

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

В. В. Жукалин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5

РАЗРАБОТКА КОМАНДЛЕТОВ POWERSHELL

по курсу:

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4326

подпись, дата

Г. С. Томчук

инициалы, фамилия

Санкт-Петербург 2025

1 Цель работы

Цель работы: изучение командлетов PowerShell.

Работа выполнялась по варианту № 19.

2 Выполненные упражнения

2.1 Упражнение 2.5

На рис. 1, 2 изображен результат выполнения упражнения 2.5.

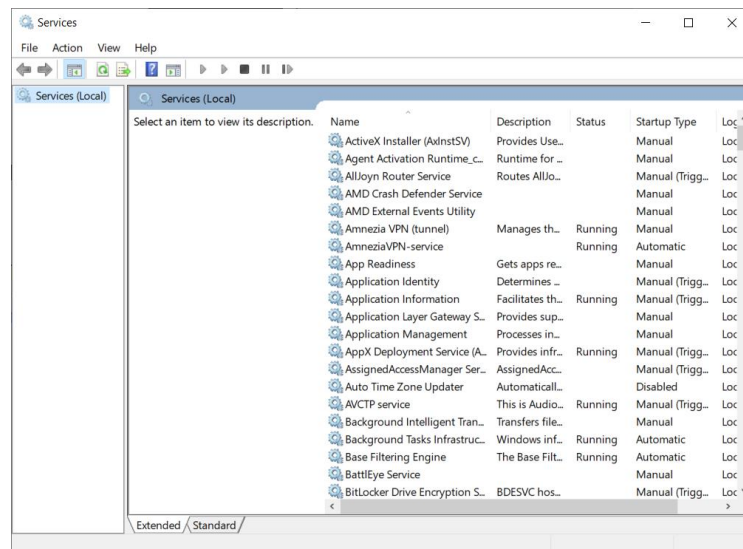


Рисунок 1 – Окно служб Windows

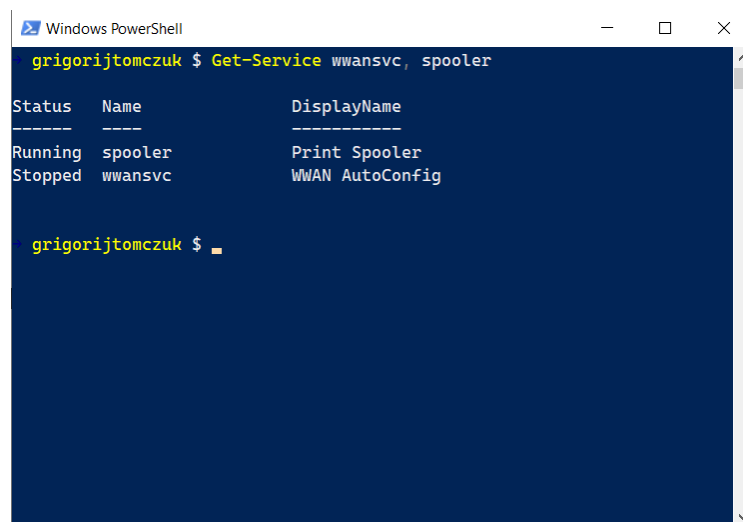
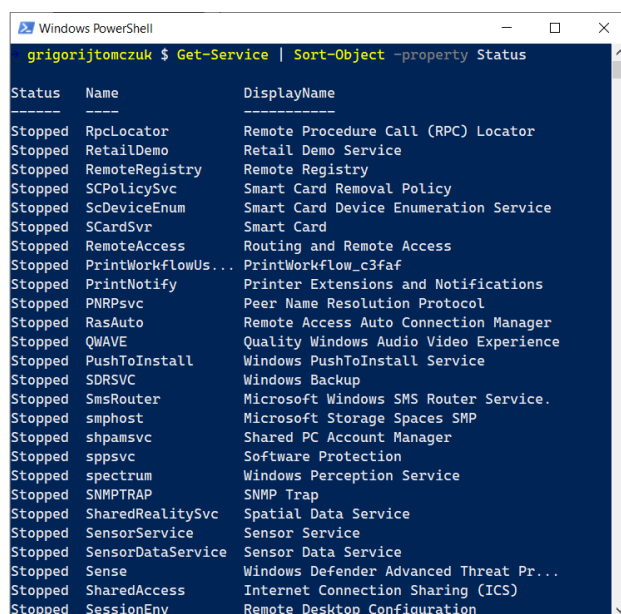


Рисунок 2 – Результат выполнения команды `Get-Service wwansvc, spooler`

2.2 Упражнение 2.6

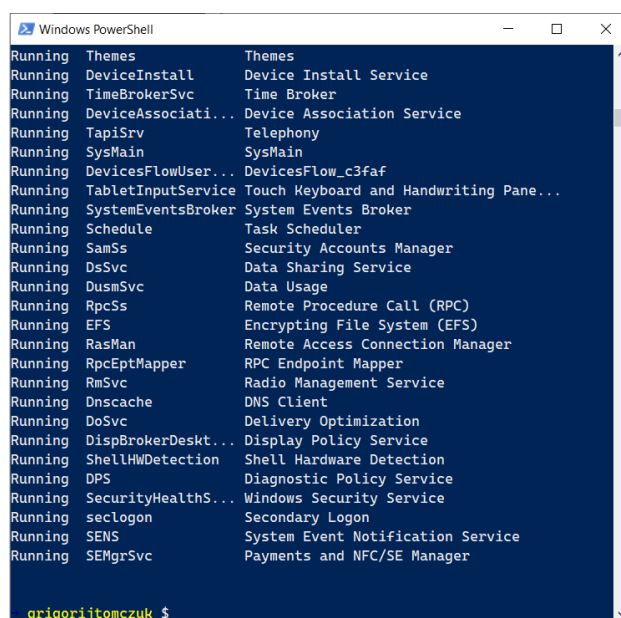
На рис. 3, 4 изображен результат выполнения упражнения 2.6.



```
grigorijtomczuk $ Get-Service | Sort-Object -property Status
```

Status	Name	DisplayName
Stopped	RpcLocator	Remote Procedure Call (RPC) Locator
Stopped	RetailDemo	Retail Demo Service
Stopped	RemoteRegistry	Remote Registry
Stopped	SCPolicySvc	Smart Card Removal Policy
Stopped	ScDeviceEnum	Smart Card Device Enumeration Service
Stopped	SCardSvr	Smart Card
Stopped	RemoteAccess	Routing and Remote Access
Stopped	PrintWorkflowUs...	PrintWorkflow_c3faf
Stopped	PrintNotify	Printer Extensions and Notifications
Stopped	PNRPsvc	Peer Name Resolution Protocol
Stopped	RasAuto	Remote Access Auto Connection Manager
Stopped	QWAVE	Quality Windows Audio Video Experience
Stopped	PushToInstall	Windows PushToInstall Service
Stopped	SDRSVC	Windows Backup
Stopped	SmsRouter	Microsoft Windows SMS Router Service.
Stopped	smphost	Microsoft Storage Spaces SMP
Stopped	shpamsvc	Shared PC Account Manager
Stopped	sppsvc	Software Protection
Stopped	spectrum	Windows Perception Service
Stopped	SNMPTRAP	SNMP Trap
Stopped	SharedRealitySvc	Spatial Data Service
Stopped	SensorService	Sensor Service
Stopped	SensorDataService	Sensor Data Service
Stopped	Sense	Windows Defender Advanced Threat Pr...
Stopped	SharedAccess	Internet Connection Sharing (ICS)
Stopped	SessionEnv	Remote Desktop Configuration

Рисунок 3 – Результат сортировки служб по статусу (начало)



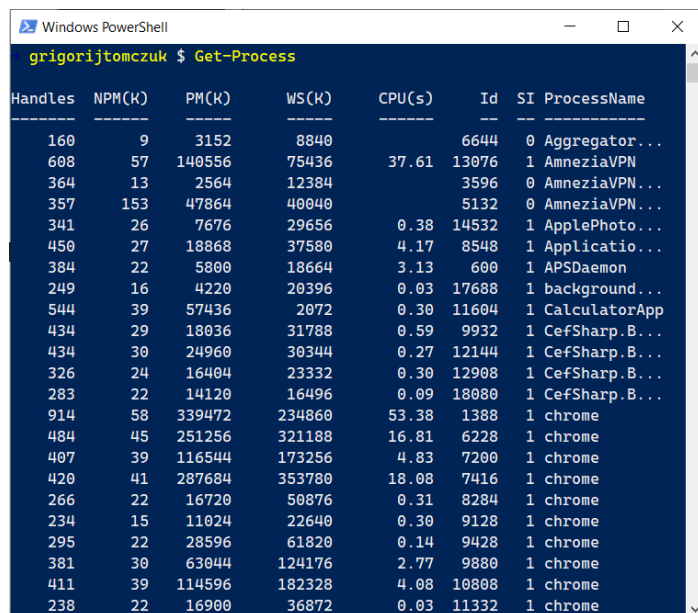
```
grigorijtomczuk $
```

Status	Name	DisplayName
Running	Themes	Themes
Running	DeviceInstall	Device Install Service
Running	TimeBrokerSvc	Time Broker
Running	DeviceAssociati...	Device Association Service
Running	TapiSrv	Telephony
Running	SysMain	SysMain
Running	DevicesFlowUser...	DevicesFlow_c3faf
Running	TabletInputService	Touch Keyboard and Handwriting Pane...
Running	SystemEventsBroker	System Events Broker
Running	Schedule	Task Scheduler
Running	SamSs	Security Accounts Manager
Running	DsSvc	Data Sharing Service
Running	DusmSvc	Data Usage
Running	RpcSs	Remote Procedure Call (RPC)
Running	EFS	Encrypting File System (EFS)
Running	RasMan	Remote Access Connection Manager
Running	RpcEptMapper	RPC Endpoint Mapper
Running	RmSvc	Radio Management Service
Running	Dnscache	DNS Client
Running	DoSvc	Delivery Optimization
Running	DispBrokerDeskt...	Display Policy Service
Running	ShellHWDetection	Shell Hardware Detection
Running	DPS	Diagnostic Policy Service
Running	SecurityHealthS...	Windows Security Service
Running	seclogon	Secondary Logon
Running	SENS	System Event Notification Service
Running	SEMGrSvc	Payments and NFC/SE Manager

Рисунок 4 – Результат сортировки служб по статусу (конец)

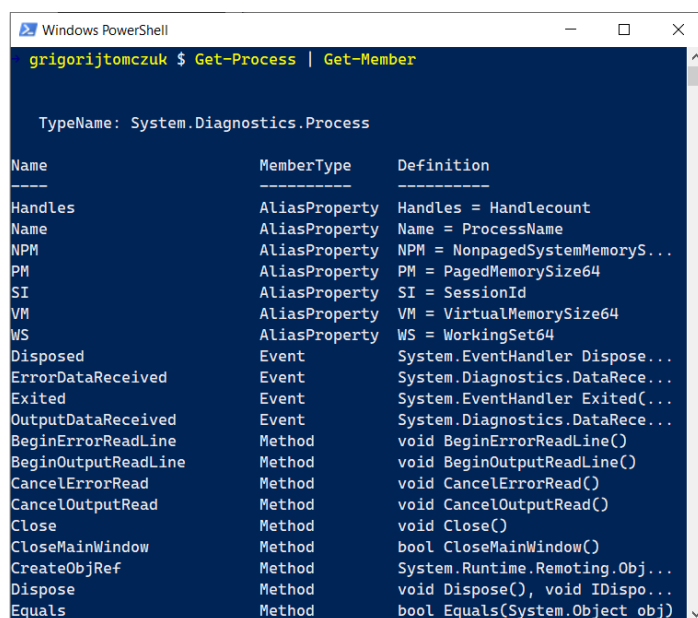
2.3 Упражнение 2.7

На рис. 5-7 изображен результат выполнения упражнения 2.7.



Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
160	9	3152	8840		6644	0	Aggregator...
608	57	140556	75436	37.61	13076	1	AmneziaVPN...
364	13	2564	12384		3596	0	AmneziaVPN...
357	153	47864	40040		5132	0	AmneziaVPN...
341	26	7676	29656	0.38	14532	1	ApplePhoto...
450	27	18868	37580	4.17	8548	1	Applicatio...
384	22	5800	18664	3.13	600	1	APSDaemon
249	16	4220	20396	0.03	17688	1	background...
544	39	57436	2072	0.30	11604	1	CalculatorApp
434	29	18036	31788	0.59	9932	1	CefSharp.B...
434	30	24960	30344	0.27	12144	1	CefSharp.B...
326	24	16404	23332	0.30	12908	1	CefSharp.B...
283	22	14120	16496	0.09	18080	1	CefSharp.B...
914	58	339472	234860	53.38	1388	1	chrome
484	45	251256	321188	16.81	6228	1	chrome
407	39	116544	173256	4.83	7200	1	chrome
420	41	287684	353780	18.08	7416	1	chrome
266	22	16720	50876	0.31	8284	1	chrome
234	15	11024	22640	0.30	9128	1	chrome
295	22	28596	61820	0.14	9428	1	chrome
381	30	63044	124176	2.77	9880	1	chrome
411	39	114596	182328	4.08	10808	1	chrome
238	22	16900	36872	0.03	11332	1	chrome

Рисунок 5 – Результат выполнения команды Get-Process



Name	MemberType	Definition
Handles	AliasProperty	Handles = Handlecount
Name	AliasProperty	Name = ProcessName
NPM	AliasProperty	NPM = NonpagedSystemMemoryS...
PM	AliasProperty	PM = PagedMemorySize64
SI	AliasProperty	SI = SessionId
VM	AliasProperty	VM = VirtualMemorySize64
WS	AliasProperty	WS = WorkingSet64
Disposed	Event	System.EventHandler Dispose...
ErrorDataReceived	Event	System.Diagnostics.DataRece...
Exited	Event	System.EventHandler Exited(...
OutputDataReceived	Event	System.Diagnostics.DataRece...
BeginErrorReadLine	Method	void BeginErrorReadLine()
BeginOutputReadLine	Method	void BeginOutputReadLine()
CancelErrorRead	Method	void CancelErrorRead()
CancelOutputRead	Method	void CancelOutputRead()
Close	Method	void Close()
CloseMainWindow	Method	bool CloseMainWindow()
CreateObjRef	Method	System.Runtime.Remoting.Obj...
Dispose	Method	void Dispose(), void IDispo...
Equals	Method	bool Equals(System.Object obj)

Рисунок 6 – Результат выполнения команды Get-Process | Get-Member

```

grigorijtomczuk $ Get-Process | Get-Member | Out-Host -Paging

TypeName: System.Diagnostics.Process

Name      MemberType Definition
-----
Handles   AliasProperty Handles = Handlecount
Name       AliasProperty Name = ProcessName
NPM        AliasProperty NPM = NonpagedSystemMemoryS...
PM         AliasProperty PM = PagedMemorySize64
SI         AliasProperty SI = SessionId
VM         AliasProperty VM = VirtualMemorySize64
WS         AliasProperty WS = WorkingSet64
Disposed   Event System.EventHandler Dispose...
ErrorDataReceived Event System.Diagnostics.DataRece...
Exited     Event System.EventHandler Exited(...
OutputDataReceived Event System.Diagnostics.DataRece...
BeginErrorReadLine Method void BeginErrorReadLine()
BeginOutputReadLine Method void BeginOutputReadLine()
CancelErrorRead Method void CancelErrorRead()
CancelOutputRead Method void CancelOutputRead()
Close      Method void Close()
CloseMainWindow Method bool CloseMainWindow()
CreateObjRef Method System.Runtime.Remoting.Obj...
Dispose    Method void Dispose(), void IDispo...
<SPACE> next page; <CR> next line; Q quit
Equals     Method bool Equals(System.Object obj)
<SPACE> next page; <CR> next line; Q quit

```

Рисунок 7 – Результат выполнения команды
Get-Process | Get-Member | Out-Host –Paging

2.4 Упражнение 2.8

На рис. 8-10 изображен результат выполнения упражнения 2.8.

```

grigorijtomczuk $ Get-Process | Sort-Object CPU -Descending

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----
1815     68     103484 106544 26,984.83 1124 1 iCloudHome
802       34     71124 69984 589.16 12524 1 TextInputHost
4153     135    331496 249016 447.14 5272 1 explorer
1885     128    160492 223584 424.13 8376 1 Playnite.De...
738      122     67692 101544 218.25 7356 1 nvcontainer
1252      42     70128 87604 145.17 13324 1 Twinkle Tray
2866      19      7100 25060 121.56 8164 1 ctfmon
1137      48     57080 69200 108.48 14304 1 iCloudDrive
1770      68    429088 216048 98.84 15484 1 WINWORD
1746     162    289948 340756 86.00 10076 1 SearchApp
288       25    118064 157208 66.16 15660 1 chrome
935       60    354680 211396 61.95 1388 1 chrome
2383      84    157940 297340 55.77 22428 1 chrome
546       23     27876 44416 46.86 7352 1 svchost
608       57    140556 75436 37.64 13076 1 AmneziaVPN
742       37     39172 99728 36.48 9232 1 StartMenuEx...
739       19      9496 34208 21.30 6072 1 sihost
484       46    243692 313104 19.89 6228 1 chrome
420       41    277772 344080 19.59 7416 1 chrome
556       27     76132 80480 14.56 13540 1 Twinkle Tray
362       24     36844 81640 13.73 13668 1 Twinkle Tray
861       47     21344 56136 10.83 9680 1 RuntimeBroker
539       49     35524 65256 9.56 12200 1 chrome
150       10     18076 9020 9.53 16772 1 PowerToys.K...
1286      39     79364 93968 9.23 1312 1 ShellExperi...
1029      26     19720 36808 9.00 16992 1 taskhostw

```

Рисунок 8 – Отсортированные в порядке убывания по CPU процессы
(Get-Process | Sort-Object CPU -Descending)

```
grigorijtomczuk $ Get-Process | Sort-Object CPU | Select-Object -First 5
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
367	15	2560	9152		3544	0	svchost
202	14	2608	12352		3652	0	svchost
218	11	2028	8108		3508	0	svchost
213	15	2528	10388		3536	0	svchost
633	30	39796	42568		3020	1	NVDisplay.Container

```
grigorijtomczuk $
```

Рисунок 9 – Результат выполнения команды
Get-Process | Sort-Object CPU | Select-Object -First 5

```
grigorijtomczuk $ $first5 = Get-Process | Sort-Object CPU | Select-Object -First 5
```

```
grigorijtomczuk $ $first5
```

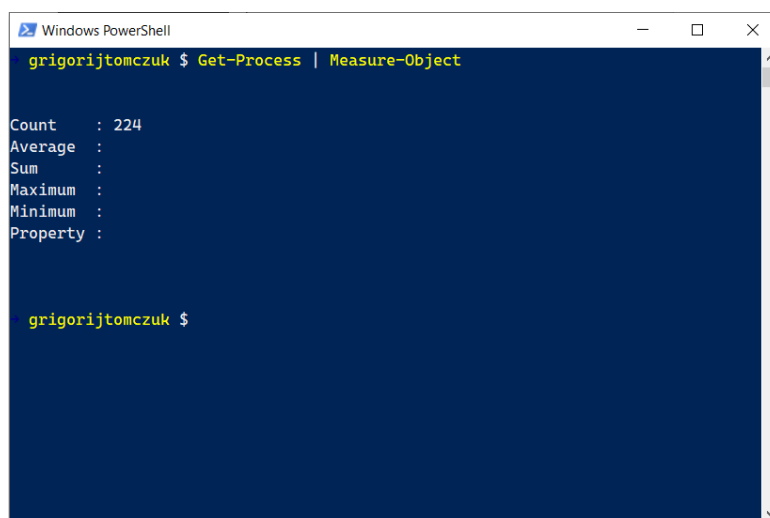
Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
367	14	2476	9136		3544	0	svchost
202	14	2608	12352		3652	0	svchost
218	11	2028	8108		3508	0	svchost
213	15	2528	10388		3536	0	svchost
796	27	33956	49652		4108	0	OfficeClickToRun

```
grigorijtomczuk $
```

Рисунок 10 – Вывод списка с использованием переменной \$first5

2.5 Упражнение 2.9

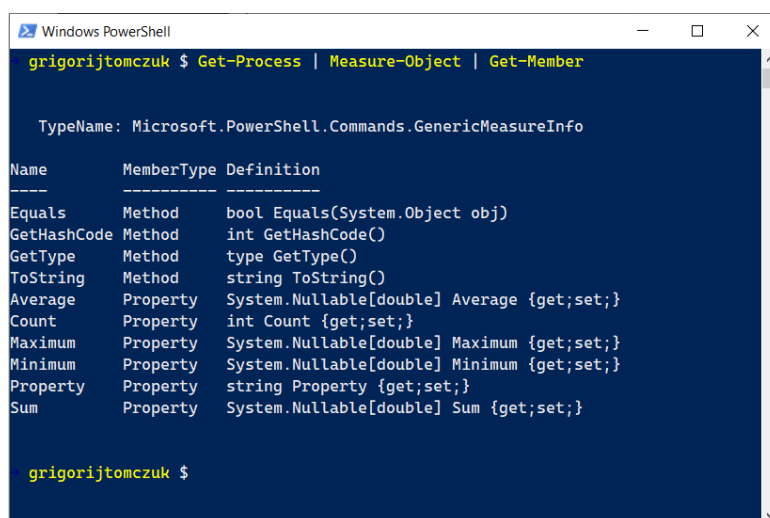
На рис. 11-15 изображен результат выполнения упражнения 2.9.



```
grigorijtomczuk $ Get-Process | Measure-Object

Count      : 224
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :
```

Рисунок 11 – Количество активных процессов



```
grigorijtomczuk $ Get-Process | Measure-Object | Get-Member

TypeName: Microsoft.PowerShell.Commands.GenericMeasureInfo

Name      MemberType Definition
-----
Equals    Method      bool Equals(System.Object obj)
GetHashCode Method    int GetHashCode()
GetType   Method      type GetType()
ToString  Method      string ToString()
Average   Property    System.Nullable[double] Average {get;set;}
Count     Property    int Count {get;set;}
Maximum   Property    System.Nullable[double] Maximum {get;set;}
Minimum   Property    System.Nullable[double] Minimum {get;set;}
Property  Property    string Property {get;set;}
Sum       Property    System.Nullable[double] Sum {get;set;}

grigorijtomczuk $
```

Рисунок 12 – Результат действий командлета Measure-Object (преобразование в GenericMeasureInfo)

```
Windows PowerShell
grigorijtomczuk $ Get-Process | Measure-Object -property VM -average -sum -mi
nimum -maximum

Count      : 222
Average    : 2126654225158.92
Sum        : 472117237985280
Maximum    : 3766023991296
Minimum    : 8192
Property   : VM

grigorijtomczuk $
```

Рисунок 13 – Размеры виртуальной памяти, занимаемой процессами

```
Windows PowerShell
lab05 $ Get-Process | Measure-Object

Count      : 228
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :

lab05 $ Get-Process | Get-Member -MemberType Property | Select-Object name |
Tee-Object -FilePath Property.txt | Measure-Object

Count      : 52
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :

lab05 $
```

Рисунок 14 – Подсчет активных процессов и вывод количества свойств, возвращаемых Get-Process

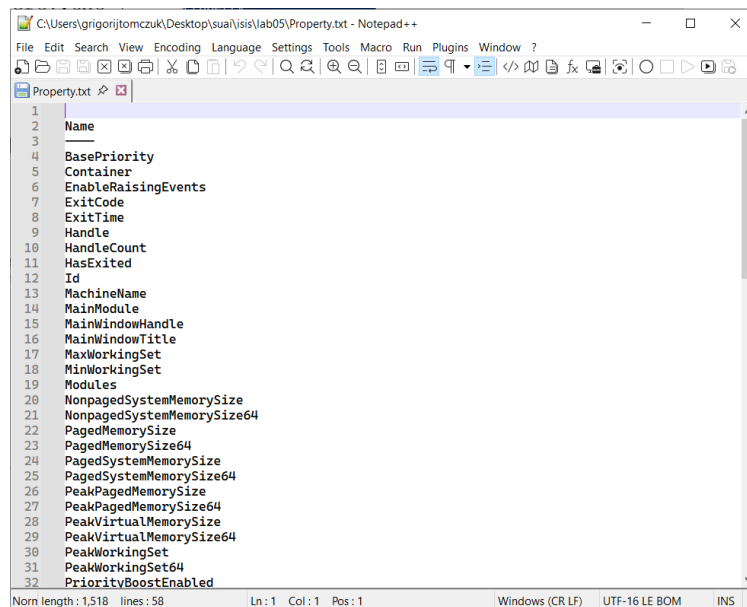


Рисунок 15 – Список свойств процесса, возвращаемых Get-Process

2.6 Упражнение 2.10

На рис. 16-18 изображен результат выполнения упражнения 2.10.

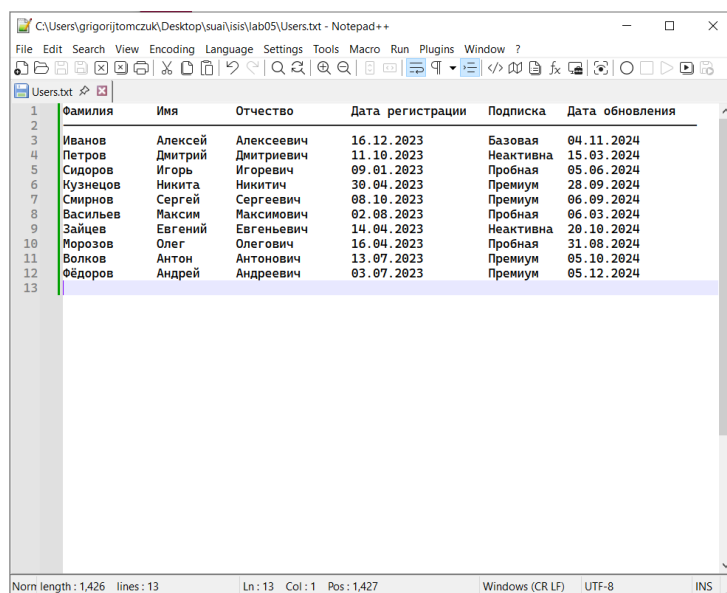


Рисунок 16 – Данные файла Users.txt

```

Windows PowerShell
Lab05 $ Get-Content .\Users.txt -Encoding UTF8

```

Фамилия	Имя	Отчество	Дата регистрации	Подписка	Дата обновления
Иванов	Алексей	Алексеевич	16.12.2023	Базовая	04.11.2024
Петров	Дмитрий	Дмитриевич	11.10.2023	Неактивна	15.03.2024
Сидоров	Игорь	Игоревич	09.01.2023	Пробная	05.06.2024
Кузнецов	Никита	Никитич	30.04.2023	Премиум	28.09.2024
Смирнов	Сергей	Сергеевич	08.10.2023	Премиум	06.09.2024
Васильев	Максим	Максимович	02.08.2023	Пробная	06.03.2024
Зайцев	Евгений	Евгеньевич	14.04.2023	Неактивна	20.10.2024
Морозов	Олег	Олегович	16.04.2023	Пробная	31.08.2024
Волков	Антон	Антонович	13.07.2023	Премиум	05.10.2024
Фёдоров	Андрей	Андреевич	03.07.2023	Премиум	05.12.2024

```

Lab05 $

```

Рисунок 17 – Результат выполнения командлета Get-Content

```

Windows PowerShell
Lab05 $ Get-Help Get-Content -Detailed

```

NAME

Get-Content

SYNOPSIS

Gets the content of the item at the specified location.

SYNTAX

```

Get-Content [-Credential <System.Management.Automation.PSCredential>] [-Delimiter
<System.String>] [-Encoding {ASCII | BigEndianUnicode | BigEndianUTF32 | Byte | Default |
OEM | String | Unicode | Unknown | UTF7 | UTF8 | UTF32}] [-Exclude <System.String[]>]
[-Filter <System.String>] [-Force] [-Include <System.String[]>] [-LiteralPath
<System.String[]>] [-Raw] [-ReadCount <System.Int64>] [-Stream <System.String>] [-Tail
<System.Int32>] [-TotalCount <System.Int64>] [-UseTransaction] [-Wait] [<CommonParameters>]

Get-Content [-Path <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-Delimiter <System.String>] [-Encoding {ASCII
| BigEndianUnicode | BigEndianUTF32 | Byte | Default | OEM | String | Unicode | Unknown |
UTF7 | UTF8 | UTF32}] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force]
[-Include <System.String[]>] [-Raw] [-ReadCount <System.Int64>] [-Stream <System.String>]
[-Tail <System.Int32>] [-TotalCount <System.Int64>] [-UseTransaction] [-Wait]
[<CommonParameters>]

```

DESCRIPTION

Рисунок 18 – HelpFile командлета Get-Content

2.7 Упражнение 2.11

На рис. 19 изображен результат выполнения упражнения 2.11.

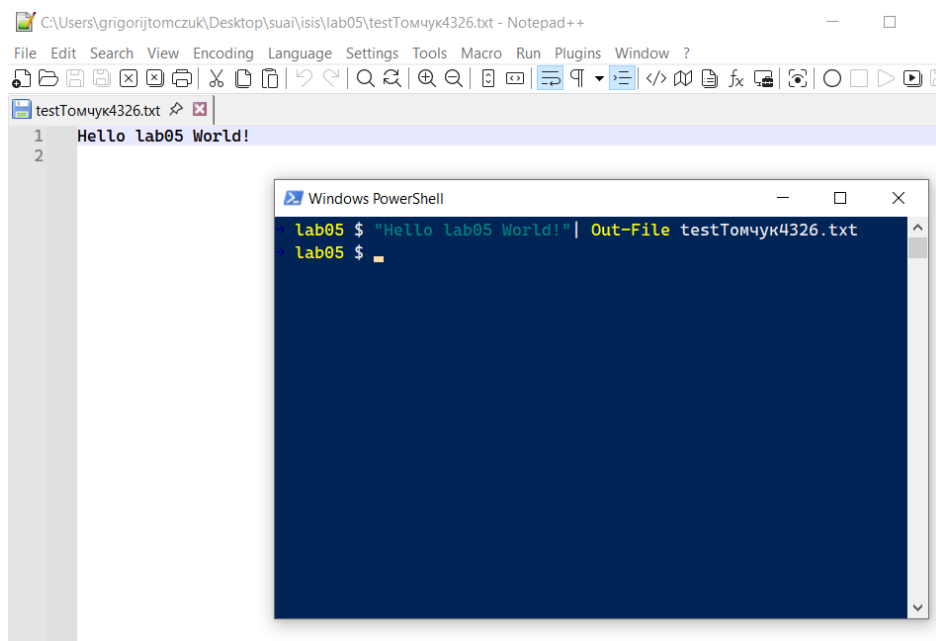


Рисунок 19 – Запись в текстовый файл

3 Результаты выполнения задания по варианту

На рис. 20-26 показаны результаты выполнения задания по варианту № 19 (командлеты: Set-Variable, Select-Object, Format-List).

Set-Variable назначает значение переменной. Базовый синтаксис: Set-Variable -Name <имя> -Value <значение>.

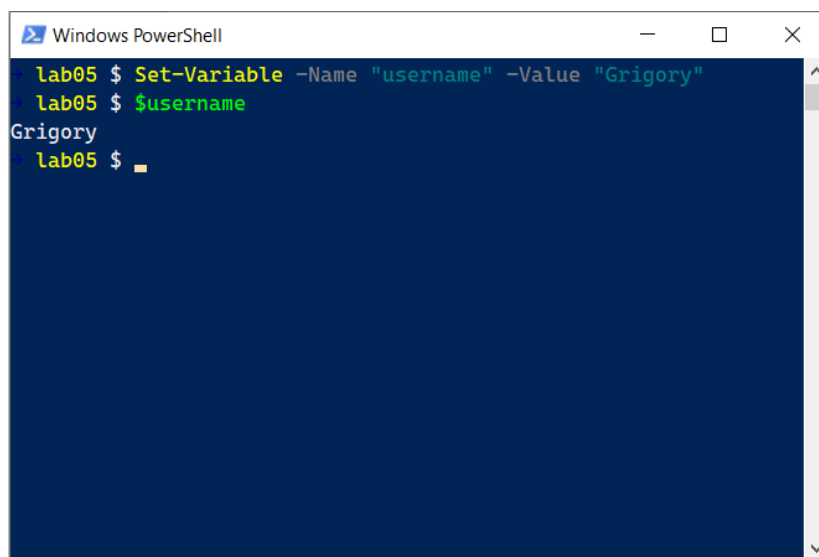
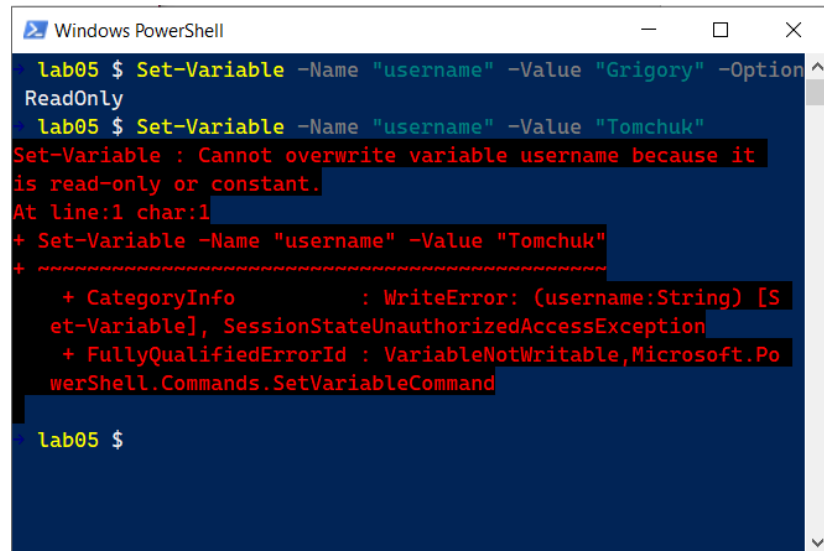


Рисунок 20 – Set-Variable

Также есть возможность задать значение переменной с флагами доступа.
Например, ReadOnly:

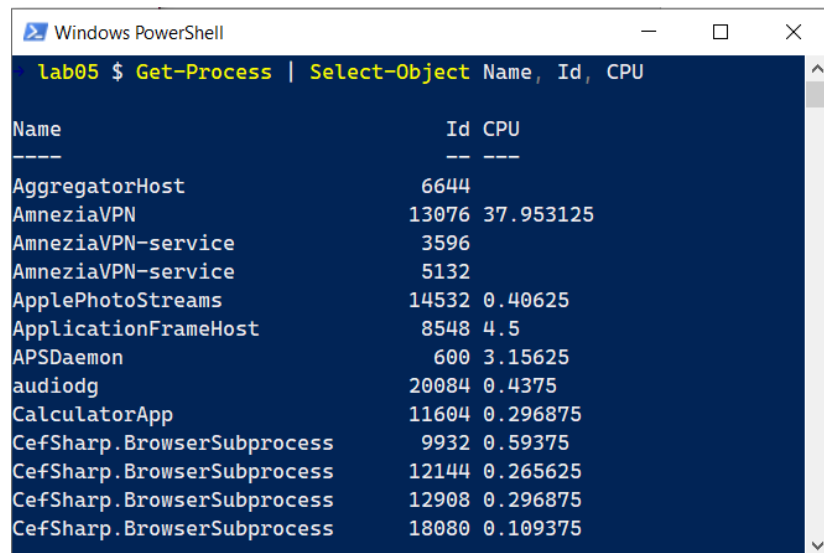


```
Windows PowerShell
lab05 $ Set-Variable -Name "username" -Value "Grigory" -Option
ReadOnly
lab05 $ Set-Variable -Name "username" -Value "Tomchuk"
Set-Variable : Cannot overwrite variable username because it
is read-only or constant.
At line:1 char:1
+ Set-Variable -Name "username" -Value "Tomchuk"
+ ~~~~~
+ CategoryInfo          : WriteError: (username:String) [S
et-Variable], SessionStateUnauthorizedAccessException
+ FullyQualifiedErrorId : VariableNotWritable,Microsoft.Po
werShell.Commands.SetVariableCommand

lab05 $
```

Рисунок 21 – Set-Variable -Option ReadOnly

Select-Object позволяет выбрать определённые свойства объекта.
Базовый синтаксис: <объект> | Select-Object <свойства>.

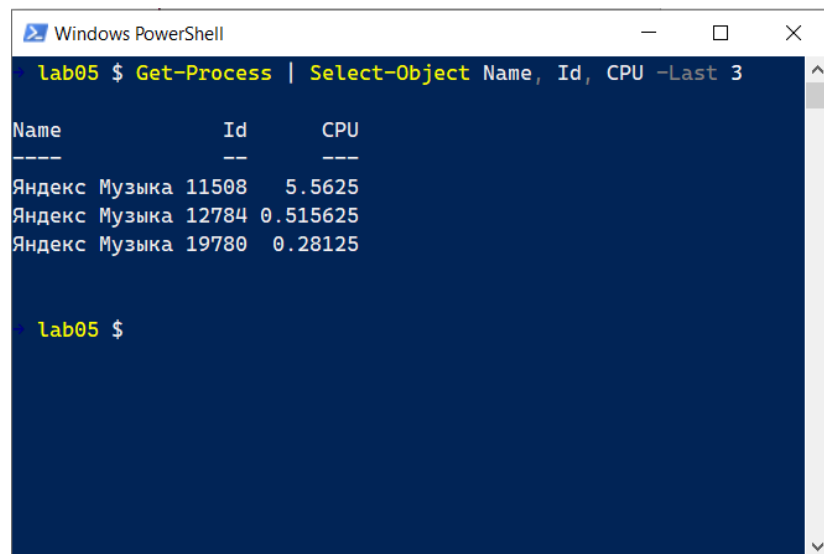


```
Windows PowerShell
lab05 $ Get-Process | Select-Object Name, Id, CPU

Name                                Id CPU
----                                -
AggregatorHost                     6644
AmneziaVPN                          13076 37.953125
AmneziaVPN-service                  3596
AmneziaVPN-service                  5132
ApplePhotoStreams                  14532 0.40625
ApplicationFrameHost                8548 4.5
APSDaemon                           600 3.15625
audiodg                             20084 0.4375
CalculatorApp                       11604 0.296875
CefSharp.BrowserSubprocess          9932 0.59375
CefSharp.BrowserSubprocess          12144 0.265625
CefSharp.BrowserSubprocess          12908 0.296875
CefSharp.BrowserSubprocess          18080 0.109375
```

Рисунок 22 – Get-Process | Select-Object Name, Id, CPU

Можно также выбрать только первые или последние строки с помощью
-First, -Last:



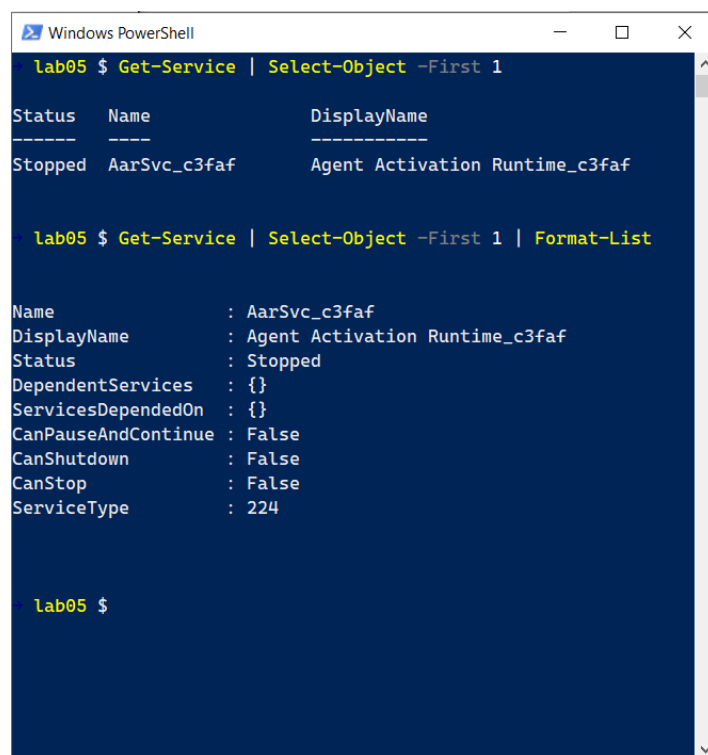
```
Windows PowerShell
lab05 $ Get-Process | Select-Object Name, Id, CPU -Last 3

Name      Id      CPU
----      -      -
Яндекс Музыка 11508  5.5625
Яндекс Музыка 12784  0.515625
Яндекс Музыка 19780  0.28125

lab05 $
```

Рисунок 23 – Get-Process | Select-Object Name, Id, CPU -Last 3

Format-List форматирует вывод в виде таблицы свойство-значение, что может быть удобно для просмотра подробностей. Базовый синтаксис: <объект> | Format-List <свойства>.



```
Windows PowerShell
lab05 $ Get-Service | Select-Object -First 1

Status  Name      DisplayName
-----  -
Stopped AarSvc_c3faf  Agent Activation Runtime_c3faf

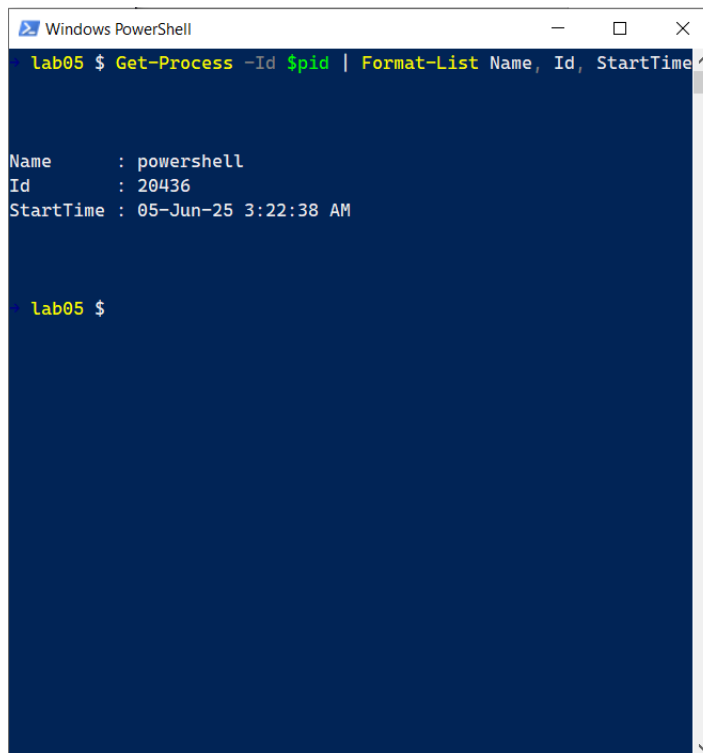
lab05 $ Get-Service | Select-Object -First 1 | Format-List

Name      : AarSvc_c3faf
DisplayName : Agent Activation Runtime_c3faf
Status    : Stopped
DependentServices : {}
ServicesDependedOn : {}
CanPauseAndContinue : False
CanShutdown : False
CanStop    : False
ServiceType : 224

lab05 $
```

Рисунок 24 – Get-Service | Select-Object -First 1 | Format-List

Можно также указать конкретные свойства для вывода:



```
Windows PowerShell
lab05 $ Get-Process -Id $pid | Format-List Name, Id, StartTime

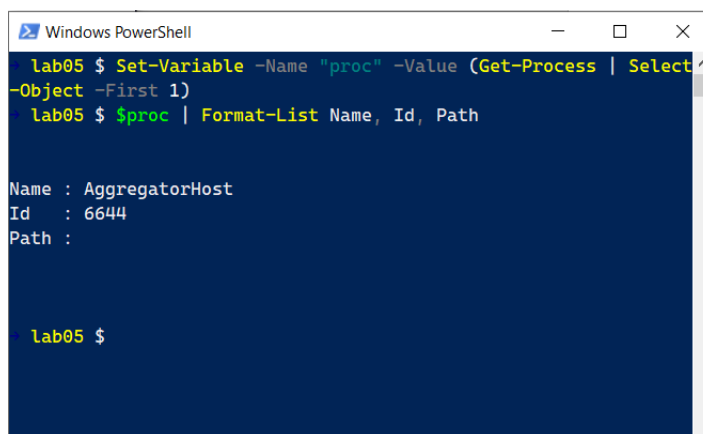
Name      : powershell
Id         : 20436
StartTime  : 05-Jun-25 3:22:38 AM

lab05 $
```

Рисунок 25 – Get-Process -Id \$pid | Format-List Name, Id, StartTime

Переменная \$pid имеет значение ID процесса рабочей оболочки PowerShell.

Все три командлета можно использовать вместе следующим образом:



```
Windows PowerShell
lab05 $ Set-Variable -Name "proc" -Value (Get-Process | Select-Object -First 1)
lab05 $ $proc | Format-List Name, Id, Path

Name : AggregatorHost
Id    : 6644
Path  :

lab05 $
```

Рисунок 26 – Запись переменной \$proc с Select-Object и ее вывод с Format-List

4 Выводы о проделанной работе

В ходе выполнения лабораторной работы были изучены базовые командлеты PowerShell.

Командлет `Get-Service` используется для получения информации о службах, установленных в системе. Он позволяет просматривать имя, статус и другие параметры сервисов, что удобно при контроле состояния системных компонентов.

`Sort-Object` применяется для сортировки данных по определённым свойствам. Это упрощает анализ больших объёмов информации и помогает выявлять нужные элементы по заданному критерию, например, отсортировать службы по статусу или процессы по потреблению ресурсов.

Командлет `Get-Process` предоставляет информацию о запущенных процессах в системе. Он позволяет отслеживать активные приложения, их ID, использование CPU и памяти, что полезно при администрировании и отладке.

`Select-Object` позволяет выбрать конкретные свойства объектов из общего вывода. Это делает вывод более компактным и фокусирует внимание на нужной информации, такой как только имя и статус службы или ID и имя процесса.

Командлет `Measure-Object` используется для получения статистических данных, таких как количество строк, слов, символов в файле или сумм и средних значений для числовых данных. Он удобен для быстрой оценки размеров и значений в выборках.

`Get-Content` позволяет читать содержимое текстовых файлов построчно. Это один из ключевых инструментов при работе с логами, конфигурационными файлами и другими текстовыми данными.

Командлет `Out-File` сохраняет вывод PowerShell-команд в текстовый файл. Он используется для документирования, логирования и дальнейшего анализа результатов выполнения скриптов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Администрирование и диагностика ОС Windows на персональном компьютере : учеб. пособие / А. В. Аграновский, К. Б. Гурнов, В. С. Павлов, Е. Л. Турнецкая. – СПб.: ГУАП, 2020. – 148 с., ил.
2. Русинович, М. Внутреннее устройство Windows / М. Русинович, Д. Соломон, А. Йосифович. – М.: ЛитРес, 2019. – 752 с.
3. Microsoft. Команды Windows, URL: <https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/windows-commands> (дата обращения 15.03.2025).