

HOWTO zum Bash-Eingabeprompt
 (C) 2006 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
 OSTC GmbH, <http://www.ostc.de>
 \$Id: bashprompt-HOWTO.txt,v 1.6 2011-04-16 16:32:39 tsbirn Exp \$

Dieses Dokument beschreibt die Konfiguration des Bash-Eingabeprompts.

Inhaltsverzeichnis

- 1) Primärer Prompt "\$PS1"
- 2) Sonderzeichen in den Prompt-Variablen
- 3) Hinweise
- 4) Weitere Prompte-Variable

1) Primärer Prompt "\$PS1"

Die Variable "\$PS1" legt das Aussehen des Eingabeprompts (primärer Prompt, Eingabeaufforderung) fest, der immer dann ausgegeben wird, wenn die Shell auf die Eingabe eines Befehls wartet. Standardwert ist:

```
PS1="\s-\v\$ "
```

Sie wird für alle Benutzer gemeinsam in "/etc/profile" (zuerst) oder für einzelne Benutzer in "~/.profile" (Login-Shell) bzw. "~/.bashrc" (interaktive Shell) gesetzt.

2) Sonderzeichen in den Prompt-Variablen

Folgende Sonderzeichen in "\$PS1" (und den anderen Prompt-Variablen) setzen spezielle (variable) Komponenten in den Prompt ein:

Zeichen	Bedeutung
\a	Klingeln [alert]
\e	ESC-Zeichen [escape]
\n	Zeilenumbruch [newline]
\r	Wagenrücklauf [carriage return]
\d	Datum in der Form "Sun Dec 24" [date]
\D{FMT}	Datum formatiert gemäß "strftime(3)"-Format FMT
\t	24h-Zeit in der Form "HH:MM:SS" [time]
\T	12h-Zeit in der Form "HH:MM:SS" [Time]
\A	24h-Zeit in der Form "HH:MM" [time]
\@	12h-Zeit in der Form "HH:MM AM/PM" [time]
\h	Rechner-Name bis zum ersten "." [host]
\H	Vollständiger Rechner-Name (FQHN) [Host]
\j	Anzahl Hintergrundprozesse [jobs]
\l	Terminalname (Teil nach letztem "/") [1]
\u	Benutzer-Name [user]
\w	Vollständiges Arbeitsverzeichnis [working directory]
\W	Arbeitsverzeichnis (Teil nach letztem "/") [Working directory]
\s	Shell-Name (basename \$0) [shell]
\v	Bash-Version [version]
\V	Bash-Version (inkl. Release und Patchlevel) [Version]
\\$	"\$"-Zeichen für normalen Benutzer, "#"-Zeichen für root
\#	Befehlszeilennummer (in aktueller Sitzung)
\!	Befehlszeilennummer (über Sitzungen hinweg)
\[Beginn nicht darstellbare Zeichen (bei Kommandolänge ignoriert!)
\]	Ende nicht darstellbare Zeichen (bei Kommandolänge ignoriert!)
\\	Backslash

Der Standardwert unter Linux beträgt für die root:

```
PS1="\h:\w # " # Host + Arbeitsverzeichnis + "#" + Leerzeichen(!)
```

und für normale Benutzer:

```
PS1="\u@\h:\w > " # User + HOST + Arbeitsverzeichnis + ">" + Leerzeichen(!)
```

3) Hinweise

* Das Leerzeichen am Prompt-Ende ist wichtig, damit die Kommandoingabe etwas abgesetzt vom Prompt beginnt.

* Der Prompt sollte auch nicht zu lang sein, da er sonst unübersichtlich wird und zu wenig Platz für die Eingabe der Kommandozeile übrig lässt.

* Einen inversen Prompt kann man durch Einsatz des Kommandos "tput" zur Ansteuerung von Terminal-Attributen erreichen. Dazu ist die Ausgabe des "tput"-Kommandos per Kommando-Substitution `...` (in jeder Shell bzw. \$(...) nur in der Bash) in die Prompt-Definition einzufügen:

```
PS1="\[$(tput smso)\]\u@\h:\w \! >\[$(tput rmso)\] "
```

"tput smso" schaltet den inversen Modus ein (set mode stand out),
"tput rmso" schaltet ihn wieder ab (remove mode stand out).

Damit beim späteren Editieren von Kommandozeilen die von "tput" erzeugten ANSI-Steuersequenzen nicht als Zeichen mitgezählt werden, sollten sie in "\[...\]" eingeschlossen werden. Vergisst man diese Klammern, dann lässt sich beim Editieren eines Kommandos der Cursor nicht mehr richtig steuern.

* Analog lässt sich ein farbiger Prompt erzeugen (z.B. "tput setaf 7" = weiße Schrift, "tput setab 4" = blauer Hintergrund, "tput sgr0" = normale Schrift):

```
PS1="\[$(tput setaf 7)\$(tput setab 4)\]\u@\h:\w \! \[$(tput sgr0)\] "
```

* Soll nur der Name des aktuellen Verzeichnisses sichtbar sein (ohne Pfad davor), kann statt "\w" auch "\W" verwendet werden:

```
PS1="\[$(tput smso)\]\u@\h:\W \! \[$(tput rmso)\] "
```

* Soll der absolute Pfad sichtbar sein, kann statt "\w" auch "\\$(pwd)" verwendet werden (Kommando-Substitution):

```
PS1="\[$(tput smso)\]\u@\h:\$(pwd) \! \[$(tput rmso)\] "
```

4) Weitere Prompt-Variable

* Die Variable "\$PS2" legt das Bereitschaftszeichen fest, das immer dann ausgegeben wird, wenn Return gedrückt wird, die Shell das (mehrzeilige) Kommando aber noch nicht als abgeschlossen betrachtet (z.B. "for VAR in WERT1 WERT2 ..." <RETURN>). Standard ist:

```
PS2="> "
```

Erscheint überhaupt kein Prompt nach der Eingabe von <RETURN>, so wartet das aufgerufene Kommando meist auf Eingaben von stdin. Diese Eingabe kann mit "<Strg-D>" oder "." ABGESCHLOSSEN werden, mit "<Strg-C>" kann das Kommando ABGEBROCHEN werden.

* Die Variable "\$PS3" steht für den Prompt beim "select"-Kommando. Standard ist

```
PS3="#? "
```

* Die Variable "\$PS4" wird vor den Debug-Meldungen ausgegeben, die durch die Shell-Option "-x" [execute trace] ausgelöst werden. Standardwert ist

```
PS4="+"
```

* Der Inhalt der Variablen "\$PROMPT_COMMAND" wird jedesmal als Kommando ausgeführt, bevor ein Eingabe-Prompt angezeigt wird. Die folgende Belegung gibt z.B. nach jedem eingegeben Kommando dessen Exit-Status aus.

```
PROMPT_COMMAND="echo 'Exit-Status: $?'"
```

#-----