

# Kleine Bash-Tricks

toheine

22.03.2025

# Vorbemerkungen

# Motivation

- Eigene Umgebung für sich selbst optimieren
- Möglichkeiten der Bash nutzen
- Nahe am Standard bleiben

# Verwendetes System

- Ausgangspunkt ist ein Debian 12
- Andere Distributionen verhalten sich weitgehend ähnlich
- Empfohlene Tools aus den regulären Debian-Paketquellen

# Ziel

- Prompt für die eigenen Bedürfnisse anpassen
- History optimieren
- Ein paar Kniffe bei der Eingabe
- Eine Auswahl an Zusatztools

# Relevante Konfigurationsdateien

Für diesen Vortrag relevant:

- `/etc/bash.bashrc` → Systemweite Initialisierung (per interactive Shell)
- `~/.bashrc` → individuelle Initialisierung (per interactive Shell)
- `~/.bash_history` → History-Datei
- `/etc/skel/.bashrc` → Vorlage der `~/.bashrc` für neue User

# Prompt Anpassungen

# Standard-Prompt

Aufbau üblicherweise:

```
username@<hostname> : <Pfad>#|$
```

Demo:

```
yoda@cltvm:~$ cd /tmp  
yoda@cltvm:/tmp$ █
```

Hilfreich um zu sehen:

- welcher User / welche Userin
- welches Gerät
- an welchem Ort im Dateibaum



# Weiterer Ausbau denkbar:

Anpassungen je nach System:

- Uhrzeit
- Farbliche Hervorhebung eigener Arbeitsgeräte
- Laborumgebungen von produktiven Umgebungen besser unterscheiden
- Git-Branch darstellen
- Zeilenumbruch um lange Pfade vom eigentlichen Kommando zu trennen
- u.v.m.

# Einfache (farblose) Umsetzung I

Anpassung der Variablen `PS1` ändert den Prompt sofort. Beispiel:

```
PS1=">>>"
```

Demo:

```
yoda@cltvm:~$ PS1=">>> "  
>>>
```

# Einfache (farblose) Umsetzung II

Verwendung von **Special Characters** bei der Zuweisung bieten Dynamik:

- `\u` → Username
- `\h` → Hostname
- `\w` → Aktuelles Arbeitsverzeichnis
- `\$` → Indikator für einen eingeschränkten User

Beispiel:

```
PS1="\u@\h:\w\$ "
```

Demo:

```
>>> PS1="\u@\h:\w\$ "  
yoda@cltvm:~$
```

# Einfache (farblose) Umsetzung III

Eigene Optimierung mit Uhrzeit und Umbruch. Zusätzliche **Special Characters**:

- `\n` → Zeilenumbruch
- `\A` → Uhrzeit im 24-Stundenformat

Beispiel:

```
PS1="\n└─ \A - \u@\h: [ \w ] \n└─\ $ "
```

Demo:

```
yoda@cltvm:~$ PS1="\n└─ \A - \u@\h: [ \w ] \n└─\ $ "  
└─ 13:36 - yoda@cltvm: [ ~ ]  
└─$
```

# Farbe rein bringen I

Bestandteile des Prompts farblich hervorheben. Auszug:

- `\[\e[33m\]` → braun
- `\[\e[37m\]` → weiss
- `\[\e[32m\]` → grün
- `\[\e[00m\]` → defaults

Beispiel:

```
PS1="\n┌─ \[\e[33m\]\A \[\e[m\]- \[\e[37m\]\u@\h\[\e[m\]: \[\e[33m\][ \w ]\[\e[m\]  
\n└─\ $ "
```

# Farbe rein bringen II

Bestandteile des Prompts farblich hervorheben. Auszug:

- `\[\e[33m\]` → braun
- `\[\e[37m\]` → weiss
- `\[\e[32m\]` → grün
- `\[\e[00m\]` → defaults

Demo:

```
└─ 13:36 - yoda@cltvm: [ ~ ]
└─ $ PS1="\n└─ \[\e[33m\]\A \[\e[m\]- \[\e[37m\]\u@\h\[\e[m\]: \[\e[33m\][ \w ]\[\e[m\] \n└─\ $ "
└─ 13:52 - yoda@cltvm: [ ~ ]
└─ $
```

# Prompt hinterlegen

- Sobald der gewünschte Prompt den eigenen Wünschen entspricht
- ... diese Variablenzuweisung in der `~/ .bashrc` unten anfügen
- Prompt wird nun bei Login wie gewünscht dargestellt

# Git-Branch im Prompt darstellen I

Hierzu in `~/.bashrc` folgende drei Änderungen durchführen:

## 1. Funktion hinterlegen:

```
function check_branch {  
    if git status > /dev/null 2>&1; then  
        branch="<<< $(printf '\uE0A0') $(git branch --show current) >>>"  
    else  
        branch=""  
    fi  
}
```

## 2. Variable PS1 anpassen:

```
PS1="\n└─ \[\e[33m\]\A \[\e[m\]- \[\e[37m\]\u@\h\[\e[m\]: \[\e[33m\][ \w ]\[\e[m\]  
\[\e[32m\]\$branch \[\e[m\]\n└─\ $"
```



# Git-Branch im Prompt darstellen II

## 3. Prompt-Command anpassen:

```
PROMPT_COMMAND="check_branch"
```

## Demo:

```
└─ 17:18 - yoda@cltvm: [ ~ ]  
└─$ ls  
testrepo  
  
└─ 17:18 - yoda@cltvm: [ ~ ]  
└─$ cd testrepo/  
  
└─ 17:18 - yoda@cltvm: [ ~/testrepo ] <<< ↵ main >>>  
└─$ █
```

# Über meine Inhalte hinaus

- Infos zu **Special Characters**: `man bash` im Abschnitt **PROMPTING**
- Infos zu Farbsequenzen: `man console_codes`
- zahlreiche Bash-Generatoren im Netz. Beispiel:  
<https://bash-prompt-generator.org/>
- Fertige Lösungen gibt es ebenfalls. Beispiel:  
<https://ohmyposh.dev/>

# History

# History aufrufen

```
14:17 - yoda@cltvm: [ ~ ]
└─$ history
 1 PS1=">>> "
 2 PS1="\u@\h:\w\$ "
 3 PS1="\n└─ \A - \u@\h: [ \w ] \n└─\$ "
 4 PS1="\n└─ \[\e[33m\]\A \[\e[m\]- \[\e[37m\]\u@\h\[\e[m\]: \[\e[33m\][ \w ]\[\e[m\] \n└─\$ "
 5 man console_codes
 6 man bash
 7 history
 8 vim .bash_history
 9 exit
10 history
```

- Ausgabe der letzten Befehle mit einer ID.
- Liegt im RAM
- Beim Logout werden Befehle in die Datei `~/.bash_history` geschrieben

# History anpassen (Beispiel) I

In der `.bashrc` folgende Variablen nach eigenen Vorlieben anpassen:

```
HISTCONTROL=ignoreboth          # entspricht ignoredups + ignorespace
HISTSIZE=30000                  # Default: 1000
HISTFILESIZE=90000              # Default: 2000
HISTTIMEFORMAT="%d.%m.%y  "
HISTIGNORE="?:??:??:bash:clear:exit:man*:*--help"
PROMPT_COMMAND="history -a;check_branch"
```

# History anpassen (Beispiel) II

Ergebnis - Aufbau der ~/.bash\_history:

```
#1742304872  
echo "Hallo"  
#1742304880  
whoami
```

Ausgabe des Kommandos history:

```
19 (18.03.25) echo "Hallo"  
20 (18.03.25) whoami  
21 (18.03.25) ls -l  
22 (18.03.25) history
```

Ein Filtern nach Datum ist nun einfach möglich z. B. via

```
history | grep 18.03.25
```

# Kleine Konsolenkniffe

# Autovervollständigung

- Die Wundervolle [TAB]-Taste:
  - Autovervollständigung
  - Mehrdeutige Möglichkeiten ausgeben für Dateien ...
  - ... und Kommandos
- Letztes Argument hinter neuen Befehl: [STRG] + [ALT] + .



# Schneller Zugriff auf die History

- Bestimmter Befehl aus der History aufrufen → `!<id>`
- Den letzten Befehl mit `!-1`, vorletzten Befehl mit `!-2`, ... aufrufen
- Den letzten Befehl mit `!!` aufrufen
- Den letzten Befehl mit voran gestellten `sudo` aufrufen → `sudo !!`
- Rekursive Suche → `[STRG] + [R]` → `<zeichenkette>`

# Wie war das nochmal? apropos

- Ich weiß den Befehl nicht: `apropos <Suchbegriff>`
- Suche mit `apropos -a` (and) eingrenzen.

Beispiel: Ich suche einen Befehl um sicher auf ein Remote-Ziel zu kopieren:

```
apropos secure          # liefert 84 Treffer
apropos copy            # liefert 19 Treffer
apropos -a secure copy  # liefert 2 Treffer
```

# Was ich sonst noch so brauche

- Mit `cd -` ins vorgehende Arbeitsverzeichnis wechseln (`oldpwd`)
- Die letzten 10 Zeilen einer Log-Datei anzeigen → `tail <dateiname>`
- Die Ausgabe live verfolgen → `tail -f <dateiname>`
- Geht auch mit Systemd-Journal → `jornalctl -fu <unitname>`
- Programm periodisch ausführen → `watch <kommando>`

# **Eine Handvoll Tools**

# ncdu

- Zweck: Festplatten-Belegung ausgeben
- Alternative mit `NCurses` basierendem `du` → `ncdu`
- Grafische Hervorhebung und absteigende Sortierung für besseren Überblick

```
apt install ncdu
```

```
ncdu 1.18 ~ Use the arrow keys to navigate, press ? for help
--- / -----
 1,4 GiB [#####] /usr
. 462,8 MiB [####] /var
 86,1 MiB [ ] /boot
. 3,9 MiB [ ] /etc
. 556,0 KiB [ ] /run
 344,0 KiB [ ] /home
. 28,0 KiB [ ] /tmp
```

# htop

- Zweck: Systemressourcen mit überwachen
- Bietet etwas mehr Möglichkeiten als das reine top
- Installation:

```
apt install htop
```

```
 0[|||
 1[
Mem[|||||
Swp[

2.0%] Tasks: 19, 4 thr, 63 kthr; 1 running
0.0%] Load average: 0.05 0.02 0.00
102M/961M] Uptime: 06:20:52
0K/975M]

Main I/O
PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%  TIME+  Command
1671 yoda        20   0 18088  6800  4932 S   0.7  0.7  0:00.28 sshd: yoda@pts/0
1878 yoda        20   0  7796  4004  3300 R   0.7  0.4  0:00.04 htop
   1 root         20   0    99M 12124  9216 S   0.0  1.2  0:01.31 /sbin/init
 214 root         20   0 49344 20556 15380 S   0.0  2.1  0:00.66 /lib/systemd/systemd-journald
 240 root         20   0 26828  6172  4616 S   0.0  0.6  0:00.14 /lib/systemd/systemd-udev
 266 systemd-ti  20   0 90092  6652  5768 S   0.0  0.7  0:00.24 /lib/systemd/systemd-timesyncd
 358 root         20   0  5872  3636  2796 S   0.0  0.4  0:00.00 dhclient -4 -v -i -pf /run/dhclient.enp0s3.p
 359 root         20   0  5872  3488  2644 S   0.0  0.4  0:00.16 dhclient -4 -v -i -pf /run/dhclient.enp0s8.p
520 systemd-ti  20   0 90092  6652  5768 S   0.0  0.7  0:00.00 /lib/systemd/systemd-journald
```

# vim (vi Improved)

- Zweck: Editor meiner Wahl (Newbies setzen evtl. eher auf nano)
- vi bereits verfügbar. vim muss evtl. nach installiert werden.

```
apt install vim
```

Empfehlenswerte Anpassungen in der `~/.vimrc`

```
1 syntax on
2 set number
3 set expandtab
4 set tabstop=4
5 |
```

# shellcheck

- **Zweck:** Ein Analyse-Tool um Fehler in eigenen Bash-Skripten zu finden
- Gibt Hinweise und Hintergrundinfos zu vereinzelt Fehlern

```
apt install shellcheck
```

Beispiel:

```
yoda@cltvm:~$ shellcheck hallo_welt.sh

In hallo_welt.sh line 4:
if [ $(whoami) == "yoda" ]; then
    ^-----^ SC2046 (warning): Quote this to prevent word splitting.

For more information:
https://www.shellcheck.net/wiki/SC2046 -- Quote this to prevent word splitt...
```



# Terminal Multiplexer

```
apt install tmux
```

Empfehlenswerte Anpassung in der `~/.tmux.conf`:

```
set -g mouse on
```

Shortcuts die man kennen sollte:

- `STRG + B + %` → Fenster vertikal teilen
- `STRG + B + "` → Fenster horizontal teilen
- `STRG + B + c` → Neues Fenster
- `STRG + B + d` → Detach (Session läuft im Hintergrund weiter)

# tmate

- Terminal Multiplexer mit **instant terminal sharing**
- Zweck: Schnell mal Hilfe auf die Konsole holen

```
apt install tmate
```

```
Tip: if you wish to use tmate only for remote access, run: tmate -F
To see the following messages again, run in a tmate session: tmate show-messages
Press <q> or <ctrl-c> to continue
-----
Connecting to ssh.tmate.io...
Note: clear your terminal before sharing readonly access
web session read only: https://tmate.io/t/ro-HMzgUZ8ju9tCwkSSz[REDACTED]
ssh session read only: ssh ro-HMzgUZ8ju9tCwkSSz[REDACTED].tmate.io
web session: https://tmate.io/t/KckjRaVQyZctBPY[REDACTED]
ssh session: ssh KckjRaVQyZctBPY[REDACTED]@lon1.tmate.io
```

# Fazit

(M)eine total subjektive Einschätzung der Inhalte hier

- Investition in die Anpassung der eigenen Umgebung lohnt sich
  - Schneller orientieren
  - effizienter Arbeiten
  - Missgeschicke vermeiden
- IMHO mehr als nur eine Spielerei
- Es gibt so viel mehr zu entdecken