

Zusammenfassung Tag 15

Einführung, etc profile, etc bashrc

- **2 Konfigurationsdateien für Shell**
 - login Shell wird gestartet zb über Ssh (/etc/profile/ und /etc/bash.bashrc)
 - nicht-login Shell wird lokal genutzt (/etc/bash.bashrc)
- **Code der immer ausgeführt werden soll wird dann in bash.bashrc geschrieben**
- Hinweis CentOS
 - Konfigurationsdatei heißt hier /etc/bashrc

Die Dateien bash_login und bash_logout

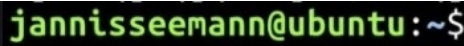
- **beim start einer login Shell wird etc/profile geladen**
 - überprüft ob ~/.bash_profile existiert
 - wenn Datei vorhanden ist wird diese geladen
 - Falls nicht vorhanden wird
 - überprüft ob ~/.bash_login existiert
 - wenn Datei vorhanden ist wird diese geladen
 - Falls nicht vorhanden wird
 - überprüft ob ~/.profile existiert
 - wenn Datei vorhanden ist wird diese geladen
- **beim logout**
 - ~/.bash_logout wird geladen

Variablen definieren (erstellen vs export)

- Variable in der Shell
- **GREETING="Hallo Welt"**
 - Erstellt die Variable GREETING als String mit dem Inhalt Hallo Welt
 - ist nur lokal
- **echo \$GREETING**
 - gibt die Variable GREETING aus.
 - Beim Aufruf einer Variable \$ benutzen
- **export GREETING="Hallo Welt"**
 - Erstellt die Variable GREETING als String mit dem Inhalt Hallo Welt
 - erstellt eine UmgebungsVariable
- **unset GREETING**
 - löscht die Variable Greeting
- **lokale Variablen**
 - können nur von der aktuellen Shell genutzt und aufgerufen werden
- **UmgebungsVariablen**

- können Shell übergreifend genutzt und aufgerufen werden
- können auch von Scripten genutzt und aufgerufen werden

Die Variable PS1

- **~/.bashrc**
 - Hier befindet sich die Variable PS1
 - Unter CentOS: /etc/bashrc
- **Variable PS1**
 - sorgt dafür das in der Shell in jeder Zeile der [User@system](#) und der aktuelle Dateipfad stehen 
- **man bash**
 - ruft das Manual von bash auf
 - unter Prompting stehen die Erklärung für die Parameter die zb für PS1 genutzt werden können
 - \u gibt den User aus
 - \h gibt den Hostname aus
 - \W gibt den aktuellen Pfad aus
 - \\$ gibt eine # aus wenn man als root eingeloggt ist oder ein \$ wenn man nicht als root eingeloggt ist

Farben in Shell, PS1 mit Farben

- **paket colortest**
 - sudo apt-get install colortest
 - Paket gibt es nicht für CentOS, Farbcodes sind aber die selben
- **colortest-16b**
 - gibt die gesamten Farbcodes aus die zur Verfügung stehen
- **echo -e "\e[32mHalloWelt"**
 - gibt Hallo Welt in grün aus
 - \e[32m steht für grün
- **echo -e "\033[41m\e[30mHallo Welt"**
 - gibt Hallo Welt in Schwarzer Schrift auf einem roten Hintergrund aus
 - \033[41m steht für den roten Hintergrund
- **\e[0m**
 - setzt die Farben auf den Standard zurück
- **Durch einsetzen der Farbcodes in die Variable P1 können die Farben angepasst werden**
 - PS1="\[\e[1;32m\]\u@\h\[\e[0m\]:\[\e[1;34m\]\W\[\e[0m\]\\$"

- damit dies auch nach einem Neustart weiterhin so bleibt kann der Code in die `/etc/bash.bashrc` eingefügt werden
Und auch in die Jeweilige Benutzer config: `/home/user/.bashrc`

Die Variablen PS2, PS3, PS4

- **Variable PS2**
 - Gibt an was am Anfang einer fortlaufenden Zeile steht
 - Standardwert: `>`
- **Variable PS3**
 - Gibt den Wert für den select befehl an. (shell-script)
- **Variable PS4**
 - wird mit `set-x` genutzt (aktiviert den Entwicklungsmodus)
 - kann mit `set+x` entfernt werden
 - gibt den Wert an der am Anfang einer Zeile steht wenn die Debug Informationen angezeigt werden
 - Standardwert: `+`

Aliase hinzufügen und entfernen

- Aliase werden genutzt um lange Befehle zu verkürzen (shortcut)
 - z.B. statt `date +%d.%M.%Y %H:%I` nur noch `d`
 - `alias d="date +%d.%M.%Y %H:%I"`
- **unalias d**
 - löst den alias `d = date` wieder auf
- **alias**
 - Gibt alle alias in der Shell aus
- Es können mehrere Befehle in einem Alias stehen
- Wenn das alias auch nach einem Neustart vorhanden bleiben soll, muss der Befehl in die `~/.bashrc` geschrieben werden. (am besten ans Ende der Datei)
 - `alias d="date +%d.%M.%Y %H:%I"`
- **Achtung**
 - Alias nicht gleich wie ein Programm/Befehl nennen, da diese sonst nicht mehr genutzt werden können

Die Inputrc-Datei

- **Inputrc-Datei**
 - Konfiguration für die eingaben
 - /etc/inputrc
 - Definition von Tastenkombinationen
- **Eigenen code in /etc/.inputrc (Ubuntu)**
 - \$include /etc/inputrc an den Anfang der Datei schreiben
- **Control-S: "ls -al"**
 - nach hinzufügen des Codes muss die Datei neu eingebunden werden
 - bind -f ~/.inputrc
 - durch drücken von strg+s wird dann ls -al ausgeführt

Bonus – Oh-My-Zsh

- **Ein Tool für eine schönere zsh Shell**
 - Es können auch Plug-Ins installiert werden
 - Mehrere Konfigurationsmöglichkeiten
- **sudo apt-get install zsh**
 - installiert die shell zsh
 - wird für das tool oh-my-zsh benötigt
- **sh c "\$(curl -fsSL <https://raw.githubusercontent.com/robbyrussell/ohmyzsh/master/tools/install.sh>)"**
 - Befehl zum installieren von Oh-My-Zsh
 - programm curl und git wird benötigt
 - sudo apt install curl
 - sudo apt install git

Nützliche Befehle:

<i>clear</i>	Bereinigt die Konsole
<i>strg+c</i>	Beendet ein Programm / unterbricht einen Befehl
<i>cat</i>	Erzeugt eine Ausgabe z.B. von einer Datei
<i>nano</i>	Einfacher Editor zum bearbeiten von Dateien
<i>commandname --help</i>	Öffnet meistens die Hilfe eines Programm
<i>man commandname</i>	Öffnet das Manual eines Programm falls vorhanden