Labo Windows: Opdracht 03: Powershell

Achtergrond

1. Powershell

Powershell is de opvolger van de opdrachtprompt onder Windows. Op de recenste versies van Windows staat de shell reeds geïnstalleerd.

Alle opdrachten die in de opdrachtprompt werken, werken ook in de Powershell. Met de Powershell kan je echter nog veel meer, zo zijn er nieuwe opdrachten **cmdlets** genaamd die gebaseerd zijn op objecten. Verder is er ook een veel betere methode om scripts te schrijven.

1.1 Cmdlets

De nieuwe opdrachten gebruikt in powershell noemen cmdlets. Ze zijn allen gebaseerd op de naamconventie: *werkwoord -zelfstandig naamwoord*
Enkele voorbeelden: `Set-Date`, `Get-Service`, `Remove-Item`

1.2 Mogelijkheden

Powershell ondersteunt pipelining, bv. `Get-Acl c:\test | Format-List`
Dit zal de rechten van de directory test opvragen en mooi in een lijst weergeven.

Om eenvoudig scripts te kunnen schrijven bestaat er ook een GUI-editor: Powershell ISE. Deze biedt o.a. Intelli-sense aan.

In Powershell zijn alle hoofdprogrammeerfuncties aanwezig: arrays, bestanden, lussen, logische bewerkingen, …

1.3 Basisopdrachten

In onderstaande tabel vind je enkele veel gebruikte opdrachten en hun tegenhangers.

Powershell	Opdrachtprompt	Linux/UNIX
Get-Help	help	man
Get-ChildItem	dir	ls
Get-Process		ps
Set-Location	cd	cd
Move-Item	move	mv

Met `Get-Command` kan je een lijst van alle bestaande opdrachten weergeven.
Filteren kan door bv. met `Get-Command Write*`, dit zal alle opdrachten weergeven beginnende met Write.

1.4 Objecten

De output van iedere cmdlet is een object. Dit object kan verder gemanipuleerd worden. Voorbeelden zijn sorteren en attributen filteren.

Geef een lijst van lopende processen en soorteer volgens CPU verbruik:

```
Get-Process | Sort-Object CPU
```

Geef een lijst weer van lopende diensten en geef enkel de hostnaam weer:

```
Get-Service | ForEach-Object {write-host $_.name}
```

Idem als vorige maar enkel als de dienst gestopt is, geef ook de status weer:

```
Get-Service | ForEach-Object {if ($_.status -eq “stopped”)
{Write-Host $_.name $_.status}}
```

Powershell objecten kunnen gesorteerd worden met `Sort-Object`, groeperen kan met `Group-Object`. Sorteren zorg voor een volgorde van bv. hoog naar laag, groeperen zal alle gelijke kenmerken samen plaatsen en tellen.

Opgaves

De basisopdrachten kunnen in Powershell uitgevoerd worden, voor de scripts wordt best Powershell ISE gebruikt.

1. Basisopdrachten

1. Geef alle bestanden weer in de huidige directory en sorteer op grote.
2. Geef enkel de mode van alle bestanden weer (gebruik hiervoor o.a. `Write-Host`
3. Geef alle opdrachten weer die iets instellen (dus beginnende met Set)
4. Filter bovenstaande opdracht zodat alleen de namen van enkel de cmdlets weergegeven worden (gebruik `ForEach-Object`, `if` en `Write-Host`)

2. Korte scripts

1. Wis het scherm, geef vervolgens alle processen weer, groepeer op producent (“Company”) en sorteer tenslotte met het meeste aantal bovenaan
2. Schrijf een eenvoudig backup-script dat een directory kopieert naar een veilige plaats. Het script stelt eerst twee parameters in ($bron en $doel) en gebruikt die daarna om te kopiëren. Geef na het kopiëren de items terug in de doeldirectory. Maak om het script te testen een brondirectory “bron” aan op je desktop, zet hierin een aantal willekeurige bestanden. De brondirectory is dan bv. voluit “c:\users\mario.wyns\desktop\bron“. Kopieer recursief om de inhoud van de directory te kopiëren.