Shell — Umgebungsvariablen, Aliase und Konfigurationsdateien

Klaus Knopper <shell@knopper.net>

Version 1.0 (4. August 1999)

Den meisten Shells unter Unix gemeinsam sind die folgenden Umgebungsvariablen. Diese sind i.d.R. nach der Anmeldung (Einloggen) schon auf sinnvolle Werte gesetzt und bestimmen das Verhalten vieler Programme und Shell-interner Kommandos.

Variable	Bedeutung
`$HOME`	Heimatverzeichnis des Benutzers, in das bei `cd` (ohne Angabe von Parametern) und beim Einloggen gewechselt wird.
`$PATH`	Suchpfad für Programme; durch **:** getrennte Liste von Verzeichnissen.
`$TERM`	Typ der aktuellen Textkonsole, wird beispielsweise von `vi` und bei der Farbdarstellung des GNU-`ls`-Kommandos ausgewertet.
`$SHELL`	Name der aktuellen Shell.
`$USER`	Benutzerkennung.

Der Benutzer kann diese und alle anderen Umgebungsvariablen selbst setzen oder verändern. Hierbei unterscheiden sich jedoch die Bourne-Shell-kompatiblen (`sh`, `ksh`, `bash`) von den C-Shell-kompatiblen (`csh`, `tcsh`) Shells in der Syntax.

Beispiel 1: Setzen des Kommandosuchpfades auf
`/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin:/opt/kde/bin:.`

```
bash: export PATH="/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin:/opt/kde/bin:."
tcsh: setenv PATH "/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin:/opt/kde/bin:."
```

Beispiel 2: Setzen des Prompt auf **User@Rechner[aktuelles Verzeichnis]**.

```
bash: export PS1="\u@\h[\w]\\$ "
tcsh: set prompt="%n@%m[%~] "
```

Merke: Beim Zugriff auf eine Variable (`echo $PATH`) wird der Variablen ein **$** vorangestellt, beim Setzen der Variablen jedoch nicht!

Welche speziellen Umgebungsvariablen welche Wirkung haben, und die genaue Definition der Inhalte von Shell-Variablen, ist in den sehr ausführlichen `man`-Pages der entsprechenden Shell nachzulesen.

Um sich für häufig benutzte Kommandosequenzen oder Optionen Tipparbeit zu sparen, kann mit Hilfe von **Aliasen** eine Abkürzung für ein Kommando oder eine Kommandosequenz definiert werden.

Beispiele:

```
bash: alias dir='ls -la'
tcsh: alias dir 'ls -la'
bash: alias del='rm -i'
tcsh: alias del 'rm -i'
bash: alias suche='find / -name'
tcsh: alias suche 'find / -name'
```

`bash` und `(t)csh` lesen vor dem Start traditionell (per Option oder beim Übersetzen aus dem Quelltext konfigurierbar) folgende Dateien ein:

bash	
`/etc/profile`	systemweit, beim Login auf der Textkonsole
`/etc/bashrc`	systemweit, bei jedem Aufruf der Shell
`$HOME/.profile`	User-spezifisch, beim Login auf der Textkonsole
`$HOME/.bashrc`	User-spezifisch, bei jedem Aufruf der Shell
tcsh	
`/etc/csh.login`	systemweit, beim Login auf der Textkonsole
`/etc/csh.cshrc`	systemweit, bei jedem Aufruf der Shell
`$HOME/.login`	User-spezifisch, beim Login auf der Textkonsole
`$HOME/.cshrc`	User-spezifisch, bei jedem Aufruf der Shell

Bei den meisten Unix-Derivaten kann sich der Benutzer seine Standard-Shell (Login-Shell) frei aussuchen, und mit Hilfe des `chsh`-Kommandos selber setzen.

Mit Hilfe der Shell als Programmiersprache in sog. **Shell-Scripten** lassen sich auch komplexe, häufig wiederkehrende Aufgaben vereinfachen.

Beispiel: Mehmaliges Kopieren einer Daten-CD.

```
#!/bin/sh

antwort="j"
while [ "$antwort" != "n" ]
do
 cdrecord -v dev=0,1,0 speed=4 -isosize /dev/cdrom
 echo "Bitte nächste CD in das CD-Rom Laufwerk, einen neuen Rohling"
 echo "in den CD-Brenner einlegen und <ENTER> drücken."
 echo -n "Eingabe ('n' für Abbruch) > "
 read antwort
done
```

Die meisten Subsysteme (Netzwerk, X-Window Sessions, NFS, ...) unter Unix werden mit Hilfe von Shell-Scripten gestartet.[1]

[1] Siehe Thema **init**.