

Arbeiten auf der Kommandozeile

Name eines Kommandos

Kommandos liegen sowohl unter Windows als auch unter Unix als *Dateien* vor, dem *Programm-Namen* oder allgemeiner *Programm*:

programm.exe

Windows: Der (vollständige) Programm-Name setzt sich unter Windows häufig aus den Bestandteilen

<name>.<suffix>

zusammen. Die Klammern „<“ und „>“ schliessen hierbei den Namen (<name> = *programm*) und den Suffix (<suffix> = *exe*) ein. Aus historischen Gründen (DOS-Ära) charakterisiert das Suffix den Ausführungs-Typ. Ebenfalls historisch ist die Länge des Suffix, die 3 Zeichen umfasst; aus den gleichen Gründen ist der eigentliche Dateiname manchmal nur 8 Zeichen lang (8.3 Konvention). Das Kommando wird case-insensitiv, d.h. unabhängig von Gross- und Kleinschreibung des Dateinamens ausgeführt; zudem kann beim Aufruf auf das Suffix verzichtet werden.

Unix: Hier wird der Programm-Name immer case-sensitiv ausgewertet:

<Programm> ≠ <programm> ≠ <PROGRAMM>

Programm-Verarbeitung von Betriebssystem/Interpreter

Das Betriebssystem – oder allgemeiner der Interpreter – liest die Eingabe auf der Kommandozeile ein:

- Das Kommando wird abgeschlossen mit der <Return> Taste (auf der Tastatur)
- Das Kommando wird abgeschlossen mit einem speziellen Zeichen; unter der Unix-Shell ist dies das Semikolon <;>
- Das Kommando wird mehrere Zeilen – bis zu einem abschliessenden <RETURN> oder <;> gelesen – wobei unter Unix das Zeilenfortführungszeichen <\> ist.

Programme mit Argumenten

Bei vielen Programmen können bzw. müssen *Argumente* angegeben werden, z.B. den Namen der Datei, die gelesen werden soll:

Programm <Option> <Parameter1> <Parameter2>
 |===== Argumente =====|

Alle Eingaben bis zum Ende der Kommandozeile (siehe oben) werden als Argumente aufgefasst.

Argumente werden in der Regel immer durch (eine oder mehrere) Leerzeichen oder Tabulatoren (gemeinsam am Whitespaces bezeichnet) voneinander getrennt.

Wir unterscheiden bei den Argumenten:

- a) *Parameter* (mandatorisch oder optional), die zum Programm-Ablauf benötigt werden.
- b) *Optionen* (immer optional), die eine erweiterte Funktionalität des Programms oder für den Parameter bereitstellen. D.h. eine Option ist speziell an den Parameter oder aber allgemein an das Programm *gebunden*.

Unix: Traditionell steht hier das einfache Minus-Zeichen <-> oder ein doppeltes Minus-Zeichen <--> den eigentlichen Optionen vor, um diese von Parametern zu unterscheiden.

Windows: Optionen sind in der Regel mit dem Slash bzw. Schrägstrich </> kenntlich gemacht. Teilweise kann hier die Optionen ohne notwendigen Whitespace dem Kommando mitgegeben werden. Bei vielen Kommandos ist aber auch das Minus-Zeichen <-> zulässig.

Reihenfolge der Argumente und Parameter

Sowohl unter Unix als auch Windows werden die Optionen immer vor den Parametern angegeben; wobei die Reihenfolge der Optionen (nicht aber die der Parameter!) beliebig ist.

Getopts-Kompatibilität unter Unix

Unter Unix werden die Optionen der Programme häufig im sog. Getopts-Stil angeboten. Hierdurch können mehrere Optionen beim Aufruf zusammengefasst werden:

`programm -h -v` = `programm -vh` = `programm -hv`

Ferner ist es hierbei möglich (aber nicht gebräuchlich), Optionen mittels des Plus-Zeichens <+> *auszuschalten* bzw. zurückzusetzen.

Getopts erlaubt das Hinzufügen von sekundären Argumenten zu Optionen:

`programm -r <datei1> -w <datei2>`

Ausführbarkeit von Programmen

Ein Programm wird in der Regel auf der Dateiebene mittels *Dateiattributen* als ausführbar gekennzeichnet. Dies ist nicht primäre Eigenschaft des Programms, sondern des Dateisystems.

Unix: Die Dateiattribute liegen hier als *Permissions* vor und werden in der *Meta-Information* zur Datei hinterlegt. Die *Permissions* werden dem einfachen Zugriffsmodell von Unix entsprechend dem Eigentümer (*Owner*), der Gruppe (*Group*) bzw. allen anderen (*Others*) zugewiesen.

Das Unix-Kommando zum Setzen der *Execute-Permissions* lautet in seiner einfachsten Form:

```
chmod +x Programm
```

<+x> sowie <Programm> sind hierbei die (zwei) mandatorischen Parameter.

Windows: Auch Windows kennt unter NTFS das Dateiaattribut 'ausführbar'. Aufgrund der Kompatibilität für alte DOS-Programme, werden jedoch hier Programme zusätzlich mit dem Suffix <exe> gekennzeichnet. DOS-Programme mit statischer Speicherplatz-Alloziierung tragen den Suffix <com>.

Das Hilfsprogramm **cacls.exe** (aus dem Windows Resource-Kit) kann genutzt werden, dedizierte Dateiattribute und -rechte zu setzen.

Interpretation des Programm-Codes

Zusätzlich zur Kennzeichnung der Ausführbarkeit eines Programms auf der Datei-Ebene, muss das Programm 'innere' Qualitäten besitzen, damit es ausführbar ist.

- Das Programm liegt in einem *Binär-Format* vor. Der Loader des Betriebssystems muss beim Aufruf angefordert werden.
- Das Programm muss *zeilenweise interpretiert* werden und liegt somit als Skript oder sog. *Batch-Datei* vor.
 - Unter **Unix** bestimmt dann die erste Zeile, welcher Interpreter anzurufen ist.
 - Bei **Windows** wird der Programm-Suffix zur Kennzeichnung des Interpreters genutzt.

Lokation der ausführbaren Programme

Damit der Anwender nicht den vollständigen Pfadname der ausführbaren Datei angeben muss, wird ein Standard-Suchpfad im Environment des Benutzers bereit gestellt.

Unix: \$PATH

Windows: %PATH%

Unix: Zum Aufruf von Programmen im aktuellen Verzeichnis des Benutzers, die nicht im Suchpfad vorhanden sind, sind diese unter Unix mit ihrer (relativen) Pfadangabe zu benennen:

```
./programm
```

Alternativ kann das aktuelle Verzeichnis in den Pfad \$PATH bei Unix mittels

```
PATH=$PATH:.
```

eingebildet werden. Unter Windows ist dies der Default. Check:

```
Unix:      echo $PATH  
Windows:  echo %PATH%
```

Standard-Optionen bei Programmen

Viele Programme bieten sowohl unter Unix als auch unter Windows Standard-Optionen beim Aufruf an:

Ausgabe von Hilfe:

```
Unix:      programm -h  
            programm --help  
Windows:  programm /h  
            programm /?
```

Erweiterte Ausgabe (verbose):

```
Unix:      programm -v  
            programm --verbose
```

Hilfe zu Betriebssystem-Kommandos

Unix: Bei Unix existieren Man-Pages zu den Betriebssystem-Kommandos, die mit Hilfe des Aufrufs

```
man command
```

genutzt werden können. Mittes <q> wird die Ausgabe der Man-Page beendet. Über <Leerzeilentaste> bzw. <Return> wird der Text weiter gescrollt; das Zeichen dient zum Zurückblättern.

Windows: Nach Aufruf von

```
cmd
```

kann in diesem Fenster die Windows-Hilfe

```
help command
```

genutzt werden.