

7.1 Bourne Shell Beispiele

```

8    ./s0    - && und ||

10   ./s1    - Variable und Umgebungsvariable
      Wertzuweisung an Variable
      ./s1a  - Wertzuweisung an Variable und
      Kommando mit temporaerer Umgebungsvariablen

12-14 ./s2    1 2 "3 4" - ohne Aufsplittung
      ./s2a  1 2 "3 4" - mit Aufsplittung
      ./s2b  1 2 "3 4" - mit Aufsplittung
      und ignorieren von Blanks
      Unterschiede von
      "$*" und "$@"

15-17 ./shows s3 - :- leer, dann Wert benutzt
      ./shows s4 - - nicht definiert, dann Wert
      benutzten
      ./shows s5 - := leer, Variable erhaelt den
      Wert
      ./shows s6 - = nicht definiert, Variable
      erhaelt den Wert

18   ./shows s7 - :? Test ob Variable einen Wert
      hat, sonst, Ausschrift+Abbruch
      ./shows s8 - ? Test ob Variable definiert,
      sonst, Ausschrift+Abbruch
      ./shows s9 - :+ Ausschrift, wenn Variable
      einen Wert hat
      ./shows s10 - + Ausschrift, wenn Variable
      definiert ist

19-21 more s12    - Addition mittels bc,
      Demo von <<EOF
      ./s12
      more s13    - Das Gleiche mit expr
      ./s13

22   more s14    - Ausgabeumlenkung ung Anfuegen mit:
      >>
      ./s14
      more s15    - Ausgabeumlenkung und Anfuegen mit:
      exec >>
      ./s15

```

```

23      s16,s17      - ohne #!/bin/sh und mit "!/bin/sh
                        am Anfang
        ./s16
        ls -lisa s16
        sh s16
        cat s17
        ls -lisa s17
        ./s17
        vi s17
        bin --> xbin
        ./s17
cat s18      # Parameter ueber $1 ... $9
        ./s18 1 2 3 4 5 6 7 8 9
        ./s18 1 2 3 4 5 6 7 8 9 10 11
        ./s18 asdf qerq=1234 -asdf=qewr -6
cat s19      # Parameter verschieben mit shift
        ./s19 1 2 3 4
        ./s19 1 2 3
        ./s19 asdf qerq=1234 -asdf=qewr -6

24-26
s20 - if-Anweisung
        ./s20
s21 - if, case -Anweisung
        cat s21
        ./s21 0
        ./s21 9
        ./s21 a
        ./s21 ab
s22 - if, while-Anweisung
        cat s22
        ./s22 2
        ./s22 3
s23 - if, until-Anweisung
        cat s23
        ./s23 4
s24 - if, for-Anweisung
        cat s24
        ./s24 p1 p2 p3 p4

27 bis . Kommandodatei
s30 - Wirkung der .-Anweisung
        cat s30a
        ./s30

28-28 continue
s24a - break
        ./s24a 1 2 3 0 4 5 6 0
s24b - continue
        ./s24a 1 2 3 0 4 5 6 0

```

s22 mit /usr/bin/ec
s22a mit test
s22b mit [

```
s31 - Wirkung von break und continue
cat s31
./s31      i mit 1 und 3
           j mit 2 und 4
           k continue fuer > 2

28 eval
    s32 - Wirkung von eval
    ./s32

28 exec
    ./s11 - exec mit Kommando beendet aktuelles Shellsc
    cd Rdist
    more run-rdist - exec zur Umleitung von
                    E/A

28 exit
    cat s33
    ./s33 13
    ./s33 1
    ./s33 -1

29 getopt
    cat s34
    ./s34 -a
    ./s34 -a -b
    ./s34 -c
    ./s34 -c 3
    ./s34 -a -d 4 -c 1

29 read - beachte Anzahl der Variablen pro read
    cat s35
    ./s35
        1 2
        1 2 3
        1 2 3

30 readonly
    cat s36
    ./s36      # CONSTA ist readonly
    cat s37    # Wirkung in Subshells
    cat t1
    ./s37

30-35
    cat shellscript
    ./shellscript
    e
    . ./shellscript
    e
    type e
```

36

```
cat shellscript1    # mit Parameter
. ./shellscript1
ep p1 p2 p3
type ep

cat useshellscript1
  ./useschellscript1 p1 p2 p3

more trap1          # Aktionen in Interrupt-
                    # handler
    ./trap1
                    ^C
                    ^C
                    ^C
                    ^C
                    ^\
more traptest       # Traps treten nur bei Ende
                    # sleep auf
    ./traptest
    ^C, ^\
```