

# 1. Einführung

UNIX ist ein ausgereiftes, mächtiges Betriebssystem, das sich in zahlreichen Einsatzbereichen bestens bewährt hat. Es ist nicht so sehr eine einzelne Idee oder ein einzelnes Programm, das UNIX so attraktiv macht, sondern die *Art und Weise*, wie mit dem Betriebssystem und seinen zahlreichen Werkzeugen gearbeitet wird.

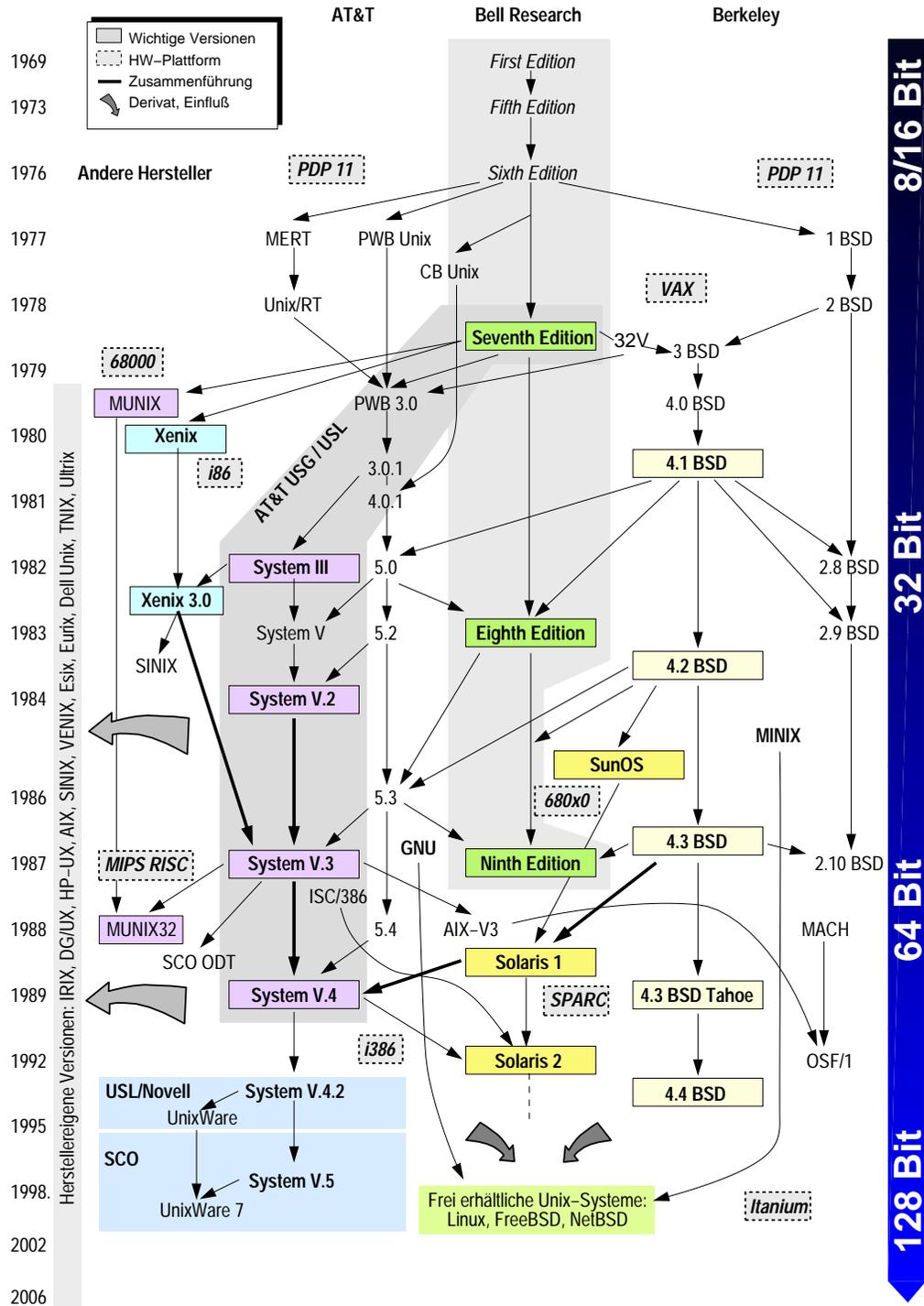
Das Geheimnis der Flexibilität und Effizienz von UNIX liegt in der Verbindung und dem Zusammenspiel des Systems und seiner Tools, also der darauf ablaufenden Programme. Viele dieser Programme erfüllen nur eine einzige Aufgabe — nicht mehr und nicht weniger — aber diese dafür höchst zuverlässig und so effizient wie möglich. Eine Aufgabenstellung wird nun in Unterfunktionen zerlegt und diese werden jeweils von den dafür geeigneten Programmen übernommen. In der Summe ergibt sich so die Lösung einer beliebig komplexen Aufgabenstellung auf Basis vieler einfacher Werkzeuge und einiger wohldurchdachter, sorgfältig implementierter Mechanismen, die das optimale Zusammenspiel dieser Werkzeuge garantieren.

Es ist daher nicht nur wichtig, daß UNIX-Benutzer wissen, welche Werkzeuge es in einem UNIX-System gibt, sondern auch, wie diese Werkzeuge effizient eingesetzt und miteinander kombiniert werden können.

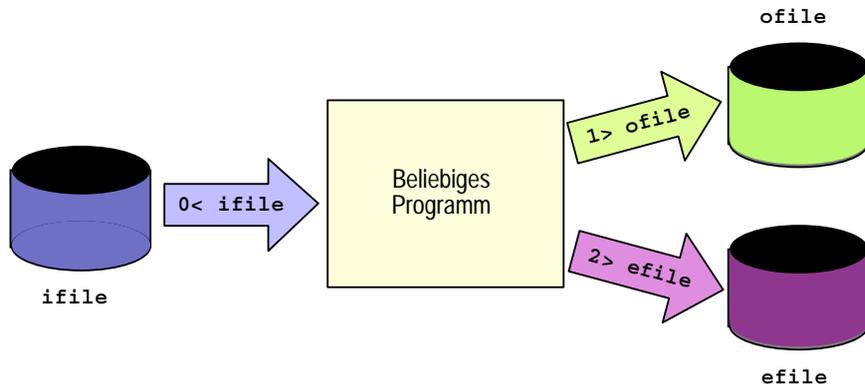
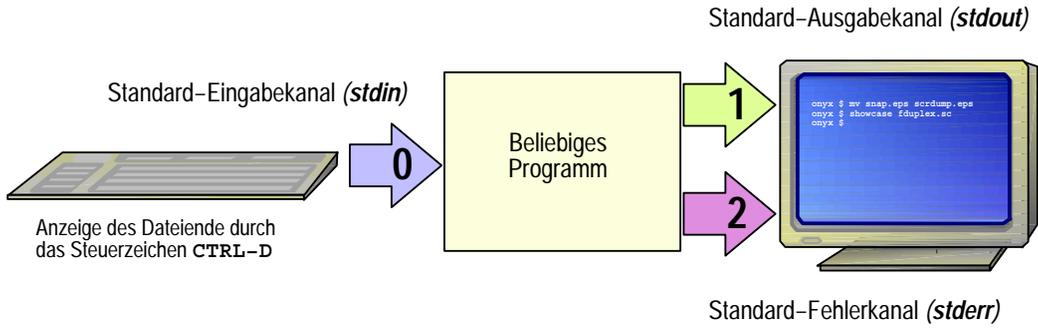
Die Themen dieses Kapitels:

- Wie alles begann ...
- Der Versionen-Stammbaum
- Moderne UNIX-Versionen
- Betriebssystemvarianten
- Benutzeroberflächen
- Systemeigenschaften und Schichtenmodell von UNIX
- Die Aufgabenverteilung bei UNIX
- Das Benutzerkonzept von UNIX

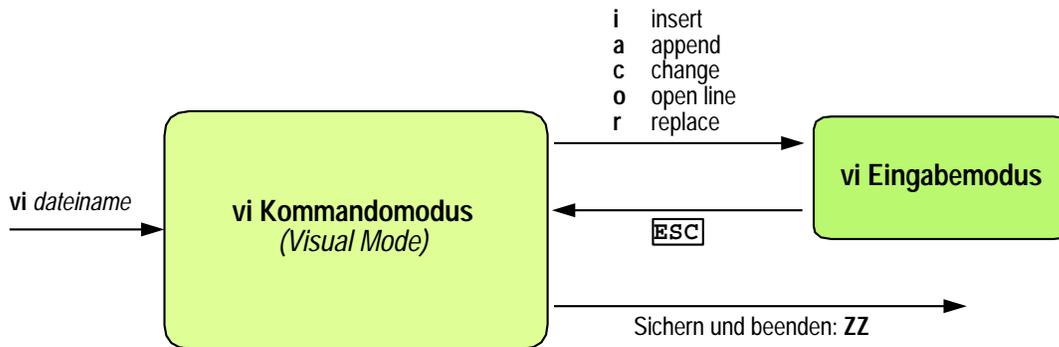
### Versionen-Stammbaum [5]



## Umlenkung der Standardkanäle [45]



## Der Texteditor vi [61]



### Cursorbewegungen

**h** ein Zeichen nach links  
**j** eine Zeile nach unten  
**k** eine Zeile nach oben  
**l** ein Zeichen nach rechts  
**w** ein Wort vorwärts  
**b** ein Wort rückwärts  
**^** zum Zeilenanfang  
**\$** zum Zeilenende  
**nG** zur Zeile Nummer *n*

### Suchen im Text

**/ pat/** *pat* nach vorne suchen  
**? pat?** *pat* nach hinten suchen  
**n** Suche wiederholen  
**N** Dito, in umgekehrter Richtung

### Löschen (Grundbefehl: **d**)

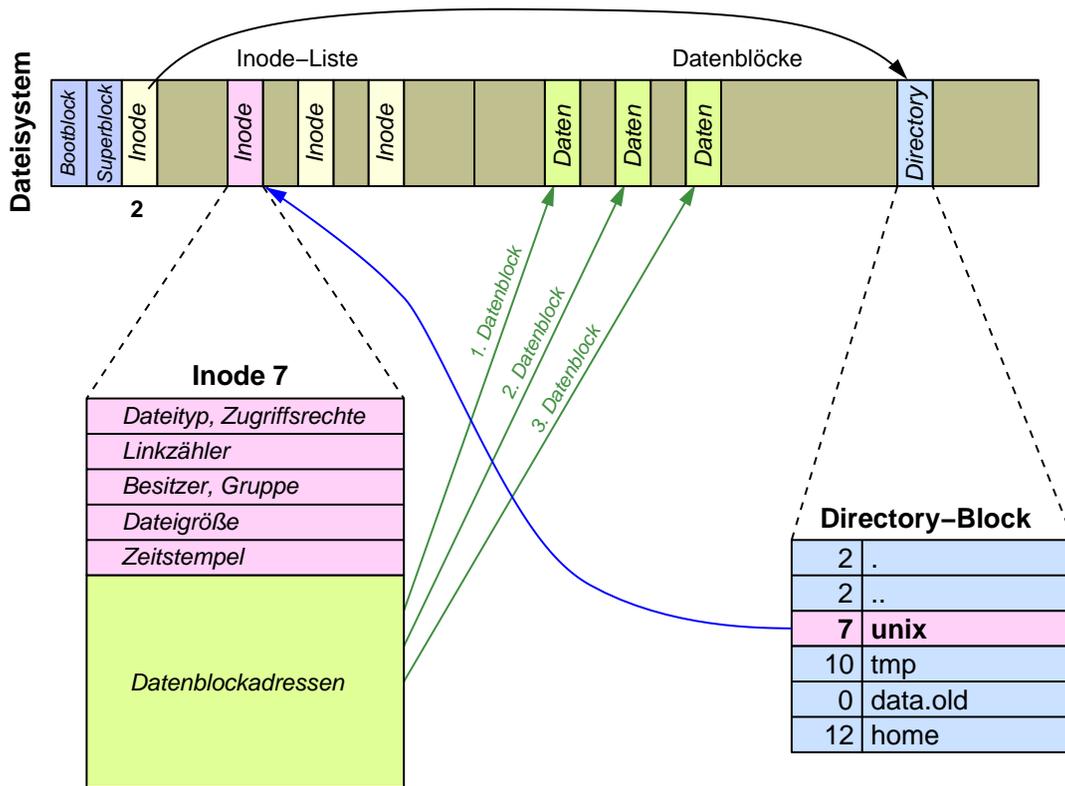
**dh** Zeichen nach links  
**dj** Zeile nach unten  
**dk** Zeile nach oben  
**dl** Zeichen nach rechts  
**dw** ein Wort vorwärts  
**db** ein Wort zurück  
**d^** bis Zeilenanfang  
**d\$** bis Zeilenende  
**d nG** bis Zeile *n*  
**ndG** bis Zeile *n*

**d / pat/** bis zum ersten Muster *pat* nach vorne  
**d ? pat?** bis zur ersten Muster *pat* nach hinten

## Aufbau und Gliederung von Manpages [67]

```
$ man man
MAN(1) MAN(1)
NAME
man - print entries from the on-line reference manuals; find manual
entries by keyword
SYNOPSIS
man [-cdwTpr] [-M path] [-T macropackage] [section] title ...
man [-M path] -k keyword ...
man [-M path] -f filename ...
DESCRIPTION
man locates and prints the titled entries from the on-line reference
manuals. man also prints summaries of manual entries selected by keyword
or by associated filename.
...
OPTIONS
-M path Use path as the search path for manual entries. path
is a colon-separated list of directories where manual
subdirectories may be found. The default path is
/usr/share/catman:/usr/share/man:/usr/catman:/usr/man.
...
ENVIRONMENT
MANPATH If set, MANPATH overrides the default manual entry search path.
MANPATH is a colon-separated list of directories where manual
subdirectories may be found. (See the discussion of -M.) -M
and -d will override MANPATH.
...
FILES
/usr/share/catman root directory of on-line reference manual
/usr/share/catman/whatis table of contents and keyword database
...
SEE ALSO
apropos(1), col(1), compress(1), csh(1), eqn(1), grep(1), gzip(1), lp(1),
lpr(1), makewhatis(1M), more(1), neqn(1), nroff(1), pack(1), pcat(1),
psroff(1), sh(1), tbl(1), troff(1), ul(1), whatis(1), zcat(1), regex(3X),
ascii(5), environ(5), man(5), term(5).
CAVEATS
apropos (man -k) and whatis (man -f) require that the whatis database be
kept up to date by running makewhatis(1M) after installing or removing
software from the system. This command is run by default on system
reboot when software is installed or removed by the inst(1m) command.
...
```

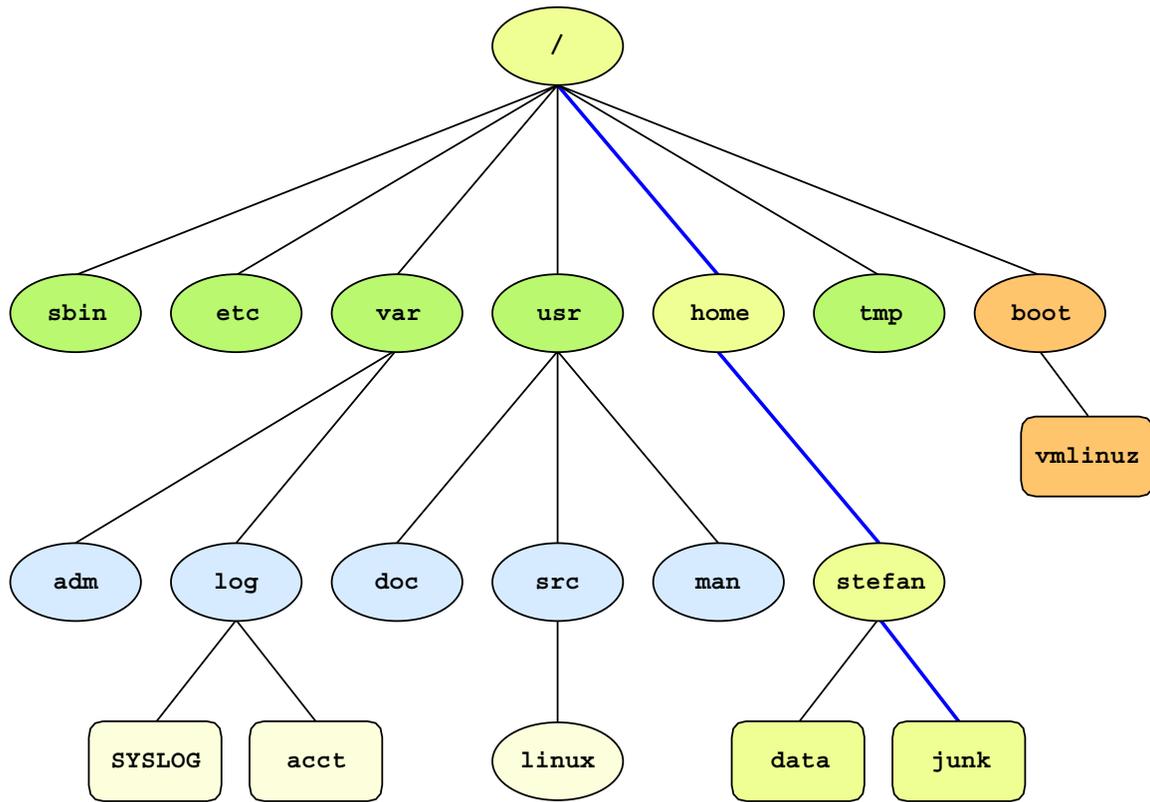
## Implementation von Directories [81]



## Dateiname (Link) entfernen [105]

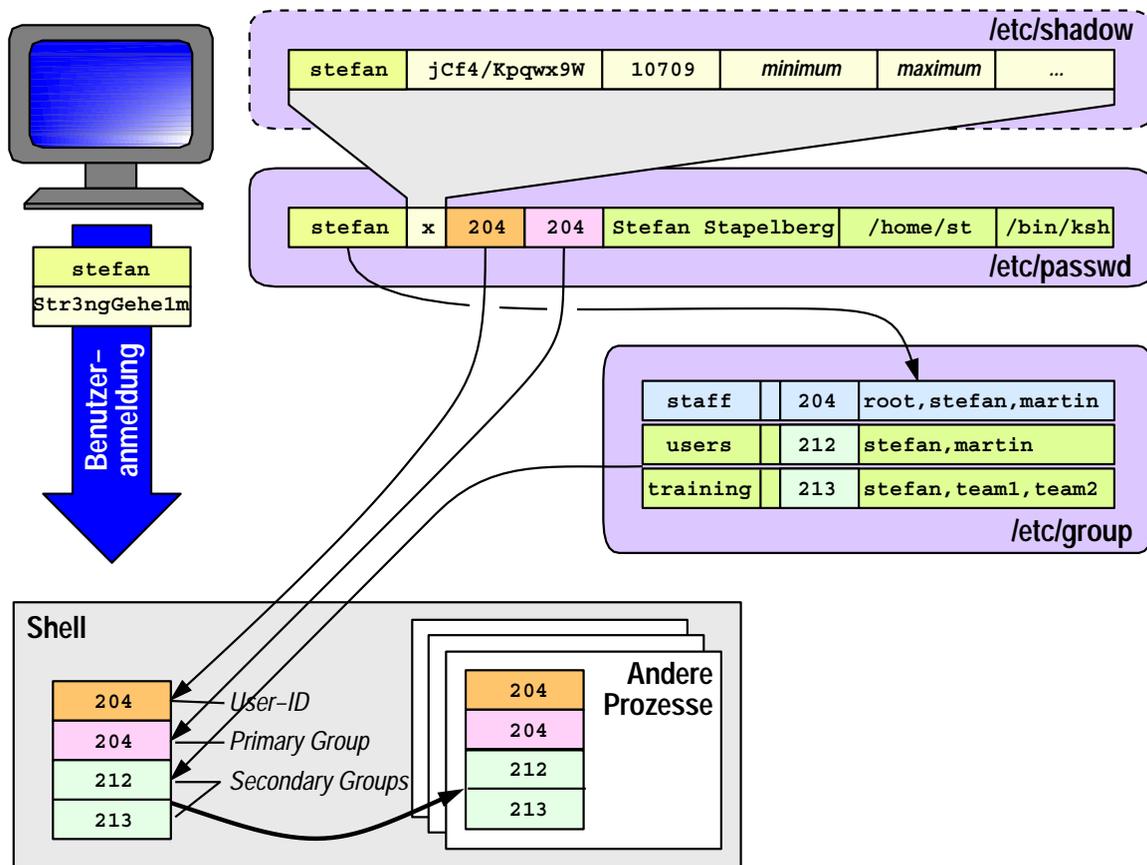
```
$ ls -l
total 2
-rw-r--r--  1 stefan  staff      156 Nov 21 12:37 data
-rw-r--r--  2 stefan  staff      156 Nov 21 12:43 newfile
-rw-r--r--  2 stefan  staff      156 Nov 21 12:43 temp
-rw-r--r--  1 stefan  staff      156 Nov 21 12:46 temp.old
-r--r--r--  1 stefan  staff         3 Nov 21 12:53 testfile
lrwxr-xr-x  1 stefan  staff         6 Nov 21 16:42 softfile -> nofile
$
$ rm testfile temp temp.old
testfile: 444 mode. Remove ? (yes/no) [no] : y
$
$ rm softfile nofile
Cannot access nofile: No such file or directory
$
$ ls
data  newfile
$
```

### Die Directory-Hierarchie von Linux [169]

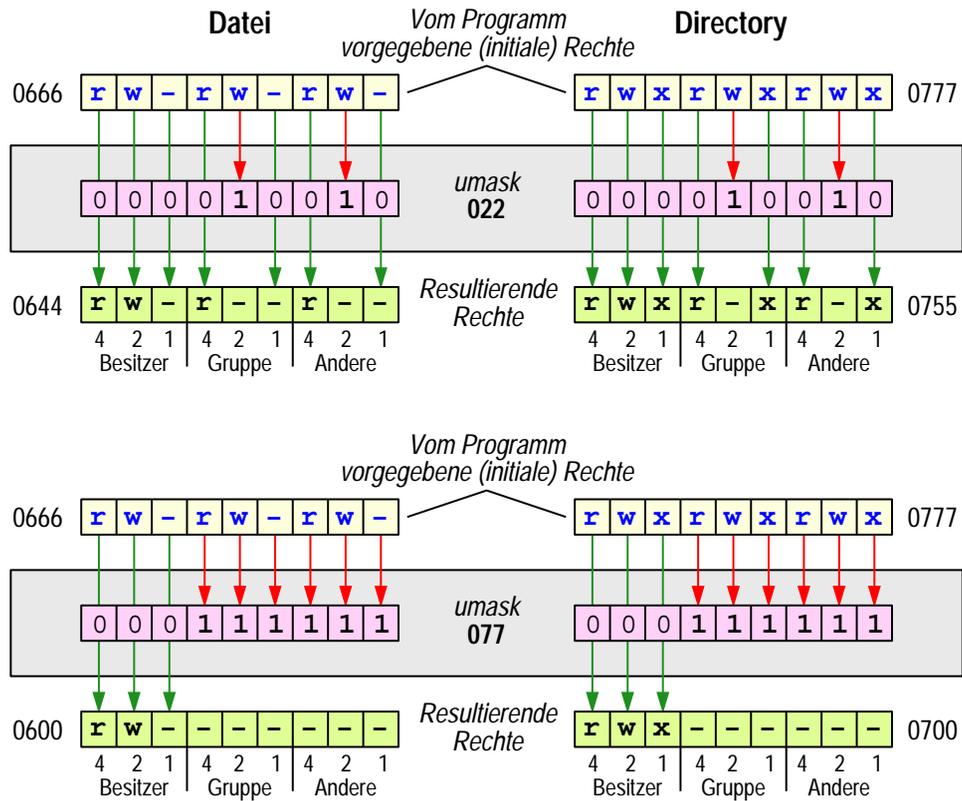


## Benutzer- und Gruppennummer [175]

```
login: stefan
Password: Str3ngGehe1m!
$
$ id
uid=204(stefan) gid=204(staff)
$ groups
staff users training
$
```



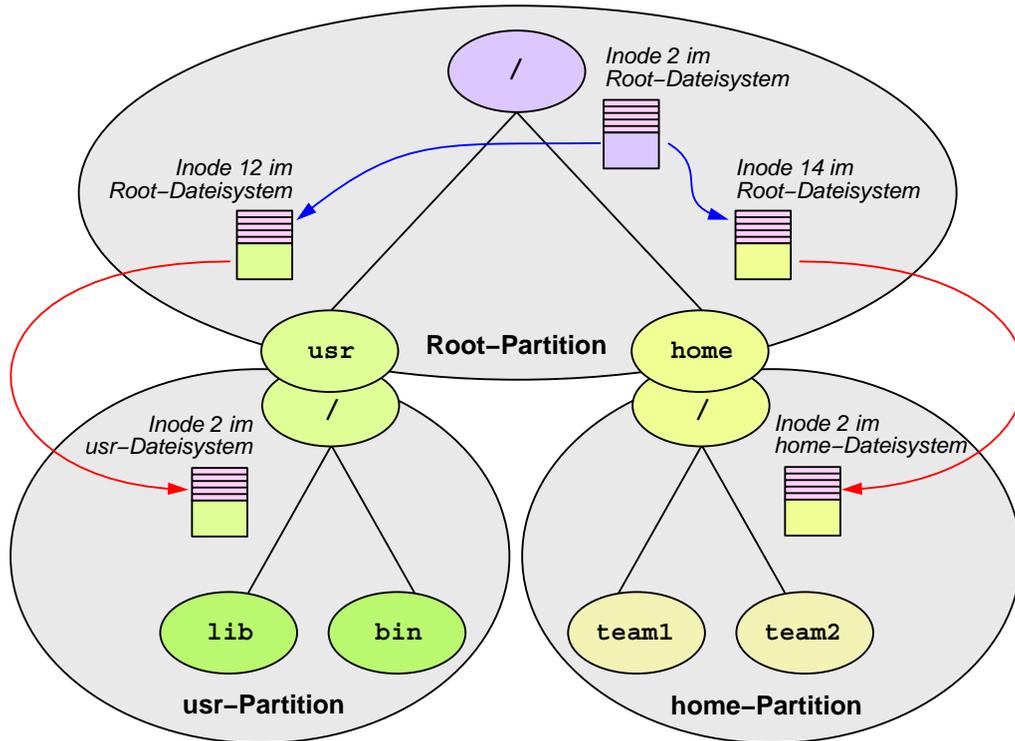
## Modifikation von initialen Zugriffsrechten durch die umask [187]



```

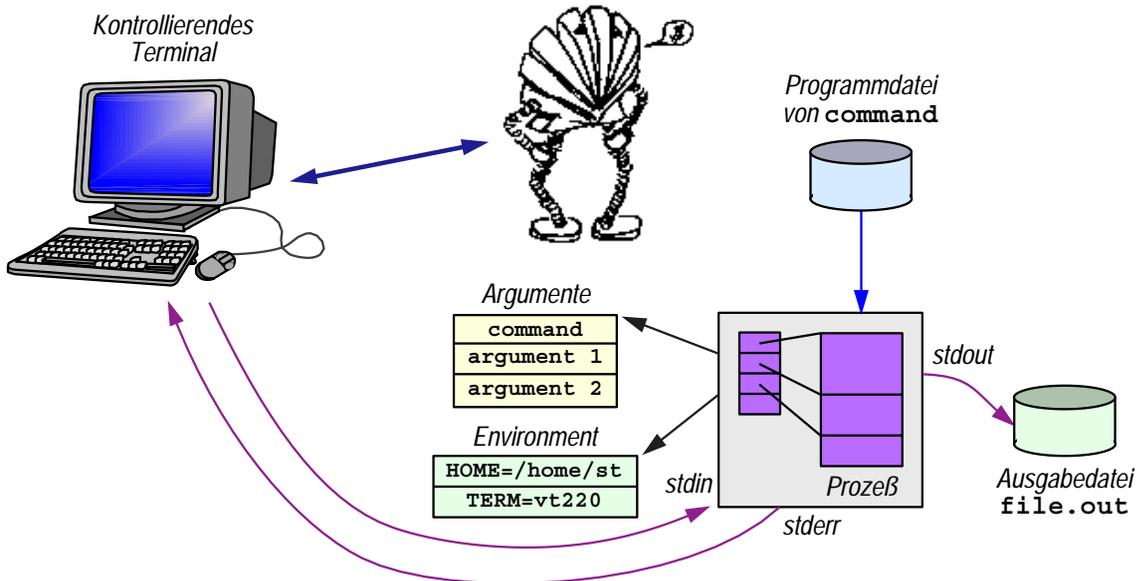
$ echo Hi >temp
$ ls -l temp
-rw-r--r--  1 stefan  staff          3 Jul  4 02:16 temp
$ umask
022
$ umask 0
$ echo Hi >junk
$ ls -l junk
-rw-rw-rw-  1 stefan  staff          3 Jul  4 02:16 junk
$
    
```

### Montierte Dateisysteme [201]



## Aufgaben eines Kommandointerpreters [211]

```
command argument\ 1 "argument 2" >file.out
```

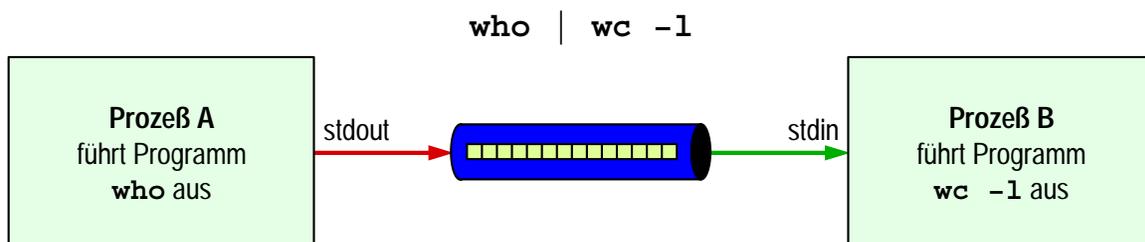


## Pipelines [221]

```

$ echo "Anzahl der Benutzer: \c"; who | wc -l
Anzahl der Benutzer: 3
$
$ find $HOME -type f -print 2>/dev/null | wc -l
243
$
$ ls -l | sort +4nr
-rw-r--r--  1 stefan  staff      690 Nov 22  2002 list
-rw-r--r--  1 stefan  staff      176 Nov 22  2002 adr-list
-rw-r--r--  1 stefan  staff      156 Nov 21  2002 data
-rw-r--r--  1 stefan  staff      101 Nov 22  2002 phone-num
-rw-r--r--  1 stefan  staff       74 Nov 22  2002 tempfile
lrwxr-xr-x  1 stefan  staff         4 Jun 28  15:05 new-data -> data
$
$ ls subd1 >s1.list
$ ls subd2 | diff s1.list -
2c2
< mydata
---
> myfile
$ rm s1.list
$

```



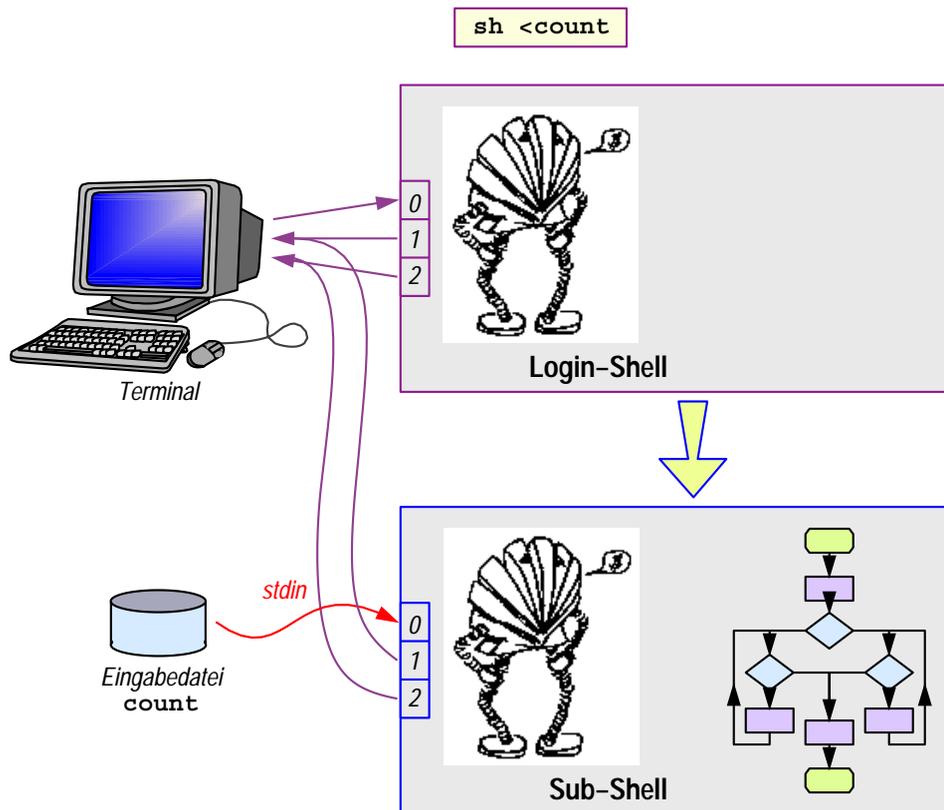
## Kommandosubstitution [245]

```

$ echo Heute ist `date`
Heute ist Wed Jul 07 12:26:12 CEST 2004
$
$ echo "Am System arbeiten
> derzeit `who | wc -l` Benutzer."
Am System arbeiten
derzeit      10 Benutzer.
$ echo "Am System arbeiten
> derzeit" `who | wc -l` Benutzer.
Am System arbeiten
derzeit 10 Benutzer.
$
$ cat temp
Microsoft spel checker for sail, worgs grate!
$ echo temp >f3
$ echo f3 >f2
$ echo f2 >f1
$ cat `cat `cat \\`cat f1\\`
Microsoft spel checker for sail, worgs grate!
$
$ cat $(cat $(cat $(cat f1)))
Microsoft spel checker for sail, worgs grate!
$ cat $(<f3)
Microsoft spel checker for sail, worgs grate!
$
$ echo $(ls -l)
total 48 -rw-r--r-- 1 stefan staff 3 Jul 7 12:22 f1 -rw-r--r-- 1 stefan staff 3
Jul 7 12:22 f2 -rw-r--r-- 1 stefan staff 5 Jul 7 12:22 f3 -rwxrw-r--+ 1 stefan
staff 3 Jul 4 02:16 junk -rw-r--r-- 1 stefan staff 46 Jul 4 03:48 temp
$ echo "$(ls -l)"
total 48
-rw-r--r--    1 stefan  staff          3 Jul  7 12:22 f1
-rw-r--r--    1 stefan  staff          3 Jul  7 12:22 f2
-rw-r--r--    1 stefan  staff          5 Jul  7 12:22 f3
-rwxrw-r--+   1 stefan  staff          3 Jul  4 02:16 junk
-rw-r--r--    1 stefan  staff         46 Jul  4 03:48 temp
$

```

## Die Shell als Programmiersprache [255]



```
$ echo who \| wc -l >count
$ cat count
who | wc -l
$ sh <count
    10
$ sh count
    10
$ chmod +x count
$ count
    10
$
```

## Tk-Skript: GUI für die Tabelle des Einmaleins [309]

```

# Definition der graphischen Objekte und
# Verknüpfung mit den Programmfunktionen
frame .oben -bg yellow

label .oben.zl -text "Zeilen:" -bg yellow
entry .oben.ze -width 5 -textvariable zeilen

label .oben.sl -text "Spalten:" -bg yellow
entry .oben.se -width 5 -textvariable spalten

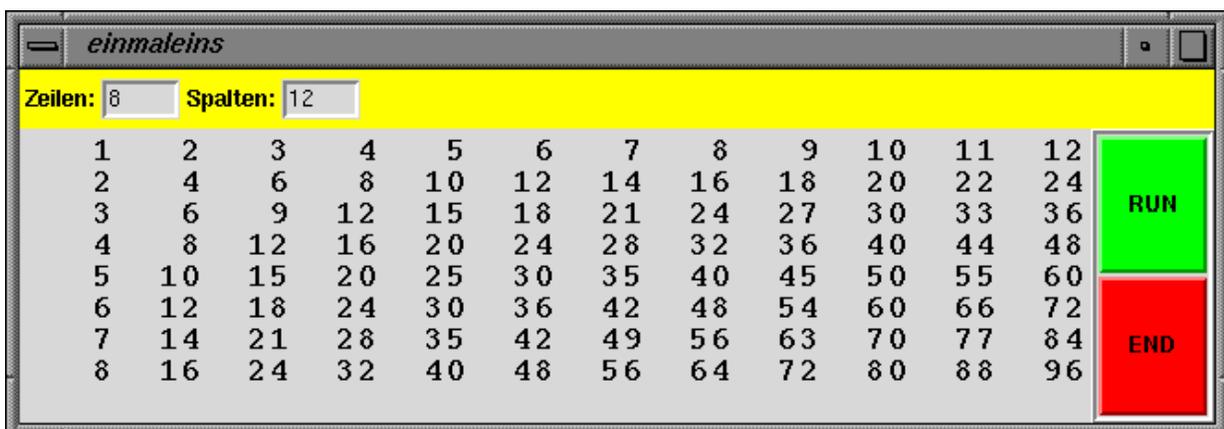
frame .rechts -relief ridge -borderwidth 3
button .rechts.br -text RUN -bg green -command einmaleins
button .rechts.be -text END -bg red -command exit

label .tab -font fixed -textvariable tabelle

# Visuelle Anordnung und Darstellung
pack .oben -ipadx 5 -ipady 5 -fill x -side top
pack .oben.zl .oben.ze .oben.sl .oben.se -side left

pack .rechts -fill y -side right
pack .rechts.br .rechts.be -fill both -expand 1 -side top
pack .tab -fill both -expand 1

```



# INHALTSVERZEICHNIS

1. Einführung	1
1.1 Wie alles begann ...	2
1.2 Der Versionen-Stammbaum	4
1.3 Moderne UNIX-Versionen	6
1.4 Betriebssystemvarianten	10
1.5 Benutzeroberflächen	12
1.6 Systemeigenschaften und Schichtenmodell von UNIX	20
1.7 Die Aufgabenverteilung bei UNIX	22
1.8 Das Benutzerkonzept von UNIX	24
1.9 Zusammenfassung	26
2. Grundlagen und Basis-Kommandos	27
2.1 Die Anmeldung am System	28
2.2 Terminal-Ein-/Ausgabe	32
2.3 Übersicht über die Shell	38
2.4 Regeln für Dateinamen	50
2.5 Übersicht über die Editoren	52
2.6 Die UNIX-Dokumentation	64
2.7 Dateien und Basiskommandos	72
2.8 Benutzerkommunikation	106
2.9 Zusammenfassung	124
3. Das UNIX-Dateisystem	125
3.1 Dateibezogene Kommandos	126
3.2 Directories und die Dateisystem-Hierarchie	152
3.3 Zugriffsrechte bei Dateien und Directories	174
3.4 Gerätedateien	192
3.5 Montierte Dateisysteme	198
3.6 Das Drucksystem von UNIX	202
3.7 Zusammenfassung	206
4. Die Shell	207
4.1 Übersicht über die Shells	208
4.2 Die Shell als Kommandointerpreter	210
4.3 Quoting-Mechanismen	214
4.4 Kommandoseparation	216
4.5 Pipelines	220
4.6 Asynchrone Programmausführung	226
4.7 Shell-Variablen	234
4.8 Kommandosubstitution	244
4.9 Weitere Konstrukte zur Ein-/Ausgabe-Umlenkung	246
4.10 Weitere Dateinamenmuster	248
4.11 Alias-Mechanismus	250
4.12 Die Anlaufdatei .profile	252
4.13 Die Shell als Programmiersprache	254
4.14 Anweisungen zur Ablaufsteuerung	262
4.15 Bedingungstests	274
4.16 Shell-Arithmetik	276
4.17 Weitere Shell-Konstrukte	280
4.18 Zusammenfassung	290

5. Weitere Tools . . . . .	291
5.1 Textbearbeitung . . . . .	292
5.2 Textverarbeitung . . . . .	300
5.3 Software-Entwicklung mit UNIX . . . . .	302
5.4 Graphische Oberflächen mit Tcl/Tk . . . . .	304
5.5 Zusammenfassung . . . . .	310
A. Link-Verzeichnis . . . . .	311
A.1 Interessante Web-Sites . . . . .	313
A.2 DOS-Kommandos — UNIX-Kommandos . . . . .	314
B. Übungen . . . . .	315
B.1 Übungen zum Kapitel Einführung . . . . .	317
B.2 Übungen zum Kapitel Dateisystem . . . . .	319
B.3 Übungen zum Kapitel Shell . . . . .	321
C. Musterlösungen . . . . .	323
C.1 Musterlösungen zum Kapitel Einführung . . . . .	325
C.2 Musterlösungen zum Kapitel Dateisystem . . . . .	329
C.3 Musterlösungen zum Kapitel Shell . . . . .	333