
Alias           Add-ProvisionedAppxPackage                         3.0        Dism
Alias           Add-ProvisioningPackage                            3.0        Provisi…
Alias           Add-TrustedProvisioningCertificate                 3.0        Provisi…
Alias           Apply-WindowsUnattend                              3.0        Dism
Alias           Disable-PhysicalDiskIndication                     2.0.0.0    Storage
Alias           Disable-PhysicalDiskIndication                     1.0.0.0    VMDirec…
Alias           Disable-StorageDiagnosticLog                       2.0.0.0    Storage
Alias           Disable-StorageDiagnosticLog                       1.0.0.0    VMDirec…
```

3. Просмотрите список всех сервисов, запущенных на вашем компьютере, исполнив команду:

   Get-Service

   Команда Get-Service служит для получения списка всех сервисов, запущенных на данном

```
Administrator: PowerShell 7-p
PS C:\Users\LIVE> get-service

Status   Name               DisplayName
------   ----               -----------
Stopped  AarSvc_3c43d4d     Agent Activation Runtime_3c43d4d
Stopped  AdobeARMservice    Adobe Acrobat Update Service
Stopped  ADPSvc             Служба платформы агрегированных данных
Stopped  ALG                Служба шлюза уровня приложения
Running  AMD Crash Defende… AMD Crash Defender Service
Running  AMD External Even… AMD External Events Utility
Running  amdpmfservice      AMD PMF Service
Running  AmdPpkgSvc         AMD Provisioning Packages Service
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  ApxSvc             Прокси-служба виртуального звукового …
```

   компьютере.

4. Просмотрите список всех процессов, запущенных в настоящий момент на вашем компьютере,исполнив команду:

```
Administrator: PowerShell 7-p
PS C:\Users\LIVE> get-process

 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
     13     3,27       2,97       3,75    3200   0 amdfendrsr
     13     2,71       2,29       3,59   42852   3 amdow
     11     2,46       3,08       3,78    2080   0 amdpmfservice
      9     1,72       2,67      50,81   63084   3 amdpmfserviceuser
      9     1,82       2,25       3,75    5516   0 AmdPpkgSvc
     22     8,23       7,22     172,19   39616   3 AMDRSServ
     51   129,45       7,19       4,34   45140   3 AMDRSSrcExt
     21    13,85       8,65       7,52   47416   3 AppActions
     20     7,72       4,64       4,94    6164   3 ApplicationFrameHost
     20     6,13       9,07     112,95   68196   3 atieclxx
     16     4,83       3,19       4,98    3192   0 atiesrxx
     17    19,11      23,78     911,97   75652   0 audiodg
     13     3,25       7,56       3,86   10244   0 AUEPDU
     19     3,89      10,55     149,83    9492   0 AUEPMaster
     20    21,19      58,39       0,59   10808   3 backgroundTaskHost
     30    18,46       1,84       0,48   77496   3 backgroundTaskHost
```

   Get-Process

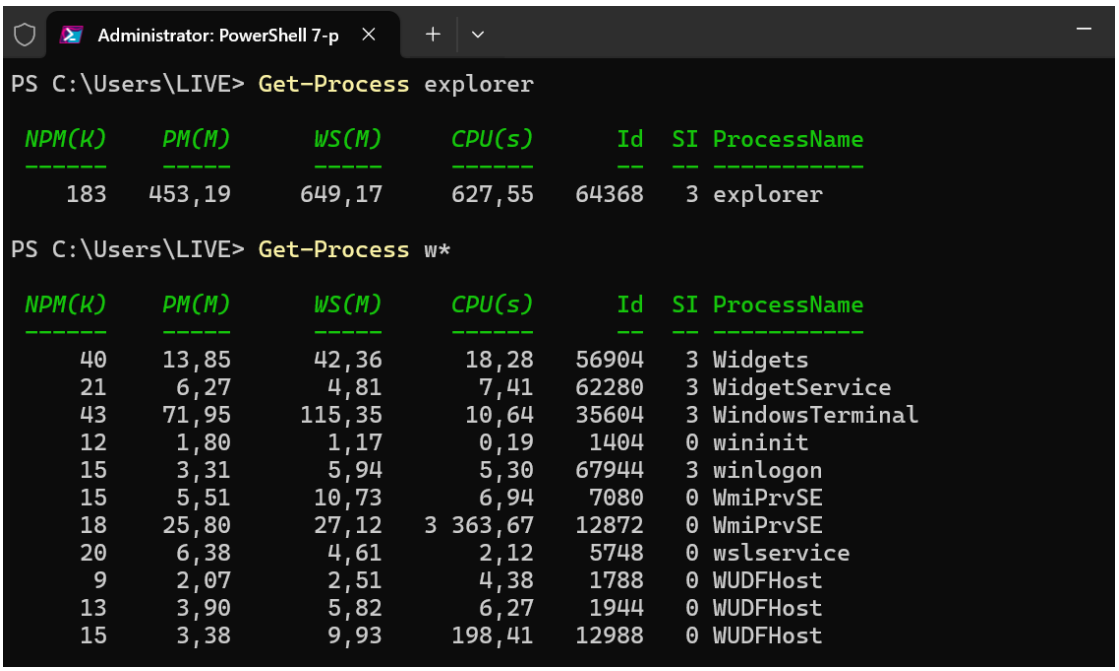
5. Для получения информации только об одном процессе в качестве аргумента команды Get-Process задается имя этого процесса. Выполните команду:

   Get-Processexplorer

   Из командной строки исполните команду:

   Get-Process w*

На экран должна быть выведена информация обо всех запущенных процессах, начинающихся на символ w.



6. По умолчанию информация выводится в виде таблицы, но на самом деле все команды возвращают объекты. Эти объекты могут быть переданы на вход другим командам с помощью символа «|».

   Исполните команду:

   Get-Process i* | format-list

   Объекты будут отформатированы в виде списка. Теперь список процессов доступен в другом представлении. Для получения подробной информации о различных форматах можно использовать следующую команду:

   Helpformat*

   Другие возможные форматы:

   Get-Process i* | format-wide

   и

Get-Process i* | format-custom

```
Administrator: PowerShell 7-p
PS C:\Users\LIVE> Get-Process i* | format-list

Id      : 0
Handles : 0
CPU     :
SI      : 0
Name    : Idle

Id      : 5560
Handles : 296
CPU     : 1,34375
SI      : 0
Name    : IpOverUsbSvc

PS C:\Users\LIVE> Get-Process i* | format-wide

Idle                                    IpOverUsbSvc
```

7. Выполняя команды, мы всегда получаем объекты, а у объектовесть свойства. Просмотрите все свойства объекта, полученногопривыполнениикомандыGet-Processиспользуяследующую команду:

   Get-Process | Get-Member

```
Administrator: PowerShell 7-p
PS C:\Users\LIVE> Get-Process | Get-Member

   TypeName: System.Diagnostics.Process

Name                       MemberType     Definition
----                       ----------     ----------
Handles                    AliasProperty  Handles = Handlecount
Name                       AliasProperty  Name = ProcessName
NPM                        AliasProperty  NPM = NonpagedSystemMemorySize64
PM                         AliasProperty  PM = PagedMemorySize64
SI                         AliasProperty  SI = SessionId
VM                         AliasProperty  VM = VirtualMemorySize64
WS                         AliasProperty  WS = WorkingSet64
Parent                     CodeProperty   System.Object Parent{get=GetParentProcess;}
Disposed                   Event          System.EventHandler Disposed(System.Object,…
ErrorDataReceived          Event          System.Diagnostics.DataReceivedEventHandler…
Exited                     Event          System.EventHandler Exited(System.Object, S…
OutputDataReceived         Event          System.Diagnostics.DataReceivedEventHandler…
BeginErrorReadLine         Method         void BeginErrorReadLine()
BeginOutputReadLine        Method         void BeginOutputReadLine()
CancelErrorRead            Method         void CancelErrorRead()
CancelOutputRead           Method         void CancelOutputRead()
```

8. Поскольку на выходе всегда получается объект, можно манипулировать им для выполнения дополнительных операций. Выполните операцию фильтрации, исполнив команду:

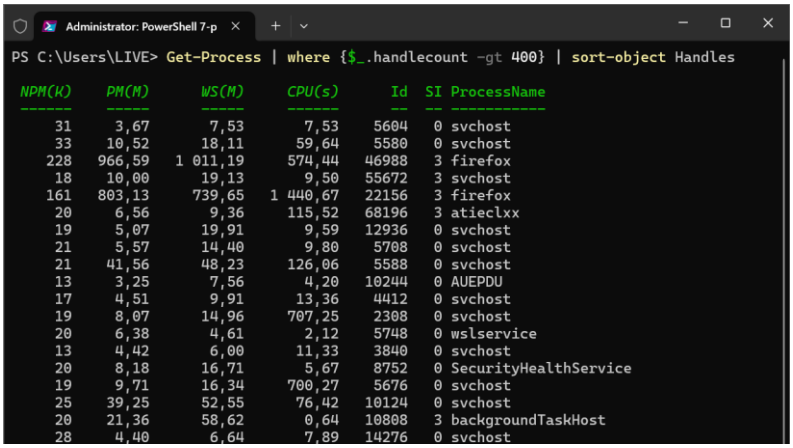
```
Administrator: PowerShell 7-p
PS C:\Users\LIVE> Get-Process | where {$_.handlecount -gt 400}

 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
     22     8,23       7,22     172,77   39616   3 AMDRSServ
     51   129,45       7,19       4,34   45140   3 AMDRSSrcExt
     20     6,16       9,09     113,97   68196   3 atieclxx
     18    19,18      23,83     916,55   75652   0 audiodg
     13     3,25       7,56       3,91   10244   0 AUEPDU
     19     3,89      10,55     149,98    9492   0 AUEPMaster
     19    21,15      58,38       0,59   10808   3 backgroundTaskHost
     34    32,39       8,86       7,69   59012   3 CrossDeviceResume
     30     2,58       2,81       7,53    1196   0 csrss
     25    15,68      16,19     155,42   60040   3 csrss
     23    18,51      24,69     128,78   51724   3 ctfmon
     68   232,70     167,39   2 451,61   77768   3 dwm
    184   454,19     649,84     633,55   64368   3 explorer
    212   433,96     415,26   3 471,59   15776   3 firefox
```

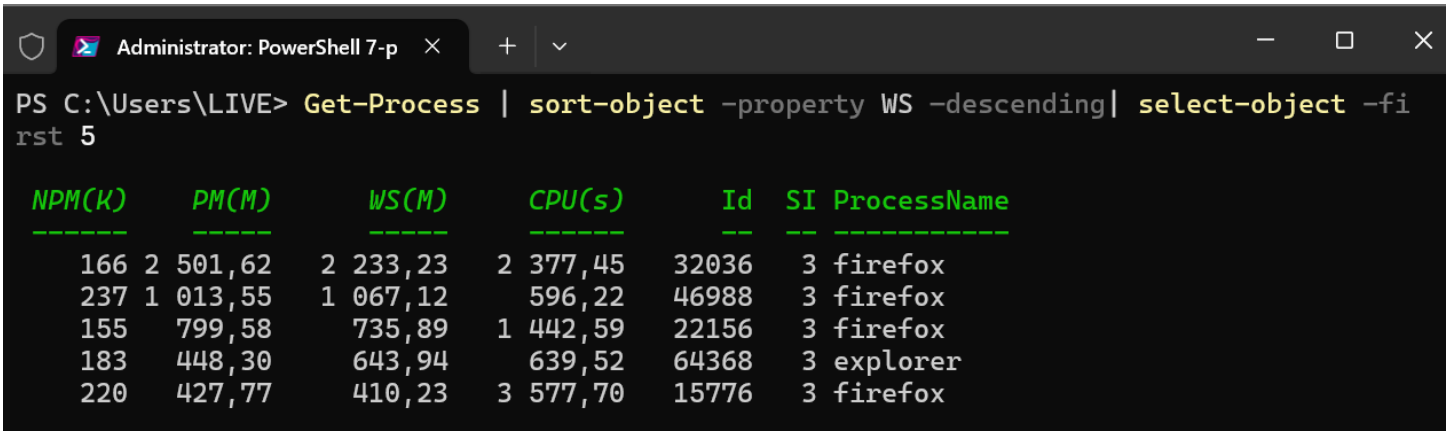
Get-Process | where {$_.handlecount -gt 400}

Выполните операцию сортировки, исполнив команду:

Get-Process | where {$_.handlecount -gt 400} | sort-object Handles



9. Произведем сортировку объектов по свойству WS (workingset) и выбор 5 процессов, занимающих больше всего памяти



Get-Process | sort-object -property WS –descending| select-object -first 5

10. Команда stop-process позволяет остановить запущенный процесс.

Запустите Notepad. Выполнитекоманду:

Get-process notepad | stop-process

Окно Блокнота закроется. Снова запустите Notepad.

Такая возможность не всегда является безопасной, поэтому лучше использовать подобные команды с опцией whatif, которая показывает, что произойдет при выполнении той или иной команды, но на самом деле команда не выполняется:

Get-Process notepad | stop-process -whatif

Кроме того, можно указывать на необходимость подтверждения перед выполнением команды:

Get-Process notepad | stop-process -confirm

В последнем примере мы получаем описание действий, которые выполняет команда, и можем выбрать, подтверждать ее выполнение или нет.

```
PS C:\Users\LIVE> Get-Process notepad | stop-process -whatif
Get-Process: Cannot find a process with the name "notepad". Verify the process name and
 call the cmdlet again.
PS C:\Users\LIVE> Get-Process notepad | stop-process -whatif
What if: Performing the operation "Stop-Process" on target "Notepad (54156)".
PS C:\Users\LIVE> Get-Process notepad | stop-process -confirm

Confirm
Are you sure you want to perform this action?
Performing the operation "Stop-Process" on target "Notepad (50116)".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "Y"):y
PS C:\Users\LIVE>
```

2.2. Работа с файловой системой

```
PS C:\Users\LIVE> New-Item psdemo.txt -ItemType File

    Directory: C:\Users\LIVE

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 psdemo.txt

PS C:\Users\LIVE> New-Item 1.txt, 2.txt -ItemType File

    Directory: C:\Users\LIVE

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 1.txt
-a---          14.12.2025     1:22              0 2.txt

PS C:\Users\LIVE> gci *.txt

    Directory: C:\Users\LIVE

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 1.txt
-a---          14.12.2025     1:22              0 2.txt
-a---          14.12.2025     1:22              0 psdemo.txt

PS C:\Users\LIVE>
```

Разберем основные команды WindowsPowerShell, применяемые для манипуляций с файловой системой: new-item, copy-item, move-item, rename-item и remove-item.

11. Создадим новый подкаталог TextFiles в текущем каталоге:

    new-itemTextFiles -itemtype directory

    Если опустить параметр -itemtype,то WindowsPowerShell спросит, что мы создаем — файл (file) или каталог (directory). Отметим,что у команды new-item есть алиас — ni.В сокращенном виде наша первая команда будет выглядеть так:

niTextFiles -itemtype directory

```
PS C:\Users\LIVE> copy *.txt .\TextFiles\
PS C:\Users\LIVE> sl TextFiles
PS C:\Users\LIVE\TextFiles> gci

    Directory: C:\Users\LIVE\TextFiles

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 1.txt
-a---          14.12.2025     1:22              0 2.txt
-a---          14.12.2025     1:22              0 psdemo.txt

PS C:\Users\LIVE\TextFiles>
```

12. Создайте несколько новых файлов в текущем каталоге: psdemo.txt, 1.txt, 2.txt:
13. Скопируем все файлы с расширением *.txt в подкаталог TextFiles, используя команду copy-item (алиасы — cpi, cp, copy):

    Если применять данную команду в пакетном файле, имеет смысл сделать ее более понятной, указав параметры -path (источник) и -destination (приемник):

    copy-item -path '.*.txt' -destination '.\TextFiles'
14. После выполнения команды копирования мы используем команду set-location для перехода в подкаталог TextFiles.

    set-locationTextFiles
15. С помощью команды rename-item переименовываем файл psdemo.txt в psdemo.bak. При необходимости можно применять опции -path и -newName:

    rename-item psdemo.txt psdemo.bak
16. После того как файл переименован, переносим его на один уровень вверх, используя команду move-item:

    move-itempsdemo.bak ..\
17. Затем применяем команду set-location, а точнее — ее алиас sl для перехода в другой каталог:

    sl ..
18. Манипуляции с файловой системой мы завершаем удалением всего каталога TextFiles, используя команду remove-item. Поскольку в каталоге TextFiles содержатся файлы, применяется опция -recurse. Если эта опция не указана, WindowsPowerShell запросит подтверждение перед выполнением команды remove-item.

    remove-itemTextFiles–recurse

```
PS C:\Users\LIVE> copy *.txt .\TextFiles\
PS C:\Users\LIVE> sl TextFiles
PS C:\Users\LIVE\TextFiles> gci

    Directory: C:\Users\LIVE\TextFiles

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 1.txt
-a---          14.12.2025     1:22              0 2.txt
-a---          14.12.2025     1:22              0 psdemo.txt

PS C:\Users\LIVE\TextFiles> ren psdemo.txt psdemo.bak
PS C:\Users\LIVE\TextFiles> mi psdemo.bak ..\
PS C:\Users\LIVE\TextFiles> sl ..
PS C:\Users\LIVE> gci psdemo.bak

    Directory: C:\Users\LIVE

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14.12.2025     1:22              0 psdemo.bak

PS C:\Users\LIVE> ri TextFiles -Recurse
PS C:\Users\LIVE> gci

    Directory: C:\Users\LIVE

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          08.11.2025    13:51                .android
d----          15.11.2025    17:59                .config
d----          10.12.2025    15:10                .docker
d----          14.11.2025     0:14                .dotnet
d----          10.12.2025    15:10                .gk
d----          21.09.2025    13:08                .mputils
```