

Чернівецький національний університет імені Юрія Федьковича  
Інститут фізико-технічних та комп'ютерних наук  
Відділ комп'ютерних технологій  
Кафедра математичних проблем управління і кібернетики

## **Звіт**

про виконання лабораторної роботи №4  
«Керування процесами та службами, отримання та аналіз  
системної інформації»  
з дисципліни  
**“Адміністрування операційних систем”**

Виконав: студент 441 групи  
Бужак Андрій

Перевірив: .....асист. Коцур М.П.

Оцінка:

Дата захисту:

Чернівці  
2019

**Завдання:** Ознайомитись із засобами PowerShell для керування процесами та службами. Отримати навички застосування засобів PowerShell для отримання системної інформації. Розробити сценарій для виконання перерахованих дій, описати використані засоби, провести тестування та оформити звіт:

- вивести список процесів;
- запустити (за вибором викладача) процес і засобами PowerShell зупинити його;
- вивести список запущених служб;
- запустити службу (за вибором викладача);
- вивести список команд що виконуються при завантаженні системи;
- вивести основні властивості ОС (назва, версія, розрядність);
- вивести список встановлених програмних продуктів;
- вивести інформацію про розмір фізичної пам'яті;
- вивести інформацію про процесор(-и);
- вивести інформацію про мережевий адаптер;
- вивести IP-адресу комп'ютера.

Інформацію слід виводити у структурованому вигляді.

Кількість балів – 8.

Додаткове завдання:

- розробити сценарій для вимкнення комп'ютера в зазначений час.

Кількість балів – 2.

Роздрукований звіт та працездатність сценарію перевіряє та оцінює викладач.

Максимальна кількість балів за роботу – 10.

Термін здачі роботи – до 22.11.19.

### **Хід виконання основного завдання**

Мною було розроблено сценарій для розв'язування основної задачі.

```
'Process list:';
Get-Process;
";
'=====';
";

$pathToStartProcess = Read-Host -Prompt 'Input filePath to start process';
$startedProcess = Start-Process -FilePath $pathToStartProcess -PassThru;
Start-Sleep -Seconds 2;
Stop-Process -Id $startedProcess.Id;

";
'=====';
";
'Running services:';
Get-Service | Where-Object -Property Status -eq Running | Format-Table Name,
DisplayName;

";
'=====';
```

```

";
$serviceNameToStart = Read-Host -Prompt 'Input name of service to start';
Start-Service -Name $serviceNameToStart;

";
'=====';
";
'Commands that runs on system loading:';
Get-WmiObject Win32_StartupCommand | Format-Table Caption, Command,
Location;

";
'=====';
";
'OS info:';
$osInfo = Get-WmiObject Win32_OperatingSystem;
'Name: ' + $osInfo.Name.Split('|')[0];
'Version: ' + $osInfo.Version;
'Architecture: ' + $osInfo.OSArchitecture;

";
'=====';
";
'Installed applications:';
Get-ItemProperty
HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\* `
| Select-Object DisplayName, DisplayVersion, Publisher, InstallDate `
| Sort-Object DisplayName `
| Where-Object DisplayName -ne $null `
| Format-Table;

";
'=====';
";
'Physical memory size: ' + (Get-CimInstance Win32_PhysicalMemory | Measure-
Object -Property Capacity -Sum).Sum / 1gb + ' GB';

";
'=====';
";
'Processors info:';
$property = 'Name', 'MaxClockSpeed', 'AddressWidth', 'NumberOfCores',
'NumberOfLogicalProcessors';
Get-WmiObject Win32_Processor -Property $property | Format-Table -Property
$property;

";

```

```
'=====';
";
'Network adapters:';
Get-NetAdapter;

";
'=====';
";
'IP-address: ' + (Test-Connection -ComputerName (hostname) -Count
1).IPV4Address;
";
'=====';
pause;
```

### Пояснення до сценарію основної задачі

1. Вказуючи властивість PassThru нам повернеться об'єкт, що містить дані про запущений процес. Запам'ятавши його в змінну можемо пізніше викликати його властивість Id що завершити процес по Id попередньо зачекавши дві секунди після старту.

```
$startedProcess = Start-Process -FilePath $pathToStartProcess -PassThru;
Start-Sleep -Seconds 2;
Stop-Process -Id $startedProcess.Id;
```

2. Wmi об'єкт Win32\_OperatingSystem у властивості Name містить окрім ім'я надлишкові дані, розділені символом '|', то зробивши Split беремо перший елемент.

```
$osInfo = Get-WmiObject Win32_OperatingSystem;
'Name: ' + $osInfo.Name.Split('|')[0];
```

3. Символ `` необхідний для розбиття виразу на декілька рядків, щоб полегшити читання коду.

```
'Installed applications:';
Get-ItemProperty
HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\* `
| Select-Object DisplayName, DisplayVersion, Publisher, InstallDate `
| Sort-Object DisplayName `
| Where-Object DisplayName -ne $null `
| Format-Table;
```

4. Так як планок пам'яті в системі може бути декілька, то повертається масив. Просумувавши ємність всіх елементів, маємо загальний розмір.

```
'Physical memory size: ' + (Get-CimInstance Win32_PhysicalMemory | Measure-Object -Property Capacity -Sum).Sum / 1gb + ' GB';
```

### **Хід виконання додаткового завдання**

Мною було розроблено сценарій для розв'язування додаткової задачі.

```
$answer = Read-Host -Prompt "Do you want enable scheduled shutdown? ['y' = enable; otherwise = disable]";  
";  
  
if (!$answer.ToLower()[0] -eq 'y')  
{  
    shutdown -a;  
    'Scheduled shutdown disabled.';  
    "  
    pause;  
    exit;  
}  
  
function IsTimeValid ([string]$time)  
{  
    if ([regex]::IsMatch($time, '\d\d:\d\d:\d\d'))  
    {  
        $timeItems = $time.Split(':');  
        if ([int]$timeItems[0] -le 23 -and [int]$timeItems[1] -le 59 -and  
[int]$timeItems[2] -le 59)  
        {  
            $true;  
        }  
        else  
        {  
            $false;  
        }  
    }  
    else  
    {  
        $false;  
    }  
}  
  
do  
{  
    $targetTime = Read-Host -Prompt "Input target shutdown time in next format -  
'hh:mm:ss', for example - 18:30:00  ";  
}  
while (!(IsTimeValid $targetTime));
```

```

$currentTime = Get-Date;
$targetTimeItems = $targetTime.Split(':');
$targetDateTime = Get-Date -Date $currentTime -Hour $targetTimeItems[0] -
Minute $targetTimeItems[1] -Second $targetTimeItems[2];

$timespan = (New-TimeSpan -Start $currentTime -End
$targetDateTime).TotalSeconds;
if ($timespan -lt 0)
{
    $timespan = 86400 + $timespan;
}

shutdown -f -s -t $timespan;

";
'Scheduled shutdown enabled.';
'System will shutdown after ' + $timespan + ' seconds.';
";
pause;

```

### Пояснення до сценарію додаткової задачі

1. Користувачу надається вибір або задати заплановане вимкнення або скасувати його.

```

$answer = Read-Host -Prompt "Do you want enable scheduled shutdown? ['y' =
enable; otherwise = disable]";
";

```

```

if (!$answer.ToLower()[0] -eq 'y')
{
    shutdown -a;
    'Scheduled shutdown disabled.';
    ";
    pause;
    exit;
}

```

2. Функція перевіряє регулярним виразом чи відповідає введений рядок формату часу і чи числові значення компонентів часу в допустимих межах.

```

function IsTimeValid ([string]$time)
{
    if ([regex]::IsMatch($time, '\d\d:\d\d:\d\d'))

```

```

    {
        $timeItems = $time.Split(':');
        if ([int]$timeItems[0] -le 23 -and [int]$timeItems[1] -le 59 -and
[int]$timeItems[2] -le 59)
        {
            $true;
        }
        else
        {
            $false;
        }
    }
    else
    {
        $false;
    }
}

```

3. Читання необхідного часу для вимкнення системи допоки він не буде введений у припустимому форматі.

```

do
{
    $targetTime = Read-Host -Prompt "Input target shutdown time in next format -
'hh:mm:ss', for example - 18:30:00 ";
}
while (!(IsTimeValid $targetTime));

```

4. Запам'ятовуємо поточний час, та будуємо цільовий час поки що припускаючи, що він буде в поточній добі.

```

$currentDateTime = Get-Date;
$targetTimeItems = $targetTime.Split(':');
$targetDateTime = Get-Date -Date $currentDateTime -Hour $targetTimeItems[0] -
Minute $targetTimeItems[1] -Second $targetTimeItems[2];

```

5. Якщо цільовий час виявиться в наступній добі, то результат (New-TimeSpan).TotalSeconds поверне від'ємне значення. Кількість секунд у добі 86400, додаємо до цього від'ємне обчислене значення \$timespan і отримаємо необхідну кількість секунд до вимкнення наступної доби.

```

$timespan = (New-TimeSpan -Start $currentDateTime -End
$targetDateTime).TotalSeconds;
if ($timespan -lt 0)
{
    $timespan = 86400 + $timespan;
}

```

## Результат виконання

### [Основне завдання]

```
Windows PowerShell
PS C:\Repositories\University-Course_4\AdministrationOfOperatingSystems\Lab_4> .\Lab_4.ps1
Process list:

Handles   NPM(K)    PM(K)      WS(K)      CPU(s)      Id   SI ProcessName
-----
375        21      15228      29468        0.20     8072   1 ApplicationFrameHost
356         38      67220     104724       15.55         8   1 chrome
198         16      10704      18444        0.33     1172   1 chrome
1361        47     136520     193536       49.73     3356   1 chrome
145         11       2048       8932        0.02     3504   1 chrome
452         43      94820      87480       47.94     3864   1 chrome
228         18      18640      30360        0.22     4820   1 chrome
238         51      63032      77652       13.56     4856   1 chrome
176          9       1712       6676        0.02     5404   1 chrome
233         93     162284     175116       9.06     6284   1 chrome
217         16      12872      20832        0.14     6968   1 chrome
459         32      16332      33448        7.06     7336   1 chrome
228         18      18768      30404        0.30     7700   1 chrome
204         12       5276      14256        0.41     1988   1 conhost
491         19       1776       5656         0.00         508   0 csrss
487         17       2400       5640         0.00         592   1 csrss
462         18      6892      19464        6.20     4768   1 ctfdmon
758         41     34532     57728        0.00         716   1 dwm
2552        107      73840     138116      18.98     5024   1 explorer
49          9       4228      10424        0.00         468   1 fontdrvhost
49          6       1656       3968        0.00         824   0 fontdrvhost
159         11      1852       8936        0.06     7268   1 hkcmd
181          9      1812       7844        0.00        1476   0 ibmpmsvc
0           0         56         8         0.00         0   0 Idle
201         12      2004       9516        0.05     7300   1 igfxpers
157         11      1832       8736        0.03     7236   1 igfxtray
1121        22      5672      15300        0.00         668   0 lsass
414         25     41552     39548        0.00        3980   0 matlab
190         13      4312       9328        0.00        2984   0 matlabserver

Windows PowerShell
2696         0        192       1156         0.00         4   0 System
738         36     18900      1256        0.50     3084   1 SystemSettings
322         16     4172     18852        0.05     6412   1 SystemSettingsBroker
264         27     5124     13752        0.33     4964   1 taskhostw
802         29     16816     45672        0.38     3888   1 WindowsInternal.ComposableShell.Ex
171         11     1400       7044        0.00         584   0 wininit
278         12     2472     11220        0.00        1012   1 winlogon
919         72     50372     2140        1.75     8016   1 WinStore.App
617         57     32856     78256       10.27         496   1 WINWORD

=====

Input filePath to start process: excel

=====

Running services:

Name                               DisplayName
-----
Appinfo                           Сведения о приложении
AudioEndpointBuilder              Средство построения конечных точек Windows Audio
Audiosrv                          Windows Audio
BFE                               Служба базовой фильтрации
BrokerInfrastructure              Служба инфраструктуры фоновых задач
Browser                           Браузер компьютеров
cbdhsvc_38d6f                     Пользовательская служба буфера обмена_38d6f
CDPSvc                           Служба платформы подключенных устройств
CDPUserSvc_38d6f                  Служба пользователя платформы подключенных устройств_38d6f
CoreMessagingRegistrar            CoreMessaging
CryptSvc                          Службы криптографии
DcomLaunch                        Модуль запуска процессов DCOM-сервера
DeviceAssociationService          Служба сопоставления устройств
Dhcp                              DHCP-клиент
```



## Commands that runs on system loading:

Caption	Command	Location
SecurityHealth	%windir%\system32\SecurityHealthSystray.exe	HKLM\SOFTWARE\Microsoft\Windows
IgfxTray	"C:\Windows\system32\igfxtray.exe"	HKLM\SOFTWARE\Microsoft\Windows
HotKeysCmds	"C:\Windows\system32\hkcmd.exe"	HKLM\SOFTWARE\Microsoft\Windows
Persistence	"C:\Windows\system32\igfxpers.exe"	HKLM\SOFTWARE\Microsoft\Windows

## OS info.

Name: Майкрософт Windows 10 Корпоративная LTSC  
Version: 10.0.17763  
Architecture: 64-разрядная

## Installed applications:

DisplayName	DisplayVersion
Any Video Converter	6.3.4
Cisco Packet Tracer	6.2 Instructor
ClickOnce Bootstrapper Package for Microsoft .NET Framework	4.7.03083
Entity Framework 6.2.0 Tools for Visual Studio 2017	6.2.61807.0
Google Chrome	78.0.3904.106
Google Update Helper	1.3.35.341
icecap_collection_neutral	15.8.27906
icecap_collectionresources	15.8.27924
icecap_collectionresourcesx64	15.8.27924
IntelliTraceProfilerProxy	15.0.17289.0

Physical memory size: 8 GB

## Processors info:

Name	MaxClockSpeed	AddressWidth	NumberOfCores	Number
Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz	2501	64	2	

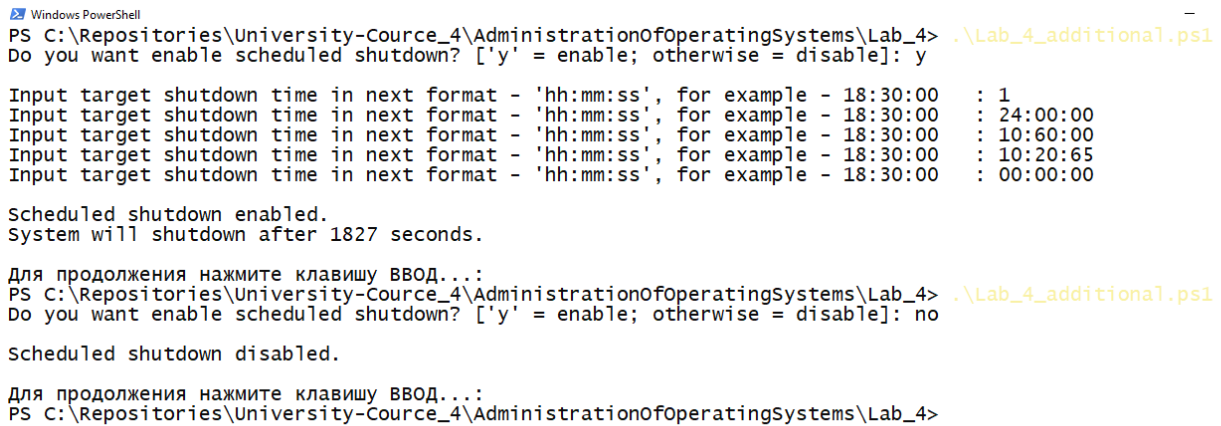
## Network adapters:

Name	InterfaceDescription	ifIndex	Status	Mac
Ethernet	Intel(R) 82579LM Gigabit Network Connection	8	Disconnected	F0-80-08-00-00-00
Беспроводная сеть	Intel(R) Centrino(R) Advanced-N 6205	7	Up	08-00-00-00-00-00

IP-address: 192.168.1.3

PS C:\Repositories\University-Course\_4\AdministrationOfOperatingSystems\Lab\_4> █

## [Додаткове завдання]



```
Windows PowerShell
PS C:\Repositories\University-Cource_4\AdministrationOfOperatingSystems\Lab_4> .\Lab_4_additional.ps1
Do you want enable scheduled shutdown? ['y' = enable; otherwise = disable]: y

Input target shutdown time in next format - 'hh:mm:ss', for example - 18:30:00 : 1
Input target shutdown time in next format - 'hh:mm:ss', for example - 18:30:00 : 24:00:00
Input target shutdown time in next format - 'hh:mm:ss', for example - 18:30:00 : 10:60:00
Input target shutdown time in next format - 'hh:mm:ss', for example - 18:30:00 : 10:20:65
Input target shutdown time in next format - 'hh:mm:ss', for example - 18:30:00 : 00:00:00

Scheduled shutdown enabled.
System will shutdown after 1827 seconds.

Для продовження натисніть клавішу ВВОД...:
PS C:\Repositories\University-Cource_4\AdministrationOfOperatingSystems\Lab_4> .\Lab_4_additional.ps1
Do you want enable scheduled shutdown? ['y' = enable; otherwise = disable]: no

Scheduled shutdown disabled.

Для продовження натисніть клавішу ВВОД...:
PS C:\Repositories\University-Cource_4\AdministrationOfOperatingSystems\Lab_4>
```

## ***Висновки***

Отже, аналіз результатів виконання роботи дозволяє стверджувати, що завдання виконане в повному обсязі – розроблений код працює згідно поставлених вимог.