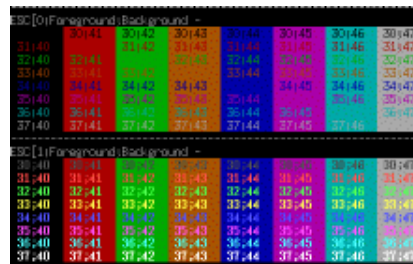


door Nico Golde
<nico/at/ngolde.de>

Over de auteur:

Op dit moment is Nico nog student. Sinds enkele jaren is hij veel bezig met Linux, en neemt hij deel aan enkele OpenSource-projecten.

Kleurrijke Shells -- Het Gebruik van ANSI-kleurcodes



Kort:

In een ANSI-bruikbare terminal (zoals xterm, rxvt, konsole ...) kan tekst getoond worden in kleuren anders dan zwart en wit. Dit artikel laat zien hoe tekst vetgedrukt en in kleur op uw scherm kan verschijnen.

Vertaald naar het Nederlands door:
Egon Willighagen
<egon.w(at)linuxfocus.org>

Algemeen

Elke Linux-gebruiker komt in contact met Bash. Dat lijkt saai maar er zijn veel mogelijkheden om de shell naar eigen smaak in te richten. Gekleurde toevoegingen aan de shell prompt maken deze uniek en ook beter leesbaar.

In dit verhaal verwijs ik naar de Bash-shell. De escape-karakterreeksen kunnen voor verschillende terminals afwijken; in dit artikel behandel ik alleen de ANSI-terminal.

Configuratie van de Shell

Het gebruik van kleuren is gedefinieerd in het persoonlijke configuratiebestand van bash `~/.bashrc` of in het algemene configuratiebestand `/etc/bashrc`. Het uiterlijk van de prompt wordt gedefinieerd met de

PS1-variabele in dit configuratiebestand.

Algemeen gesproken ziet deze regel er ongeveer uit als:

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

\s staat voor de naam van de shell en -\v voor de versie ervan. Op het einde staat een \$ als markering van het einde van de prompt. Omdat dit nogal saai is, kan ook de volgende gebruikt worden - die vaak de standaard is voor Linuxdistributies:

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

Deze prompt laat zien "gebruiker@computerNaam huidige_directory \$". Dit zal veel Linux-gebruikers bekend voorkomen.

Escape-karakterreeks

Om een eigen draai te geven door de prompt een kleurtje te geven, maken we gebruik van escape-karakterreeks. Een escape-karakterreeks is een instructie voor de shell om iets speciaals uit te voeren. Een escape-karakterreeks begint meestal met een ESC (vandaar de naam). In de shell is dat ^[. De reeks \033 heeft hetzelfde resultaat (ESC is ASCII 27 decimaal = 33 octaal).

Om een escape-karakterreeks direct bij een shell in te voeren moeten we het laten voorafgaan door een `ctrl-v`: *CTRL-v ESC*.

Het Gebruik van de Kleuren van de Shell

Nu ga ik uitleggen hoe kleuren van de shell gebruikt kunnen worden met een voorbeeld prompt:

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

Dit laat een volledig groene prompt zien. Zoals:

```
nico@ebrain:~$
```

\033 begint de escape-karakterreeks en met [beginnen we de kleur definitie. De 0 die dan volgt geeft de standaard fontbreedte. Andere breedtes behandel ik later. Door de string binnen een \[en \] te zetten voorkom ik dat de escape-karakterreeks in de prompt verschijnt en teveel plek inneemt.

Daarna kiezen we de kleur van de voorgrond (in dit geval 32, dat groen is). De achtergrondkleur 40 is zwart. Om te voorkomen dat de tekst na de prompt ook groen wordt, sluiten we af met \033[0m, dat de standaard kleur van de shell is. Voor zowel de voor- als achtergrondkleur zijn acht kleuren beschikbaar.

De kleuren: zwart (30), rood (31), groen (32), geel (33), blauw (34), magenta (35), cyaan (36) en wit (37).

De nummers voor de achtergrond kleur zijn in dezelfde volgorde maar beginnen bij 40, dus 40 voor zwart, 41 voor rood, enz.

Een voorbeeld:

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

Dit geeft wel een heel kleurrijke prompt:

```
nico@ebrain:~$
```

Om deze prompt-instellingen te testen gebruiken we `export PS1="string"`, en uiteindelijk kunnen we deze toevoegen aan `.bashrc`. Mijn huidige prompt ziet er zo uit:

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

```
[ nico@ebrain:~]
```

Teksteigenschappen

Zoals reeds eerder genoemd geeft de '0' achter de eerste escape-karakter de standaardteksteigenschap aan van de shell. De eigenschappen worden aangegeven met de volgende nummers: 0, 1, 22, 4, 24, 5, 25, 7, 27. Deze geven als resultaat: standaard, vet, niet vet, onderstreept, niet onderstreept, knipperend, niet knipperend, geïnverteerd, en niet geïnverteerd.

Met het onderstaande korte script kunnen we zien welke combinaties mogelijk zijn.

```
#!/bin/sh
#####
# Nico Golde <nico(at)ngolde.de> Homepage: http://www.ngolde.de
# Last change: 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
    done
    printf '\n'
done
printf '\033[0m'
done
```

Het script kan als `tar.gz` overgehaald worden van: `showansicol.tar.gz`

Nog een toepassing

De mogelijkheid om de kleur te veranderen van output in de shell is niet alleen handig voor het maken van een kleurrijker shellprompt, maar kan ook handig zijn voor een console-programma.

Voor elk gebruik van kleuren is het gebruik van een bibliotheek zoals *slang* of *ncurses* nodig, maar dit zou enorm de grootte van het uitvoerbare programma doen toenemen. *Ncurses* heeft het voordeel redelijk onafhankelijk te zijn van het type terminal.

Voorbeelden in C

Een 'Dag Wereld' in groene tekst:

```
#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;
}
```

Een andere bruikbare escape-karakterreeks is `printf("\033[2J")`, dat hetzelfde effect heeft als `system(clear)`, maar dan zonder de noodzaak de headerfile `unistd.h` toe te voegen.

Met `printf("\033[1K")` kunnen we verder een regel wissen.

Voorbeelden voor init-Scripts

Als we een meer kleurrijke en daarmee leesbaardere bevestiging willen van het al dan niet succesvol opstarten van de *init* scripts in `/etc/init.d`, en niet alleen een eenvoudige '.' kunnen we ook een escape-karakterreeks gebruiken.

Dit is een stukje uit de *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 "."
;;
```

Het bovenstaande script geeft alleen een punt als *cron* juist opgestart is. Een kleuruitbreiding en de tekst [Ok] zou het leesbaarder maken. Dit is voor elkaar te krijgen door de string te veranderen in bijvoorbeeld:

```
#!/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]"
;;
```

Het toepassen van deze truc voor alle scripts is zeer tijdrovend, temeer omdat we de escape-karakterreeks `\O33` gebruiken, omdat *Ctrl-v* niet als karakter geïnterpreteerd kan worden..

Feedback

Feedback, kritiek, fouten, enz. kunnen gestuurde worden per email naar "nico at ngolde.de". Veel plezier...

<p>Site onderhouden door het LinuxFocus editors team © Nico Golde "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Vertaling info: de --> -- : Nico Golde <nico/at/ngolde.de> de --> en: Jürgen Pohl <sept.sapins(at)verizon.net> en --> nl: Egon Willighagen <egon.w(at)linuxfocus.org></p>
---	---