Практическая работа № 3

группа 2-МВ-4 | Шахбалаева Марина

Задания 1-3

Ознакомилась, успех

Задание 4

Выводим файлы и подкаталоги из каталога Windows на экран и сохраняем в файл с помощью команды Tee-Object (усл: начинаются на SY, сортируем по дате)

Get-ChildItem - команда для получения содержимого каталога (- path "C:\Windows")

Filter "SY*" - фильтр имен: выбирает только те файлы/папки, которые начинаются с "SY"

LastWriteTime - свойство, по которому выполняется сортировка (дата последнего изменения)

2, 7	Файлы и подкаталоги	По дате	Первые буквы имени SY

```
PS C:\Users\Марина> Get-ChildItem -Path "C:\Windows" -Filter "SY*" | Sort-Object LastWriteTime | Format-Table Name, LastWriteTime

Name            LastWriteTime
----            -------------
System          07.12.2019 12:14:52
system.ini      22.08.2021 13:45:42
SystemApps      21.12.2022 21:10:02
SystemResources 21.10.2025 12:44:23
SysWOW64        07.11.2025 14:07:27
System32        09.11.2025 14:21:17
SystemTemp      11.11.2025 12:27:08
```

Задание 5

С помощью команды **get-process** получаем список всех запущенных процессов и сохраняем в переменную **$proc**. Всю полученную информацию сохраняем в файл **proc.txt** и с помощью **$proc.count** (свойство .count возвращает количество элементов в переменной) считаем кол-во. Выводим на экран число, а за ним и сам список процессов.

```
PS C:\Users\Марина> $proc=get-process
PS C:\Users\Марина> $proc>proc.txt
PS C:\Users\Марина> $proc.count
227
PS C:\Users\Марина> type proc.txt

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    170      11     2176       8476       0,08  16348   8 AdobeIPCBroker
    165       9     4660      10220              8176   0 AggregatorHost
    403      23    15936      15220       1,52  15500   8 ApplicationFrameHost
    308      17    17776      20936     445,56  11744   0 audiodg
     50       4      608       3312       0,02  18208   8 CCXProcess
    558      26    90184      94008       7,16   4884   8 chrome
    217      10     6836       8736       0,11   5760   8 chrome
    408      23    31700      70288       0,95   6968   8 chrome
    278      20    22168      41864       0,20   9364   8 chrome
    226      14    13080      19080       0,30   9780   8 chrome
```

Задание 6

Создаем текстовый файл, содержащий список выполняемых процессов, упорядоченный по возрастанию времени старта

2, 6	Id, Имя процесса, время старта, Handles	Время старта	Id > 40

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

   Id Name                     StartTime           Handles
   -- ----                     ---------           -------
15468 sihost                   11.11.2025 21:27:20     793
11108 nvcontainer              11.11.2025 21:27:20     563
17872 svchost                  11.11.2025 21:27:20     422
 3100 svchost                  11.11.2025 21:27:20     165
 2156 igfxEM                   11.11.2025 21:27:20     672
12916 svchost                  11.11.2025 21:27:20     543
18308 taskhostw                11.11.2025 21:27:20     313
15172 explorer                 11.11.2025 21:27:20    4436
 2904 ctfmon                   11.11.2025 21:27:22     945
12416 svchost                  11.11.2025 21:27:22     502
 7112 StartMenuExperienceHost  11.11.2025 21:27:23     639
 3356 RuntimeBroker            11.11.2025 21:27:24     312
17736 PSAgent                  11.11.2025 21:27:24     158
11988 SearchApp                11.11.2025 21:27:24    1427
15508 RuntimeBroker            11.11.2025 21:27:24     617
 2432 LockApp                  11.11.2025 21:27:26     556
11392 RuntimeBroker            11.11.2025 21:27:26     445
```

Задание 7

Основа из предыдущего задания, добавляем **Convertto-html** и **Invoke-Item**, которые позволяют создать html-файл и открыть его

Id	Name	StartTime	Handles
15468	sihost	11.11.2025 21:27:20	824
11108	nvcontainer	11.11.2025 21:27:20	563
17872	svchost	11.11.2025 21:27:20	418
3100	svchost	11.11.2025 21:27:20	165
2156	igfxEM	11.11.2025 21:27:20	676
12916	svchost	11.11.2025 21:27:20	546
18308	taskhostw	11.11.2025 21:27:20	320
15172	explorer	11.11.2025 21:27:20	4122

```
PS C:\Users\Марина> Get-Process | Where-Object {$_.Id -gt 40 -and $_.StartTime} | Select-Object Id, Name, StartTime, Handles | Sort-Object StartTime | Convertto-html > proc.html
PS C:\Users\Марина> Invoke-Item proc.html
PS C:\Users\Марина>
```

Задание 8

Найдем суммарный объем всех графических файлов (bmp, jpg), находящихся в каталоге Windows и его подкаталогах

Так как на диске С не оказалось графических файлов, берем на рассмотрение диск F

Идем по всем подкаталогам с помощью **-Recurse**, суммируем с помощью **-Sum -** их объем. Что суммировать задаем с помощью **-Property Length -** (Length = размер файла в байтах). **-ErrorAction SilentlyContinue** позволит скрыть на экране ошибку доступа к папкам

```
PS C:\Users\Марина> (Get-ChildItem -Path "F:\" -Include "*.bmp", "*.jpg" -Recurse -File -ErrorAction SilentlyContinue | Measure-Object -Property Length -Sum).Sum
12797322393
PS C:\Users\Марина> _
```

Задание 9

Выводим на экран сведения о ЦП компьютера

Get-CimInstance Win32_Processor - получает информацию о процессоре через CIM (Common Information Model)

Format-List - форматирует вывод в виде списка

```
PS C:\Users\Марина> Get-CimInstance Win32_Processor | Format-List


Caption           : Intel64 Family 6 Model 158 Stepping 10
DeviceID          : CPU0
Manufacturer      : GenuineIntel
MaxClockSpeed     : 2592
Name              : Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
SocketDesignation : U3E1
```

Задание 10

Находим максимальное, минимальное и среднее значение времени выполнения командлетов dir и ps

1..10 - создали последовательность чисел от 1 до 10, то есть измерили время выполнения 10 раз. Время измеряем с помощью **measure-command {dir}** в секундах **(.totalseconds)**

И с помощью **measure-object -maximum -minimum -average** вычислили статистику и вывели на экран

1. **Dir**

```
PS C:\Users\Марина> $dir = 1..10 | foreach-object {(measure-command {dir}).totalseconds}
PS C:\Users\Марина> $dir
0,066982
0,0015898
0,0015076
0,0015268
0,0015213
0,0015573
0,0015214
0,0015321
0,001517
0,001516
PS C:\Users\Марина> $dir | measure-object -maximum -minimum -average


Count    : 10
Average  : 0,00807713
Sum      :
Maximum  : 0,066982
Minimum  : 0,0015076
Property :
```

2. **Ps**

```
PS C:\Users\Марина> $ps = 1..10 | foreach-object {(measure-command {ps}).totalseconds}
PS C:\Users\Марина> $ps_
0,0273165
0,0028249
0,0026852
0,0026476
0,0025447
0,0199222
0,0032114
0,0028537
0,0026565
0,0025474
PS C:\Users\Марина> $ps | measure-object -maximum -minimum -average


Count    : 10
Average  : 0,00692101
Sum      :
Maximum  : 0,0273165
Minimum  : 0,0025447
Property :
```

Задание 11

Разрабатываем командлет для нахождения количества различных чисел, хранящихся в файле nn.txt и для нахождения наибольшего числа, хранящихся в файле nn.txt

1. Обращаемся к файлу и читаем его содержимое с помощью **Get-Content "nn.txt" .** Превращаем полученную информацию в числа **- ForEach-Object { [int]$_ }** и сохраняем в переменную **$number**.
2. Считаем кол-во разных чисел (сортируем, оставляем уникальные и считаем)
3. Ищем наибольшее число и выводим результат на экран

```
PS C:\Users\Марина> $numbers = Get-Content "nn.txt" | ForEach-Object { [int]$_ }
>> Write-Host "Different numbers: $(($numbers | Sort-Object -Unique).Count)"
>> $max = ($numbers | Measure-Object -Maximum).Maximum
>> Write-Host "Largest number: $max"
Different numbers: 8
Largest number: 555
PS C:\Users\Марина> _
```

```
nn.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
1
2
3
4
65
35
77
555
3
1
```