PowerShell

Pipeline in File System

Exercise 1

Run a PowerShell as administrator.

a. Sort the content of the Windows folder (only files) descending according to their length (size) and export the result to a CSV file *dateien.csv* in the current directory.

b. Determine the number of files with the extension .exe in the Windows folder. Also print their total and average size.

Exercise 2

a. Solve the Knuth problem (s. slides for chapter 3) in PowerShell. Use the text file *words.txt* for testing your solution.

   Hints:
   - For parsing the text file into single words you can use the expression `(Get-Content words.txt) -Split '\W+'` at the start of the pipeline.
   - The parameter n can be accessed as first command line argument via `$Args[0]`.

b. Compare your solution to the UNIX shell solution by Doug McIlroy. Which features and differences between the shells can you spot?

Exercise 3 (Bonus)

Construct a pipeline expression, that output the type of the returned object for a given Cmdlet as input. The figure shows Cmdlet Get-LocalUser as input. The type of the corresponding returned object is *Microsoft.PowerShell.Commands.LocalUser*. Part of the output is omitted.

```
PS C:\Users\anr> Get-LocalUser | Get-Member

   TypeName: Microsoft.PowerShell.Commands.LocalUser

Name                 MemberType Definition
----                 ---------- ----------
Clone                Method     Microsoft.PowerShell.Commands.LocalUser Clone()
Equals               Method     bool Equals(System.Object obj)
GetHashCode          Method     int GetHashCode()
```

Your command should precisely produce the following output:

```
Microsoft.PowerShell.Commands.LocalUser
```

Use various Cmdlets you know to test your pipeline and print the type of the resulting object.