

Arithmetik in der tcsh

- Variablen speichern Zeichenketten (also Strings/Wörter)
- @ statt set \rightsquigarrow Interpretation als arithmetische Ausdrücke
(aus Ziffern, (,), +, -, *, /, % bestehend)

Beispiele:

@ var = (3 + 4) * 5	\rightsquigarrow	var = 35
@ var++	\rightsquigarrow	var = 36
@ modulo = \$var % 5	\rightsquigarrow	modulo = 1
@ modulo--	\rightsquigarrow	modulo = 0

- Es gelten die üblichen Vorrangregeln.
- Operanden müssen von Operatoren durch *whitespace* voneinander getrennt sein.

Wiederholung von Kommandos

1. **repeat**-Anweisung: `repeat count Kommando`

- *Kommando* wird genau *count* mal ausgeführt
- *Kommando* darf nicht Alias, Pipeline oder Kommandoliste sein!

2. **foreach**-Schleife: `foreach Variable Wortliste`
 `Kommando`
 `Kommando`
 `:`
 `Kommando`
 `end`

- Wiederholung der Kommandos so oft, wie die *Wortliste* lang ist
- Bei der *k*-ten Wiederholung nimmt die *Variable* den Wert des *k*-ten Wortes der *Wortliste* an.

3. **while**-Schleife: `while` (*Ausdruck*)
 Kommando
 Kommando
 :
 Kommando
 `end`

- Wiederholung der Kommandos solange, bis der *Ausdruck* den Wert 0 annimmt
 - **arithmetischer** *Ausdruck*
 - **boolescher** *Ausdruck* mit Werten 1 für true oder 0 für false
 - * durch Vergleich arithmetischer Ausdrücke mit ==, !=, <=, >=, <, >
 - * durch Vergleich zweier Zeichenketten mit ==, !=
 - ein **Kommando** in { ... } ergibt genau dann 1 (true), falls es fehlerfrei ausgeführt wurde (*Exit-Status 0*)

– Datei-Tests:

- e *datei* 1 gdw. *datei* existiert
- f *datei* 1 gdw. *datei* ist gewöhnliche Datei
- d *datei* 1 gdw. *datei* ist Verzeichnis
- z *datei* 1 gdw. *datei* hat Größe null
- s *datei* 1 gdw. *datei* hat nicht Größe null
- o *datei* 1 gdw. User ist Eigentümer von *datei*
- r *datei* 1 gdw. User hat Leserecht
- w *datei* 1 gdw. User hat Schreibrecht
- x *datei* 1 gdw. User hat Ausführungsrecht

– logische Verknüpfungen:

- && und-Verknüpfung
- || oder-Verknüpfung
- ! Negation

Bedingte Ausführung von Kommandos

1. **if**-Anweisung: `if (Ausdruck) Kommando`

- *Kommando* wird genau dann ausgeführt, wenn der *Ausdruck* den Wert 1 (`true`) annimmt
- *Kommando* darf nicht Alias, Pipeline oder Kommandoliste sein!
- wichtige *Kommandos*:
 - `break` \rightsquigarrow führt zum vorzeitigen (sofortigen) Verlassen der (direkt umgebenden) Schleife

Geschachtelte Schleifen können verlassen werden, indem entsprechend viele `break`-Anweisungen aufgerufen werden.
 - `continue` \rightsquigarrow führt zum Abbruch des aktuellen und Beginn des nächsten Schleifendurchlaufs

2. **if-then-else**-Anweisung: `if (Ausdruck) then`
 Kommando
 Kommando
 :
 Kommando
`else`
 Kommando
 Kommando
 :
 Kommando
`endif`

- Die Kommandos nach `then` werden ausgeführt, falls der *Ausdruck* den Wert 1 (`true`) annimmt, sonst die Kommandos nach `else`.
- `else` und `endif` müssen die ersten Wörter einer Zeile sein.
- Der `else`-Teil ist optional.