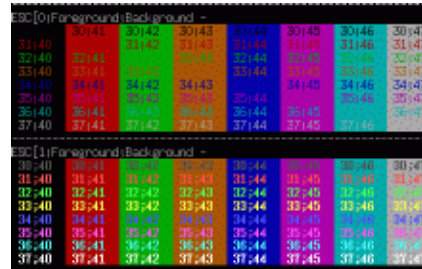


by Nico Golde  
 <nico/at/ngolde.de>

## Bunte Shells – Benutzung der Shellfarben



### Abstract:

In einem zum ANSI Terminal Standard kompatiblen Terminal (wie xterm, rxvt, konsole, ...) kann man Text nicht nur in schwarz/weiß darstellen. Dieser Artikel zeigt wie man Text in Fettdruck oder Farbe erhält.

### About the author:

Nico ist zur Zeit noch Schüler, beschäftigt sich seit einigen Jahren intensiv mit Linux und gehört zu mehreren Open Source Projekten.

## Allgemeines

Mit der bash kommt praktisch jeder Linux-Benutzer irgendwann in Berührung. Auf den ersten Blick sieht diese sehr langweilig aus, doch man hat viele Möglichkeiten seine Shell individuell zu gestalten. Farbliche Hervorhebungen des Shell-Promptes machen die Shell zum Einen individuell und zum Anderen gut lesbar.

In diesem Dokument werde ich mich auf die Shell bash beziehen. Die Escape Sequenzen können sich von Terminal zu Terminal unterscheiden, ich benutze für diesen Text ein ANSI Terminal.

## Konfiguration der Shell

Bei der bash ist die zentrale Konfigurationsdatei, die auch bei der Konfiguration der Shellfarbe zum Einsatz kommt, die Datei ~/.bashrc oder die globale bash-Konfigurationsdatei /etc/bashrc. Mittels der PS1 Variable in der bashrc werden die Einstellungen für das Aussehen der Shell verändert.

Normalerweise sollte dieser Eintrag folgende Form haben:

```
~/.bashrc: PS1="\s-\v\$ "
```

Wobei \s für den Namen der Shell und \v für die Version steht. Am Ende des Promptes wird dann ein \$ gesetzt. Da dies ein wenig langweilig ist, kann man stattdessen folgenden Eintrag verwenden, den die meisten Linux-Distributionen als Standardeinstellung haben:

```
~/.bashrc: PS1="\u@h \w \$ "
```

Dies steht für user@host aktuelles\_verzeichnis \$

Dies ist der normale Shell-Prompt, den die meisten Linux-Benutzer gewöhnt sein werden.

## Escape Sequenzen

Um diesem Prompt mittels verschiedener Farben eine persönliche Note zu verleihen, verwendet man nun Escape (engl: flüchten, hier aber: Sondersquenz) Sequenzen. Eine Escape Sequenz ist mindestens ein Steuerzeichen, das die Shell veranlasst etwas bestimmtes auszuführen. Eine Escape Sequenz wird normalerweise mittels `^[` eingeleitet. Diese Schreibweise ist etwas gewöhnungsbedürftig, `\033` hat den selben Effekt.

In der Shell kann man statt die Sequenz zu tippen, allerdings auch `Strg-V <ESC>` drücken.

## Farben der Shell benutzen

Die Farben der Shell werden zunächst an einem Beispiel-Prompt erklärt.

```
~/ .bashrc: PS1="\[\033[0;32;40m\u@\h:\w\$ \]"
```

Dies stellt den kompletten Prompt in Grün dar. Etwa so:

Dies stellt den kompletten Prompt in grün dar. `\033` leitet die Escape Sequenz ein, `[` eröffnet die Farbangabe. Die anschließende `0` gibt an, dass eine Normaldarstellung benutzt wird. Welche anderen Möglichkeiten man an dieser Stelle hat, wird später erwähnt. Der ganze String ist in `[` und `]` eingeschlossen, damit die Escape-Sequenzen nicht in die Ausgabe mit reinkommen und Platz auf der Shell wegnehmen.

Als nächstes wird die Vordergrundfarbe gewählt (in diesem Fall `32`, entspricht grün). Die Hintergrundfarbe `40` steht für die Farbe Schwarz. Möchte man in unserem Beispiel nicht, dass die Schrift nach dem Prompt auch grün ist, so fügt man an den Schluss die Escape Sequenz `\033[0m` an. Dies ist die Voreinstellung für die Shellfarbe. Sowohl für den Vordergrund als auch für den Hintergrund stehen 8 Farben zur Verfügung.

Auswahl: schwarz, rot, grün, gelb, blau, magenta, cyan und weiß. Die Zahlen dafür sind: `30`, `31`, `32`, `33`, `34`, `35`, `36`, `37`.

Das Setzen der Hintergrundfarbe verläuft genauso, allerdings statt `3` mit `4`. Also `40`, `41`, `42`, `43`, `44`, `45`, `46`, `47`.

Beispiel:

```
~/ .bashrc: PS1="\[\033[0;37;44m\u@\033[0;32;43m\h:\033[0;33;41m\w$\033[0m\]"
```

Das ergibt einen sehr bunten Prompt:

Am besten testet man diese Einstellungen per export PS1="string" in der Kommandozeile und kann die Einstellungen später in die bashrc eintragen. Der aktuelle Prompt des Autors sieht z.B. so aus:

```
PS1="\[\033[1;34;40m\[\033[1;31;40m\u@\h:\w\033[1;34;40m\]\033[1;37;40m $\033[0;37;0m\] "
```



## Eigenschaften der Schrift

Wie oben erwähnt ist die 0 direkt nach der ersten Escape Sequenz die Voreinstellung für die Schrift des Shell-Promptes. Für die Schrifteigenschaft sind die folgenden Werte sinnvoll: 0, 1, 22, 4, 24, 5, 25, 7, 27 und sie bedeuten: Standard, dick, nicht dick, unterstrichen, nicht unterstrichen, blinkend und nicht blinkend, invers, nicht invers.

Mit folgendem kleinen Shell Script kann man sich die Farbkombinationen ansehen.

```
#!/bin/sh
#####
# Nico Golde <nico@ngolde.de> Homepage: http://www.ngolde.de
# Letzte Änderung: Mon Feb 16 16:24:41 CET 2004
#####

for attr in 0 1 4 5 7 ; do
  echo "-----"
  printf "ESC[%s;Foreground;Background - \n" $attr
  for fore in 30 31 32 33 34 35 36 37; do
    for back in 40 41 42 43 44 45 46 47; do
      printf '\033[%s;%s;%sm %02s;%02s ' $attr $fore $back $fore $back
    done
    printf '\n'
  done
  printf '\033[0m'
done
```

Das Script gibt es auch hier als tar.gz zum herunterladen: [showansicol.tar.gz](http://www.ngolde.de/showansicol.tar.gz)

## Eine andere Form der Anwendung

Die Fähigkeit der Farbsetzung in der Shell macht nicht nur Sinn, wenn man einen schöneren Shell-Prompt haben möchte, sondern kann auch in der Programmierung eines Programmes für die Konsole sinnvoll sein.

So müsste man bei jeder Benutzung von Farben auf Bibliotheken wie slang oder ncurses zurückgreifen, was die Größe der Binärdatei beachtlich steigen ließe. Ncurses hat allerdings den Vorteil, dass es mehr oder weniger unabhängig vom Terminal-Typ ist.

## Beispiele in C

Ein Hello World mit grüner Schrift:

```
#include <stdio.h>
int main(void) {
    const char *const green = "\033[0;40;32m";
    const char *const normal = "\033[0m";
    printf("%sHello World%s\n", green, normal);
    return 0;
}
```

Eine weitere nützliche Escape Sequenz ist `printf("\033[2J");` Sie hat den selben Effekt wie `system(clear);` allerdings kann man damit auf die Headerdatei `unistd.h` verzichten.

Mit `printf("\033[1K");` lässt sich eine Zeile löschen.

## Beispiel für die init-Scripte

Möchte man nicht, dass beim Starten der init-Scripte in `/etc/init.d` nur ein `.` beim erfolgreichen Starten des Prozesses angezeigt wird, sondern statt dessen eine farbliche, besser lesbare Ausgabe, so ist dies auch mittels einer Escape-Sequenz lösbar.

Hier ein Auszug aus einem Cron init-Script:

```
#!/bin/sh
# Start/stop the cron daemon.
test -f /usr/sbin/cron || exit 0

case "$1" in
start) echo -n "Starting periodic command scheduler: cron"
      start-stop-daemon --start --quiet --exec /usr/sbin/cron

      echo "."
;;
```

Beim erfolgreichen Starten von Cron wird dann ein Punkt ausgegeben. Farblich könnte man das auch durch [Ok] gestalten, indem man den echo String verändert. Z.B. so:

```
#!/bin/sh
# Start/stop the cron daemon.
test -f /usr/sbin/cron || exit 0
case "$1" in
start) echo -n "Starting periodic command scheduler: cron"
      start-stop-daemon --start --quiet --exec /usr/sbin/cron
echo "\[ \033[1;34;40m[ \033[1;32;40mOk \033[1;34;40m]\033[0m\" "
      ;;
```

Bei einigen Systemen kann es sein, dass `\033` beim Starten des Systems nicht als Escape Sequenz interpretiert wird. An dieser Stelle muss man `Strg-V <ESC>` verwenden. Möchte man diese Einstellungen allerdings für alle init-Scripte vornehmen ist dies unter Umständen sehr zeitaufwändig.

## Feedback

Feedback, Kritik, Bugs etc. bitte per Mail an "nico at ngolde.de" schicken. Viel Spaß...

Webpages maintained by the LinuxFocus Editor team

© Nico Golde

"some rights reserved" see [linuxfocus.org/license/](http://linuxfocus.org/license/)

Translation information:

de ---> --- : Nico Golde <nico/at/ngolde.de>

