

# Einführung in die Betriebssysteme

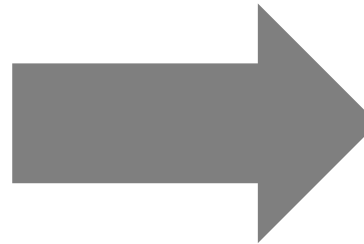
Martin Spörl

# Shellprogrammierung I – Windows Command Prompt

# Windows Command Prompt

## Grundlagen

- Ursprung in MS-DOS
- auch als „cmd.exe“ bekannt
- Standard Command-line Interpreter seit Windows-NT
- Ist mit COMMAND.COM (MS-DOS) kompatibel, hat aber Erweiterungen
  - Pfeiltasten um durch die Befehlshistorie zu wechseln
  - Vervollständigung für Datei- und Ordernamen
  - „|“ ist eine echte Pipe zwischen Prozessen (keine temporären Dateien wie bei MS-DOS)



## Probleme

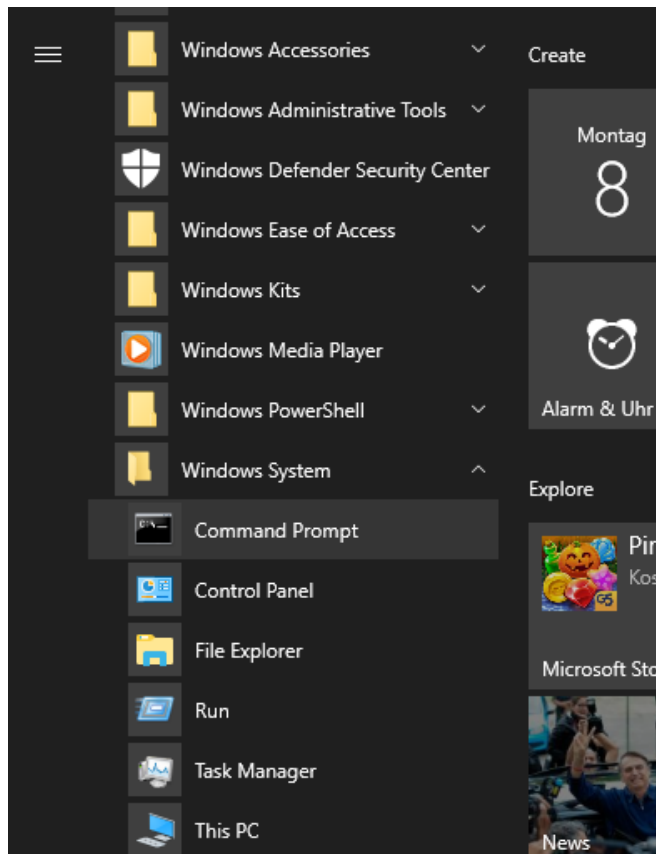
- kaum noch Erweiterungen
- oft Bugs die seit MS-DOS existieren
- wenig Flexibilität



Microsoft Powershell als Ersatz

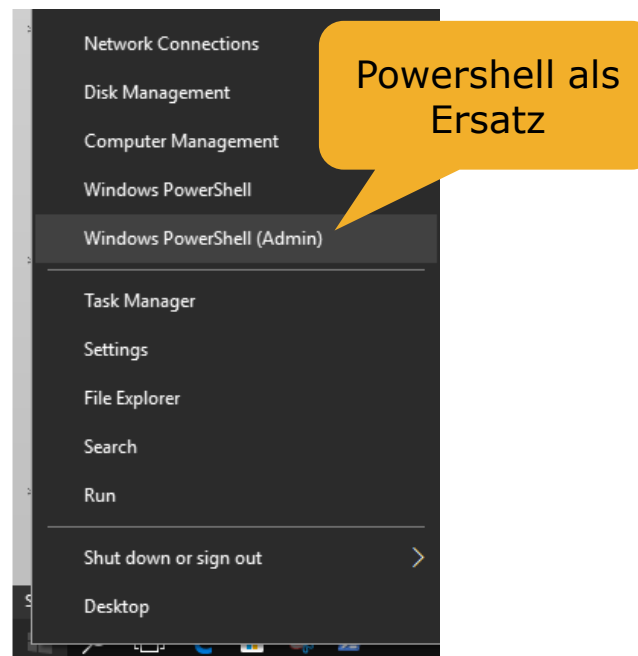
# CMD öffnen

## Startmenü



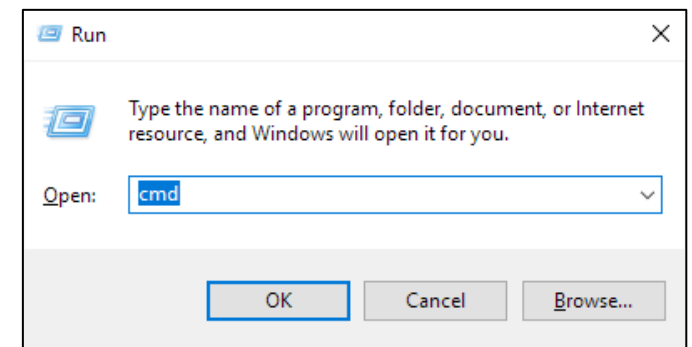
## Start Kontextmenü

- Rechtsklick auf Windows-Logo



## Tastenkombination

- Windows + R
- „cmd“



# Häufige Befehle

Befehl	Erklärung	(Basic) Syntax	Beispiel
copy	Kopiert Datei(en)	copy <source> <target>	copy C:\MyFile C:\My_New_File
move	Verschiebt eine Dateien/Ordner	move <source> <target>	move C:\MyFile C:\My_New_File
rename	Benennt Datei(en) um	rename <source> <target>	rename C:\MyFile C:\My_New_File
del	Löscht Datei(en) (auch rem)	del <file>	del C:\MyFile
cd	Wechselt das Verzeichnis	cd <directory name>	cd C:\MyDir
dir	Listet Dateien/Ornder im Verzeichnis	dir (aktuelles Verzeichenes) / dir <directory name>	dir C:\MyDir
echo	Gibt einen Text aus	echo "<Text>"	echo "Hello World"
date	Gibt / setzt das aktuelle datum	date	date
type	Gibt den Inhalt einer Datei aus (alles)	type <filename>	type C:\MyFile
more	Gibt den Inhalt einer Datei („soviel wie in Fenster passt“)	more <filename>	more C:\MyFile
mkdir	Erstellt einen Ordner	mkdir <directory name>	mkdir C:\MyDir

move vs rename:  
rename nur im  
selben Ordner &  
kann keine Datei  
überschreiben

# Allgemeine Hinweise

- Bei Leerzeichen Pfaden: "<Pfad>" schreiben
- /? Als Parameter gibt i.d.R. die Hilfe aus
- mehrere Befehle mit &&, & oder || verknüpfbar
  - && - Befehle werden ausgeführt, wenn vorheriger erfolgreich
  - & - Befehle werden ausgeführt, egal ob erfolgreich oder nicht
  - || - Befehl wird ausgeführt, wenn vorheriger nicht erfolgreich
- mit < (input) und > (output) kann der I/O Strom umgelenkt werden
- | (Pipe) verbindet Befehle (Output des ersten ist Input des zweiten)
- ^ ist Escape-Zeichen

# Windows Command Line

@echo off  
REM Befehl ohne Ausgabe  
dir  
@echo on  
:: Befehl mit Ausgabe  
dir

Verhindert Ausgabe der Befehle

Kommentar  
(**Schreibweise nicht empfohlen**  
– **Verwechslung mit remove**)

Optional – schaltet Ausgabe der Befehle ein (auch im Code)

Kommentar

Speichern als basic\_script.bat Datei

```
C:\INF - Betriebssysteme\Code\Batch>Basic_Script.bat
Volume in drive C is SOE
Volume Serial Number is 92AD-3351

Directory of C:\INF - Betriebssysteme\Code\Batch

08.10.2018  20:41    <DIR>          .
08.10.2018  20:41    <DIR>          ..
08.10.2018  20:42                29 Basic_Script.bat
               1 File(s)                29 bytes
               2 Dir(s)  16.262.447.104 bytes free

C:\INF - Betriebssysteme\Code\Batch>dir
Volume in drive C is SOE
Volume Serial Number is 92AD-3351

Directory of C:\INF - Betriebssysteme\Code\Batch

08.10.2018  20:41    <DIR>          .
08.10.2018  20:41    <DIR>          ..
08.10.2018  20:42                29 Basic_Script.bat
               1 File(s)                29 bytes
               2 Dir(s)  16.262.447.104 bytes free
```

# Variablen

## Hinweise

- Keine Typisierung!
- Vordefinierte Variablen (Auszug)

Befehl	Erklärung
COMPUTERNAME	Gibt den Computernamen an
HOMEDRIVE	Gibt die Partition des Benutzerordners an
HOMEPATH	Gibt den Pfad des Benutzerordners (ohne Laufwerksbuchstaben) an
PATH	Gibt den Suchpfad zurück
SystemDrive	Gibt die OS Partition zurück
SystemRoot	Gibt den Pfad zum Windowsverzeichnis an
TEMP / TMP	Gibt den Pfad zum temporären Verzeichnis an
USERNAME	Gibt Benutzernamen des ausführenden User an

```
@echo off
SET myvar=1234
echo %myvar%
```

```
C:\INF - Betriebssysteme\Code\Batch>variablen.bat
1234
```

```
@echo off
echo "Hello %USERNAME%"
echo "Your Windows is Running on %SystemRoot%"
echo "Your home folder is at
%HOMEDRIVE%%HOMEPATH%"
```

```
C:\INF - Betriebssysteme\Code\Batch>predefined_variables.bat
Hello Martin
Your Windows is Running on C:\WINDOWS
Your home folder is at C:\Users\Martin
```



# Benutzerinteraktion

## Benutzereingabe lesen

```
@echo off  
set /p myvar=Please enter your  
name:  
echo "Hello %myvar%"
```

/p fordert User zur  
Eingabe

„Zuweisung“ ist Text der  
dem User angezeigt wird

## Auf User warten

```
@echo off  
echo let's do it!  
echo ok first get some coffee  
pause  
echo ok - let's rock it!
```

# String-Operationen

## Substring

```
@echo off
set mystr=Hello World!
set hello=%mystr:~0,5%
echo %hello%
set world=%mystr:~-6%
echo %world%
set world2=%mystr:~6,5%
echo %world2%
```

Startindex  
(0-Based)

Länge

## Replace

```
@echo off
set mystr=Hello World!
set newstr=%mystr:World=Universe%
echo %newstr%
```

Search

Replace

## Concat

```
@echo off
set mystr1=Hello
set mystr2=World!
set constr=%mystr1% %mystr2%
echo %constr%
```

## Split

```
@echo off
set mystr=Hello;World!
for /F "tokens=1,2 delims=/" %%a in
("%mystr%") do (
    echo "%%a"
    echo "%%b"
)
```

Token 1 und 2 die von „;“  
separiert sind

String Operationen nur rudimentär (z.B.  
kein String-length!)

# Schleifen & Verzweigung

## GOTO

**VORSICHTIG VERWENDEN –  
kann zu Spaghetti Code führen**

```
@echo off
goto Sprung
echo "Ich werde nie gesehen"
:Sprung
echo "Aber mich sieht man"
```

```
C:\INF - Betriebssysteme\Code\Batch>goto.bat
"Aber mich sieht man"
```

## IF

*Kann mit GOTO kombiniert werden*

```
@echo off
SET myvar=1
IF %myvar% == 2 (
    echo "Yes - 2!"
) ELSE (
    echo "Damn... not 2 :("
)
```

```
C:\INF - Betriebssysteme\Code\Batch>if.bat
"Damn... not 2 :("
```

## FOR

```
@echo off
for /L %%N IN (0 2 10) DO (
    echo "Current: %%N"
)
```

```
C:\INF - Betriebssysteme\Code\Batch>for.bat
"Current: 0"
"Current: 2"
"Current: 4"
"Current: 6"
"Current: 8"
"Current: 10"
```

# for im Detail

```
for /L %%N IN (0 2 10) DO ( Code )
```

## Erklärung

- ***for***
- /L – die Liste beinhaltet Nummern
  - Alternativen /R, /D, /F
- %%N – Laufparameter (Name max. 1 Buchstabe lang)
- ***IN***
- (0 2 10) – (Startwert, Abstand, Endwert)
- ***DO***

## Alle Batch-Dateien ohne Pfad

```
@echo off  
for %%F IN (*.bat) DO (  
    echo "Current File: %%F"  
)
```

## Alle Batch-Dateien mit Pfad

```
@echo off  
for /R %%F IN (*.bat) DO (  
    echo "Current File: %%F"  
)
```

## Alle Ordner mit Pfad

```
@echo off  
for /D %%D IN (C:\*) DO (  
    echo "Current Line: %%D"  
)
```

# Arithmetische Operationen I

## Addition

```
@echo off
set myvar1=1
set myvar2=2
set /a sum=%myvar1%+%myvar2%
echo %sum%
```

/a zeigt  
Rechenoperation an

## Multiplikation

```
@echo off
set myvar1=1
set myvar2=2
set /a
mul=%myvar1%*%myvar2%
echo %mul%
```

## Modulo

```
@echo off
set myvar1=1
set myvar2=2
set /a mod=%myvar1% %%
%myvar2%
echo %mod%
```

## Subtraktion

```
@echo off
set myvar1=1
set myvar2=2
set /a diff=%myvar1%-%myvar2%
echo %diff%
```

## Division (ganzzahlig)

```
@echo off
set myvar1=1
set myvar2=2
set /a
quot=%myvar1%/ %myvar2%
echo %quot%
```

## Shift

```
@echo off
set /a "shiftright=1 << 2"
echo %shiftright%
set /a "shiftright=4 >> 1"
echo %shiftright%
```

# Arithmetische Operationen II

## Bitwise OR

```
@echo off
set myvar1=1
set myvar2=2
set /a "sum=%myvar1% |
%myvar2%"
echo %sum%
```

## Bitwise XOR

```
@echo off
set myvar1=1
set myvar2=2
set /a "sum=%myvar1% ^
%myvar2%"
echo %sum%
```

## Bitwise AND

```
@echo off
set myvar1=1
set myvar2=2
set /a "sum=%myvar1% &
%myvar2%"
echo %sum%
```

## Unterschiedliche Zahlensysteme

```
@echo off
::Hex 1
set myvar1=0x1
::Octal 16
set myvar2=020
set /a "sum=%myvar1% + %myvar2%"
echo %sum%
```

# Dateioperationen

## Datei einlesen

```
@echo off  
set /p tmp=<FileRead.bat  
echo "I read %tmp%"
```

## Datei (über-)schreiben

```
@echo off  
set tmp=This is not readable  
echo %tmp% > Test.txt  
set tmp2=This is readable  
echo %tmp2% > Test.txt  
type Test.txt
```

## Datei anhängen

```
@echo off  
set tmp=This is readable  
echo %tmp% > Test.txt  
set tmp2=This is also readable  
echo %tmp2% >> Test.txt  
type Test.txt
```



>> = Anhängen  
> = Ersetzen

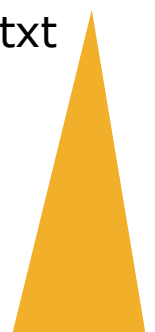
# Befehlsausgabe lesen

## mit „FOR“ (Text-Split)

```
@echo off
for /f "tokens=1-2,14" %%i in ('ipconfig') do (
    if "%%i %%j"=="IPv4 Address." set IPADDR=%%k
)
echo %IPADDR%
```

## durch temporäre Datei

```
@echo off
dir | findstr "File(s)" > temp.txt
set /p files=<temp.txt
echo %files%
```



„findstr“ sucht  
bestimmten String in der  
Ausgabe



# Funktionen

```
@echo off  
set erg=0  
call:MyFunction erg 2 3  
echo %erg%  
goto:eof
```

Funktionen immer zum  
Schluss

```
:MyFunction  
set param1=%~2  
set param2=%~3  
set /a %~1=%param1%*%param1%+%param2%  
goto:eof
```

Reihenfolge der Parameter  
entspricht Reihenfolge im  
Aufruf

# Übergabe Parameter

## Übergabe Parameter

```
@echo off  
echo "Hello, my name is %~0"
```

„%~0“ ist immer das  
Script selbst

## Erweiterte Parameter

```
@echo off  
call:MyFunction %~0  
echo "My Drive is %~d0"  
goto:eof  
  
:MyFunction  
echo "My file extension is %~x1"  
goto:eof
```

Kombinationen wie  
„%~nx1“ möglich

Extension	Erklärung
f	Voller Dateipfad
d	Nur Laufwerksbuchstabe (mit :)
p	Nur Pfad (ohne laufwerksbuchstabe)
n	Nur der Dateiname
x	Nur die Dateierweiterung
s	8.3 Konvention verkürzte Pfad/Namen
a	Nur die Dateiattribute
t	Zeitstempel der Datei
z	Größe der Datei

8.3 Konvention = Maximal 8 Zeichen oder Ziffern, gefolgt von Punkt, gefolgt von 3 Zeichen Dateierweiterung.

z.B. Textfile.txt wird zu TEXTFI~1.TXT  
(Hauptsächlich aus alten FAT-Dateisystem – also DOS Zeiten – verbleibend)