

Die Shell

- Kurzakte -

Die Shell ist:

- ein Kommandozeileninterpreter
- eine Benutzerumgebung
- eine Programmiersprache

Zu den Hauptaufgaben von Shells gehören:

- Analyse der Befehlseingabe (Syntax)
- Suche nach Befehlen
- Ersetzen von Sonderzeichen
- Steuerung der Ein-/Ausgabe
- Steuerung des Ablaufs des Befehls
- Erzeugung von Vorder- und Hintergrundprozessen
- Prüfung der Zugriffsberechtigungen
- Erzeugen von Subshells (wiederholter Aufruf der Shell als eigenständiger Prozess)

Arten von Shells:

Shell	Aufruf	Beschreibung	History	Alias	Prompt
Bourne-Shell	sh	kleinster Kdo.umfang	-	-	\$
C-Shell	csh	C-Syntax	x	x	%
Korn-Shell	ksh	sh+csh	x	x	\$
T-Shell	tcsh	erweiterte csh			
Z-Shell	zsh	sh, tcsh, bash			
Bourne-Again	bash	sh	x	x	>

Sonderzeichen der Shell:

Zeichen	Bedeutung	sh	t-, csh	ksh, bash
*	ein Zeichen	x	x	x
?	mehrere Zeichen	x	x	x
[]	eins dieser Zeichen	x	x	x
[!]	keins dieser Zeichen	x	-	x
#	Kommentar	x	x	x
;	Befehlstrenner	x	x	x
&	Hintergrundbefehl	nur jsh	x	x
()	Befehl in Subshell	x	x	x
\$	liest Variablenwert	x	x	x
\	maskiert ein Sonderzeichen	x	x	x
' '	maskiert alle Sonderzeichen	x	x	x
" "	maskiert fast alle Sonderz.	x	x	x
` `	Kommandosubstitution	x	x	x
~	Heimatverzeichnis	-	x	x
-	Verzeichniswechsel	-	-	x
>	Ausgabeumlenkung	x	x	x
<	Eingabeumlenkung	x	x	x
2>	Fehlerausgabeumlenkung	x	-	x
	Pipe	x	x	x

Möglichkeiten der Kommandoeingabe:

Schreibweise von Kdos.	Bedeutung
befehl &	Der Befehl wird im Hintergrund ausgeführt.
befehl1 ; befehl2	Mehrere Befehle werden in einer Befehlszeile hintereinander ausgeführt.
(befehl1 ; befehl2)	Die Befehle werden hintereinander in einer Subshell ausgeführt.
befehl1 befehl2	Der erste Befehl übergibt mit Hilfe des Pipe-Symbols seine Ausgabe als Eingabe an den zweiten Befehl.
befehl1 `befehl2`	Beim zweiten Befehl wird eine Befehlssubstitution durchgeführt und seine Ausgabe wird dem ersten Befehl als Argument übergeben.
befehl1 \$(befehl2)	Beim zweiten Befehl wird eine Befehlssubstitution durchgeführt, und seine Ausgabe wird dem ersten Befehl als Argument übergeben.
befehl1 && befehl2	Bei dieser UND-Verknüpfung wird der zweite Befehl nur ausgeführt, wenn der erste Befehl erfolgreich war.
befehl1 befehl2	Bei dieser ODER-Verknüpfung wird der zweite Befehl nur ausgeführt, wenn der erste Befehl fehlgeschlagen ist.
{ befehl1 ; befehl2 }	Die Befehle werden in der laufenden Shell ausgeführt.

Konfigurationsdateien:

	Login-Shell	Nicht-Login-Shell	grafische Shell
Start			
systemübergreifend	/etc/profile	/etc/bashrc	
benutzerbezogen	~/.bash_profile ~/.bash_login ~/.profile	~/.bashrc	~/.xsession
Beenden			
benutzerbezogen	~/.bash_logout		

Jobkontrolle:

Befehl	Wert
Jobs	Alle im Moment aktiven Jobs ausgeben
fg	Den zuletzt in den Hintergrund gestellten Job in den Vordergrund stellen
fg %n	Einen Job mit der Job-ID n in den Vordergrund stellen
(Ctrl)+(z)	Einen Vordergrundjob stoppen
bg	Den zuletzt gestoppten Job im Hintergrund wieder starten
bg %n	Einen gestoppten Job mit der Job-ID n im Hintergrund starten
stop %n	Einen Job mit der Job-ID n im Hintergrund stoppen
kill %n	Einen Job mit der Job-ID n im Hintergrund beenden
(enter)	Einen beendeten Hintergrundjob anzeigen

```

notebookneu:~ # sleep 500 &
[1] 6023
notebookneu:~ # sleep 1000 &
[2] 6024
notebookneu:~ # sleep 5000 &
[3] 6025
notebookneu:~ # jobs
[1]  Running          sleep 500 &
[2]-  Running          sleep 1000 &
[3]+  Running          sleep 5000 &

```

In der ersten Spalte steht die laufende Hintergrundjobnummer, in der zweiten Spalte ein Pluszeichen für den zuletzt in den Hintergrund gestellten Job oder ein Minuszeichen für den davor in den Hintergrund gestellten Job. Alle bereits früher im Hintergrund gestarteten Jobs haben kein Kennzeichen.

```

notebookneu:~ # fg
sleep 5000

```

