

Отчет 3 (PowerShell)

(задачи 1-3 знакомство с PowerShell)

Задача 4

Вывод на экран файлов из каталога Windows размером больше 10000 и отсортированных по размеру.

-Files позволяет вывести только файлы, Where-Object {\$.Length -gt 10000} –фильтр по размеру, Sort-Object Length –сортировка по размеру. Выводим только имена файлов и размеры – ничего лишнего:

```
PS C:\> Get-ChildItem -Path "C:\Windows" -File | Where-Object {$.Length -gt 10000} | Sort-Object Length | Select-Object Name, Length
```

Name	Length
-----	-----
DirectX.log	10009
DtcInstall.log	11101
write.exe	11264
winhlp32.exe	11776
hh.exe	18432
diagwrn.xml	22863
diagerr.xml	22863
CoreSingleLanguage.xml	29857
Core.xml	29857
DPINST.LOG	30866
mib.bin	43131
pyshellext.amd64.dll	50960
bootstat.dat	67584
twain_32.dll	76800
bfsvc.exe	93696
splwow64.exe	164352
notepad.exe	200704
WMSysPr9.prx	316640
regedit.exe	370176
PFR0.log	451020
pyw.exe	763664
py.exe	765208
HelpPane.exe	1065984
RtlExUpd.dll	2877104
explorer.exe	6089584
MEMORY.DMP	790708303

Эту информацию направляю в текстовый файл и через Type проверяю, что все записалось:

```
PS C:\> Get-ChildItem -Path "C:\Windows" -File | Where-Object {$.Length -gt 10000} | Sort-Object Length | Select-Object Name, Length >> C:\Users\Dasha\Documents\кринж\ОССТ\LAB3\files.txt
PS C:\> cd C:\Users\Dasha\Documents\кринж\ОССТ\LAB3\
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> type files.txt
```

Name	Length
-----	-----
DirectX.log	10009
DtcInstall.log	11101
write.exe	11264
winhlp32.exe	11776
hh.exe	18432
diagwrn.xml	22863
diagerr.xml	22863
CoreSingleLanguage.xml	29857
Core.xml	29857
DPINST.LOG	30866
mib.bin	43131
pyshellext.amd64.dll	50960
bootstat.dat	67584
twain_32.dll	76800
bfsvc.exe	93696
splwow64.exe	164352
notepad.exe	200704
WMSysPr9.prx	316640
regedit.exe	370176
PFR0.log	451020
pyw.exe	763664
py.exe	765208
HelpPane.exe	1065984
RtlExUpd.dll	2877104
explorer.exe	6089584
MEMORY.DMP	790708303

Задача 5

Создала переменную, в которую записаны все процессы, возвращаемые get-process, и их свойства. Затем записала значение переменной в txt файл, count возвращает количество процессов. Через type проверила, что в текстовый файл занесена информация обо всех процессах.

```
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> $process=Get-process
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> $process>processes.txt
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> $processes.count
234
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> type processes.txt
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
777	28	47188	43756	0,81	14792	6	Acrobat
872	66	234128	280568	4,17	18556	6	Acrobat
686	30	35568	61600	0,22	840	6	AcroCEF
799	30	16108	50692	0,83	6196	6	AcroCEF
339	21	45488	86148	1,23	7308	6	AcroCEF
256	17	8108	21308	0,06	8564	6	AcroCEF
328	21	10232	30064	0,08	16068	6	AcroCEF
335	21	35304	73744	0,97	21508	6	AcroCEF
140	8	2084	8924	0,19	12560	6	Adobe Crash Processor
1114	91	100208	152364	6,59	19408	6	Adobe Desktop Service
400	19	7436	21568	0,05	22236	6	AdobeCollabSync
558	25	17908	37968	0,31	22288	6	AdobeCollabSync
265	22	6184	12484	0,25	17544	6	AdobeIPCBroker
415	25	10852	3888	0,09	10604	6	AdobeNotificationClient
364	15	3852	12136		4888	0	AdobeUpdateService
163	9	3320	8484		7092	0	AggregatorHost
153	10	2484	6944		2844	0	amdfendrsr
388	23	24652	33348	0,05	22436	6	ApplicationFrameHost
351	18	4356	18048		4792	0	armsvc
327	16	3928	14860		14324	6	atieclxx
199	9	1888	6496		2852	0	atiesrxx
578	31	20620	11220	0,22	11372	6	Calculator
50	4	628	3172	0,02	19632	6	CCLibrary
50	4	612	3060	0,02	13804	6	CCXProcess
144	7	1748	9204	0,05	5196	6	CompPkgSrv
108	7	6260	4988		4368	0	conhost
273	14	6572	19164	1,52	9364	6	conhost
111	7	6344	11408	0,02	18716	6	conhost
111	8	6352	11556	0,02	20280	6	conhost
774	36	22220	49628	0,48	17636	6	CoreSync
220	10	2096	7884	0,00	3716	6	crashhelper
970	46	42968	103592	3,59	19428	6	Creative Cloud
501	28	11336	36440	0,22	9220	6	Creative Cloud Helper
241	18	8320	19788	0,09	4896	6	Creative Cloud UI Helper
713	35	237308	287432	13,05	7200	6	Creative Cloud UI Helper
296	19	14308	32740	0,47	14144	6	Creative Cloud UI Helper
671	35	112652	125864	0,73	19928	6	Creative Cloud UI Helper
719	27	2476	6132		832	0	csrss
814	40	10128	6728		17072	6	csrss
557	18	4864	23468	1,06	9808	6	ctfmon
339	14	3400	9288		2640	0	dasHost
98	6	1016	3688		4624	0	dasHost

и т. д.

Задача 6

В текстовый файл нужно записать ID, Имя процесса, Время старта и Handles. Сортировка по времени старта. Только для процессов с ID>40.

Where-object – задаем фильтры ID>40 и наличие времени старта процесса. Select-object – указываем параметры процесса, которые нужно вывести в файл (ID, Имя процесса, Время старта и Handles). Sort-object – указываем параметр процесса, по которому сортируем (время старта процесса). И Out-file – вывод в файл. Через type проверяю – файл записан корректно, все условия выполнены.

```
PS C:\Users\Dasha\Documents\кринж\OCCT\LAB3> Get-Process | Where-Object {$_.Id -gt 40 -and $_.StartTime} | Select-Object Id, Name, StartTime, Handles | Sort-Object StartTime | Out-File "processes_sorted.txt"
PS C:\Users\Dasha\Documents\кринж\OCCT\LAB3> type processes_sorted.txt
```

Id	Name	StartTime	Handles
8460	NahimicAPO4Volume	06.10.2025 10:01:06	132
3672	sihost	06.10.2025 10:01:06	593
17644	svchost	06.10.2025 10:01:06	473
3972	svchost	06.10.2025 10:01:06	165
15296	svchost	06.10.2025 10:01:06	538
15456	softinfo	06.10.2025 10:01:06	427
1504	taskhostw	06.10.2025 10:01:06	323
17516	taskhostw	06.10.2025 10:01:06	397
10460	explorer	06.10.2025 10:01:06	2610
20004	svchost	06.10.2025 10:01:07	342
13344	taskhostw	06.10.2025 10:01:07	260
12104	StartMenuExperienceHost	06.10.2025 10:01:07	786
13128	RuntimeBroker	06.10.2025 10:01:07	397
18768	SearchApp	06.10.2025 10:01:08	1596
6276	RuntimeBroker	06.10.2025 10:01:08	533
9944	UserOOBEBroker	06.10.2025 10:01:08	144
18968	RuntimeBroker	06.10.2025 10:01:09	375
9808	ctfmon	06.10.2025 10:01:09	659
4468	PhoneExperienceHost	06.10.2025 10:01:10	1324
8684	TextInputHost	06.10.2025 10:01:17	588
4740	SecurityHealthSystray	06.10.2025 10:01:19	165
12592	RtkAudUService64	06.10.2025 10:01:20	399
4300	figma_agent	06.10.2025 10:01:21	391
19868	Firefox	06.10.2025 10:01:23	1588
3716	crashhelper	06.10.2025 10:01:24	228
20196	Firefox	06.10.2025 10:01:24	785
11116	Firefox	06.10.2025 10:01:24	216
17744	Firefox	06.10.2025 10:01:24	366
18604	Firefox	06.10.2025 10:01:24	318
196	Firefox	06.10.2025 10:01:25	182
11136	Firefox	06.10.2025 10:01:25	332
4672	Firefox	06.10.2025 10:01:25	320
12060	LauncherBSXRuntime	06.10.2025 10:01:25	117

Задача 7

Нужно записать указанные параметры процессов, отсортированных по времени запуска, в html файл. Использую командлет Convertto-html. Invoke-item позволяет открыть html файл

```
PS C:\Users\Dasha\Documents\кринж\OCCT\LAB3> Get-process | Where-object {$_.StartTime} | Select-object Id, Name, StartTime, Handles | Sort-object StartTime | Convertto-html > processes.html
PS C:\Users\Dasha\Documents\кринж\OCCT\LAB3> Invoke-Item processes.html
```

Открыла в браузере - все корректно отсортировано:

HTML TABLE

file:///C:/Users/Dasha/Documents/кринж/OCCT/LAB3/processes.html

Индекс — быстрый п... Pinterest СПБ ГУПТД LibreView hay-doshirak (Dasha) Задания к практике P...

Id	Name	StartTime	Handles
4500	sihost	06.10.2025 11:57:17	582
4548	svchost	06.10.2025 11:57:17	165
4556	svchost	06.10.2025 11:57:17	476
4728	svchost	06.10.2025 11:57:17	532
5272	softinfo	06.10.2025 11:57:17	429
5324	taskhostw	06.10.2025 11:57:17	309
6828	ctfmon	06.10.2025 11:57:17	516
6900	explorer	06.10.2025 11:57:17	2538
1132	svchost	06.10.2025 11:57:18	329
7616	NahimicAPO4Volume	06.10.2025 11:57:18	131
8324	StartMenuExperienceHost	06.10.2025 11:57:18	647
8784	RuntimeBroker	06.10.2025 11:57:19	316
9144	SearchApp	06.10.2025 11:57:19	1468
9468	RuntimeBroker	06.10.2025 11:57:19	541
11180	RuntimeBroker	06.10.2025 11:57:20	370
9652	PhoneExperienceHost	06.10.2025 11:57:22	1315
3712	TextInputHost	06.10.2025 11:57:28	586

Задача 8

Подсчет размера всех jpg, bmp файлов в каталоге и его подкаталогах. Создаю переменную для хранения размера всех файлов и, используя цикл ForEach, прохожусь по всем файлам заданного каталога. -Recurse позволяет осуществить рекурсивный поиск в подкаталогах. Where-object отфильтровывает только jpg и bmp и их размер добавляется к общему. Вывожу значение переменной total_size – объем всех jpg и bmp файлов в байтах.

```
PS C:\Users\Dasha\Documents\кринж> $total_size=0
PS C:\Users\Dasha\Documents\кринж> ForEach ($f in (Dir C:\Users\Dasha\Documents\кринж -Recurse | Where-Object {$_.Extension -eq ".bmp" -or $_.Extension -eq ".jpg"})) {$total_size += $f.Length}
PS C:\Users\Dasha\Documents\кринж> $total_size
321370863
```

Задача 9

Вывод информации о процессоре:

```
PS C:\> Get-WmiObject Win32_Processor | Format-List

Caption           : AMD64 Family 25 Model 80 Stepping 0
DeviceID          : CPU0
Manufacturer      : AuthenticAMD
MaxClockSpeed     : 3201
Name              : AMD Ryzen 7 5800H with Radeon Graphics
SocketDesignation : FP6
```

Задача 10

Нужно найти максимальное, минимальное, среднее время выполнения dir и ps.

Инициализируем массив из 5 элементов, каждому элементу присваиваем значение времени выполнения командлета dir. Для наглядности выводим получившийся массив.

```
PS C:\> $dir_times = 1..5 | ForEach-Object {(Measure-Command {dir}).TotalSeconds}
PS C:\> $dir_times
0,0020037
0,0005551
0,0005197
0,001284
0,000523
```

Для этого массива используем Measure-Object и выводим нужные нам значения:

```
PS C:\> $dir_times | Measure-Object -maximum -minimum -average

Count      : 5
Average    : 0,0009771
Sum        :
Maximum    : 0,0020037
Minimum    : 0,0005197
Property   :
```

Аналогично для командлета ps:

```
PS C:\> $ps_times = 1..5 | ForEach-Object {(Measure-Command {ps}).TotalSeconds}
PS C:\> $ps_times
0,0198842
0,002719
0,0024465
0,0021932
0,0020698
PS C:\> $ps_times | Measure-Object -maximum -minimum -average

Count      : 5
Average    : 0,00586254
Sum        :
Maximum    : 0,0198842
Minimum    : 0,0020698
Property   :
```

Задача 11

1. Рекурсивная функция для вычисления факториала числа numb
Параметр функции – целое число, далее вычисление факториала. Работает корректно.

```
PS C:\Users\Dasha> function Get-Factorial {
>>     param([int]$numb)
>>     if ($numb -le 1) { return 1 }
>>     return $numb * (Get-Factorial ($numb - 1))
>> }
PS C:\Users\Dasha> Get-Factorial 1
1
PS C:\Users\Dasha> Get-Factorial 3
6
PS C:\Users\Dasha> Get-Factorial 6
720
PS C:\Users\Dasha> Get-Factorial 0
1
PS C:\Users\Dasha> 
```

(Функция 2 на след. странице)

2. Функция для вывода максимального и минимального числа из текстового файла.
Параметр функции – принимает путь к нужному файлу.
Далее создаем массив, который заполняем числами из файла (каждое число-отдельная строка),
каждый элемент приводим к числовому типу double.
Для данного массива используем командлет measure-object для получения минимального и
максимального значения.

```
PS C:\Users\Dasha> function Get-minmax {  
>> param([string]$FilePath)  
>> $numbers = Get-Content $FilePath | ForEach-Object { [double]$_ }  
>> $min = ($numbers | Measure-Object -Minimum).Minimum  
>> $max = ($numbers | Measure-Object -Maximum).Maximum  
>> return $min, $max  
>> }
```

Проверим работу используя файл nn.txt:

```
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> type nn.txt  
11  
199  
23  
22  
46  
9  
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3> Get-minmax nn.txt  
9  
199  
PS C:\Users\Dasha\Documents\кринж\ОССТ\LAB3>
```

Все работает.

Конец!