Lab-1-1: Introduction to PowerShell

Objectives

- Gather a basic understanding of and experience with Microsoft PowerShell

Preparation

1. Open the PowerShell ISE (To do that type 'powershell' in the search field on your workstation

2. Select Windows **PowerShell ISE**). You can also perform this lab in the basic PowerShell environment (Windows **PowerShell**), but you may be missing the code completion facilities of the Interactive Scripting Environment (ISE).

3. You can also pin it to your taskbar for ease of access in future.

Exploring PowerShell

Solve the following tasks based on the **integrated help** (get-help), **lecture slides** and PowerShell **online reference**.

Write down the commands you used to achieve the tasks.

- Press Ctr+r. Press Ctr+r again. As you can see you are switching between windows.
- Type ls. Click the run button or F5.

  ls is a bash command, but you are using powershell. Lets check out if there are more bash commands available.

Q1. Type alias. Find at least five bash commands that you are familiar with from the list and write those along with their aliases below:

Ans:
    cd -> Set-Location
    cp -> Copy-Item
    group -> Group-Object
    move -> Move-Item
    rd -> Remove-Item

- Try the `cmd-lets` instead of the bash commands and see if they work

Q2. What is the use of aliases?

Ans: Can let user get, add, change, clear, and delete aliases in Powershell

- Find help about `get-childitem` command let. Use -detailed, -online and -examples options (use tab for auto completion).

Q3. Use `get-childitem` to show all files on your Desktop. Type the command as answer below:

    Ans: `get-childitem` to show all the files on the desktop

- Find help for `get-process` cmdlet with -online option

Q4. Use `get-process` cmdlet to find the username of the owner of powershell process (you might have to launch powershell with administrative privilege)

Ans: `get-process` get the process that are running on the local computer
    `get-process | get-member`
    `get-server`

Q5. Find the *username* of the owner of powershell using its process `id`

    Ans: Get-Process -Name YoudaoEh -IncludeUserName

```
PS C:\Users\cuip1\Desktop> Get-Process -Name YoudaoEh -IncludeUserName

Handles     WS(K)    CPU(s)     Id UserName                ProcessName
-------     -----    ------     -- --------                -----------
    128      9788      0.11   5852 OPNET\cuip1             YoudaoEH
```

Q6. Try to find the commands whose names start with '`Write`'
    Construct the command in two steps:
    a. First show all commands.

    b. Filter the results using '`-Name write*`'.

Ans:

Q7. Create a new file on your C drive using the cmdlet `new-item`. Try other ways of creating files (e.g. using *echo* or *touch*) and ensure they work.

Ans:

Q8. Everything is an object. Play around with this.
    a. Show the current date.
    b. Pipe it into '| `format-list`' to show all properties.
    c. Explore the properties using get-member.

Ans:
    Get-date
    Get-date | Format-List
    Get-date | Format-List

Q9. Show all folders and subfolder on the your Desktop. Do it in two steps. Use the parameter '–`Recurse`' to recursively show the files and folders.

[If you are receiving errors, identify why you got those. Which statement part is causing those? Adding the command '–`ErrorAction silentlyContinue`' will solve this.]

Ans:

Q10. Count the files and folders on C: drive. (Note: Extend the previous command.)

Ans:

Q11. Write the names of all files in C: to the file "files.log". (Look at the cmdlet *out-file*.) You can also redirect the output to the clipboard by piping it to 'clip'.

Ans:

Q12 What does the function 'ft' stand for?

Ans:

Q13. Show the top 15 processes that use most processor time.
        Do it in three steps and pipeline the results:
    d. Get all processes.
    e. Sort them based on CPU usage (highest usage first). Check the help for the 'sort' command
    f. Select the first 15.

Ans:

Q14. Determine the total used CPU time (Hint: have a look at *get-process*).

Ans:

<mark>Get-service</mark> this command display all the services

```
PS C:\Users\cuip1\Desktop> Get-Service

Status    Name               DisplayName
------    ----               -----------
Stopped   AarSvc_2244f0      Agent Activation Runtime_2244f0
Running   AdobeARMservice    Adobe Acrobat Update Service
Running   AdobeUpdateService AdobeUpdateService
Running   AERTFilters        Andrea RT Filters Service
Running   AGMService         Adobe Genuine Monitor Service
Running   AGSService         Adobe Genuine Software Integrity Se...
Stopped   AJRouter           AllJoyn Router Service
Stopped   ALG                Application Layer Gateway Service
Running   AMD External Ev... AMD External Events Utility
```

```
PS C:\Users\cuip1\Desktop> Get-Service | Where-Object {$_.Status -eq
"stopped"}

Status    Name               DisplayName
------    ----               -----------
Stopped   AarSvc_2244f0      Agent Activation Runtime_2244f0
Stopped   AJRouter           AllJoyn Router Service
Stopped   ALG                Application Layer Gateway Service
Stopped   AppIDSvc           Application Identity
Stopped   AppMgmt            Application Management
Stopped   AppReadiness       App Readiness
Stopped   AppVClient         Microsoft App-V Client
Stopped   AppXSvc            AppX Deployment Service (AppXSVC)
```