



Aufgabe 1

Starten Sie eine PowerShell als Administrator.

- a. Verwenden Sie das Cmdlet *Get-Process*, um sich alle laufenden Prozesse anzeigen zu lassen.
- b. Nutzen Sie eine Pipe¹, um das Ergebnis entweder als Tabelle (Cmdlet: *Format-Table*) oder als Liste (Cmdlet: *Format-List*) ausgeben zu lassen.
- c. Nutzen Sie eine Pipe, um das Ergebnis (wie mit *more* in der Eingabeaufforderung) seitenweise ausgeben zu lassen. Das zuständige Cmdlet ist *Out-Host -Paging*².
- d. Verwenden Sie den Befehl *Get-Process* / *Get-Member*, um sich alle vom Betriebssystem verwalteten Eigenschaften eines Prozesses anzeigen zu lassen.

¹ Pipe ist der Fachausdruck für das Zeichen |, das im QWERTZ-Layout die Drittbelegung auf der Taste links von ‚y‘ ist. Durch das Verketteten von Cmdlets mit einer Pipe, kann der Output eines Cmdlets als Input an das nächste Cmdlet weitergereicht werden.

² Das *-Paging* ist ein Parameter für das Cmdlet *Out-Host*. Ein Parameter bestimmt das Verhalten eines Programms, Befehls oder Cmdlets genauer.



Informationen

Das Cmdlet *Where-Object* ermöglicht die Filterung von Ergebnissen entsprechend bestimmter Eigenschaften (engl. *Property*). Hinter dem Namen des Cmdlets wird in geschweiften Klammern der Filterausdruck³ angegeben. Dabei ist *\$_* die Bezeichnung (fachsprachlich: die Referenz) jedes Objektes, das in das Cmdlet hereingereicht wird⁴.

Beispiel: Auflisten aller Prozesse der *svchost.exe* mit PowerShell.

```
Get-Process | Where-Object { $_.Name -eq "svchost" }
```

Man benötigt also lediglich den Namen der gewünschten Eigenschaft (Property), den Schwellwert sowie den passenden Vergleichsoperator:

- *-eq* (equal) für „ist gleich“
- *-ne* (not equal) für „ungleich“
- *-gt* (greater than) für „größer als“
- *-lt* (lesser than) für „weniger als“
- *-ge* (greather than or equal) für „größer oder gleich“ (ugs. „mindestens“)
- *-le* (lesser than or equal) für „kleiner oder gleich“ (ugs. „höchstens“)
- *-like* (like) für Mustererkennung in Zeichenketten (Strings)



Aufgabe 2

Verwenden Sie die Cmdlets *Get-Process* und *Where-Object* in einer Pipe.

- Listen Sie alle Prozesse auf, die Dienste laufen lassen:
Property: *SI* (Sitzungs-ID), *Schwellwert:* 0 (für Services)
- Listen Sie alle Prozesse auf, die mehr als 10 MB Arbeitsspeicher verwenden:
Property: *WorkingSet*, *Schwellwert:* 10000000
- Listen Sie alle Prozesse auf, die mindestens 0,5 Sekunden Rechenzeit verbraucht haben.
Property: *CPU*, *Schwellwert:* 0.5

³ Bei den Filtern handelt es sich um sogenannte Prädikate. Das sind Funktionen bzw. Ausdrücke, die entweder den Wert (logisch) *Wahr* oder (logisch) *Falsch* zurückliefern. Das logische *Wahr* heißt in der PowerShell *\$True* und das logische *Falsch* *\$False*.

⁴ In der Programmierung ist *Where-Object* äquivalent zu einer Schleife (mit anonymer Laufvariable) mit bedingtem If-Ausdruck darin.



- (Bonus) Finden Sie die Anzahl⁵ der Prozesse mit den Namen *svchost.exe* heraus.

Property: Name, *Prüfwert:* *svchost*

- (Bonus) Finden Sie heraus, wie viele Eigenschaften ein Object vom Typ *System.Diagnostics.Process* gemäß Output von *Get-Member* besitzt.

Aufgabe 3 (Bonus)

Neben den Vergleichsoperatoren kennt die PowerShell auch grundlegende logische Operatoren:

- *-And* für das logische Und
- *-Or* für das logische (einschließende) Oder
- *-Not* für das logische Nicht
- *-Xor* für das logische Entweder-Oder (ausschließendes Oder)



a. Verwenden Sie die Cmdlets *Get-Process* und *Where-Object* in einer Pipe.

- Listen Sie alle Prozesse auf, deren Name mit dem Buchstaben S beginnt:

Property: ProcessName, *Schwellwert:* S*

- Listen Sie alle Prozesse auf, deren Namen mit dem Buchstaben S oder mit dem Buchstaben I beginnt (welche Art von Oder verwenden Sie hier?):

Property: ProcessName, *Schwellwert:* S* und I*

- Listen Sie alle Prozesse auf, deren Namen nicht mit dem Buchstaben S beginnt:

Property: ProcessName, *Schwellwert:* S*

b. Filtern Sie den Output des Befehls *Get-Childitem C:\Windows -File* mit dem Cmdlet *Where-Object*.

- Listen Sie alle Dateien auf, die größer als 5 Kilobyte (kB) und kleiner als 100 Kilobyte (kB) sind.

Property: Length, *Schwellwert:* 5kB⁶ und 100kB

- Listen Sie alle Dateien auf, die kleiner als 10 kB sind oder deren Name mit win beginnt.

Property: Length und Name, *Schwellwert:* 10kB und win*

⁵ Verwenden Sie hierfür am Ende Ihrer Pipe das Cmdlet *Measure-Object*.

⁶ In der PowerShell können für Größenangaben die Abkürzungen kB (Kilobyte), MB (Megabyte) und GB (Gigabyte) verwendet werden.



- Listen Sie alle Dateien auf, die entweder kleiner als 10 kB sind oder deren Name mit *win* beginnt. Welche Dateien fehlen im Output dieses Befehls und warum?



Aufgabe 4

Starten Sie eine Instanz des Editors *Notepad*⁷.

Holen Sie sich das zugehörige Prozessobjekt mit *Get-Process -Name notepad*.

Beenden Sie den Prozess mit dem Cmdlet *Stop-Process* durch Angabe des Namens⁸.

Starten Sie mit *Start-Process* eine neue Instanz von notepad.

Beenden Sie diesen Prozess mit *Stop-Process* durch Angabe der Prozess-ID⁹.

Aufgabe 5



- a. Führen Sie den Befehl *Get-Command -Noun Computer* aus, um herauszufinden, welche Cmdlets im Zusammenhang mit dem Objekt *Computer* verfügbar sind.



- b. Notieren Sie das passende Cmdlet zum Neustarten des Computers und testen Sie es.
- c. Suchen Sie das passende Cmdlet zum Herunterfahren des Computers. Führen Sie dieses Cmdlet mit dem Parameter *-WhatIf* aus. Beschreiben Sie, welche Aufgabe der für (fast) jedes Cmdlet definierte Parameter *-WhatIf* hat.

⁷ Unter GNU/Linux einen Texteditor der entsprechenden Distribution wählen.

⁸ Der Parameter *-Name* kann optional verwendet werden.

⁹ Der Parameter *-Id* kann optional verwendet werden.