Linux Basics

Gregor G

April 8th, 2020

Scope der Präsentation

Wir möchten euch das Grundlegende Arbeiten mit einem Linux System näherbringen.

Die Präsentation ist als kleiner Überblick gedacht, der es euch erlaubt erfolgreich ins Selbststudium zu starten.

Unix ist von 1970, Linux von 1991, es gibt genügend Literatur :)

Linux Allgemein

- Kernel (“Redet mit der Hardware”) -> Linux
- Userspace (“Eure Anwendungen”)

Freie Software mit viraler Lizenz -> GNU GPL

Warum Linux : Auf (Web)servern & Cloud hat Linux “gewonnen”. Mindshare für Entwickler Tools ist dort.

Distributionen

Durch die Freie Lizenz gibt es verschiedene Distributionen, eine Kombination von Kernel, Software, Dokumentation, Support und Community

- Debian
- Ubuntu
- RedHat
- SUSE
- Arch Linux

Wir nehmen für die Präsentation ein Debian System an. Insbesondere beim Installieren von weiterer Software bestehen Unterschiede.

Herausforderungen für Neulinge

- Normalerweise keine graphische Benutzeroberfläche
- Keine zentrale Einstellungsanwendung
- Andere Dateipfade `/home/gregor/test.txt`

Die Komandozeile Shell

Der Prompt zeigt links den Benutzernamen und Systemnamen ein, rechts davon blinkt der Cursor für eure Eingaben:

`root@demo:~#`

- `ls` : List Files
- `ls -a` : List all Files
- `ls --help` : Show Help for Command

Remote Zugang

Die Shell ist nur Text, deswegen kann sie effizient auch remote übertragen werden.

Der Standard hierfür ist **SSH**, welches verschlüsselte Verbindungen aufbaut. Nach dem Ausführen von *ssh benutzer@servername* wird die aktuelle Shell durch die Remote Shell ersetzt.

Das Dateisystem

Relativ standardisierter Aufbau, leichte Unterschiede je nach Distribution

- /home/$USERNAME : Das Heimatverzeichniss
- /etc/ : Ordner für Optionen
- /opt/ : Programmpakete
- /usr/bin : Ausführbare Dateien

/ zeigt auf das Rootverzeichniss. Pfade die damit mit starten sind absolut, alle anderen relativ zum aktuellen Ordner.

Autocomplete, History und Text

Die zwei wichtigsten Tools im Kampf mit der Shell.

- *Tab* vervollständigt Dateien (und teilweise Optionen)
- *Pfeil-Oben* geht durch die Historie der Komandos
- *POS1* und *ENDE* Springen ans Ende oder Anfang der Zeile
- *Strg+Pfeil-Rechts/Links* Bewegt den Cursor pro Wort.

Sammlung von Standard Tools

Die Unix Philosophie : Kleine, hochoptimierte Einzeltools

- `cat` : Concat
- `mkdir` : Make Dir
- `mv` : Move
- `grep` : Search Text in File
- `sed` : Stream Editor (*Suchen und Ersetzen*)
- `rm` : Delete File
- `ln` : Link (*Verknüpfung*)
- `head`, `tail`, `less` : Ersten/Letzen Zeilen, Interaktiv

IO und IO Redirection

Programme haben einen Input und Output

Der Output ist normalerweise an die Shell Ausgabe gebunden.

- `cat test.txt` gibt Inhalt von test.txt auf dem Bildschirm aus.

Der Input kann per < gesetzt weren:

- `cat < test.txt` führt zum gleichen Ergebnis

Output kann aber auch umgeleitet werden

- In eine Datei: `cat test.txt > output.txt`
- Zum Input eines weiteren Programms : `cat test.txt | cat`

Kurze Beispiele:

- ls | grep pdf
- grep echo < test.txt
- cat test.txt | grep echo

Quoting

Zum Gruppieren werden Anführungszeichen verwendet

- `grep "ec ho" < test.txt`

Beim Schachteln von Anführungszeichen können einfache Anführungszeichen verwendet werden

- `grep 'ech"o"' < text.txt`

Variablen

Über die $NAME Syntax können Variablen Vergeben werden.

```
export TEST=test1234
echo $TEST
echo "$TEST" # gibt 'test1234' aus
echo '$TEST' # gibt '$TEST' aus
```

Einige Variablen sind bereits vordefiniert, (z.B $HOME für das Heimatverzeichniss)

Benutzer und Rechte

Jede Datei hat einen User, Gruppe und Zugriffsrechte. Jeder Benutzer hat einen User und mehrere Gruppen

```
total 136K
drwxr-xr-x 1 gregor gregor   58 Apr  7 22:25 .
drwxr-xr-x 1 gregor gregor 2.0K Apr  7 22:23 ..
-rw-r--r-- 1 gregor gregor 5.0K Apr  7 23:01 myslides.html
-rw-r--r-- 1 gregor gregor 122K Apr  7 23:27 out.pdf
-rw-r--r-- 1 gregor gregor 3.2K Apr  7 23:28 slides.md
```

Berechtigungen können über *chown* und *chmod* gesetzt werden.

Sudo & Root

Ein Normaler Benutzer hat normalerweise reduzierte Rechte. Selbst ein Admin sollte nicht standardmäßig als Root eingeloggt sein, sondern gezielt erweitere Rechte anfordern

Das Komando “sudo” kann temporär Rechte erhöhen.

```
sudo cat test.txt
```

Root Nutzer haben vollen Zugriff, bzw können sich diesen beschaffen

Texteditoren : VIM & Nano

Textbasierte Editoren erlauben das schnelle Editieren von Dateien.

- **Nano** ist ein minimaler Editor, auf Debian vorinstalliert.
- **Vim** und **Emacs** sind erweitere Editoren mit Shortcuts.

Eine der Populärsten Stackoverflow Fragen ist “How to Exit VIM” *(Esc + :q + Enter)*

Außerhalb vom Scope dieses Tutorials

Hinweis: Direkt auf dem Server sollte eigentlich nicht im Produktiv System gearbeitet werden, die Konfiguration sollte von Git oder Ansible oder ähnlichem verwaltet werden.

Software Verwaltung

Freie Software wird zentral verwaltet und bereitgestellt. Sozusagen ein App-Store der von der Distribution bereitgestellt wird. Dank freier Lizenz können die Pakete von dritten verwaltet und angepasst werden.

Auf Debian ist *Apt* das Tool zur Software Verwaltung.

- *apt update* : Appstore Aktualisieren
- *apt upgrade* : Update Installieren
- *apt search* : Suche nach Paketen
- *apt install* : Installation

Software in den Standard Repositories ist von den Distributionen geprüft.

Dienste Verwalten

Es hat sich `systemd` für die Systemverwaltung durchgesetzt. Es ist eine modulares System welches Aufgaben übernimmt die vorher auf Einzelsysteme verteilt war.

Wichtige Standardkomponenten: Dienste am Laufen halten (z.B durch Neustarts bei Abstürzen), zu bestimmten Zeiten Starten und Logging

- *systemctl status* : Dienst Status
- *systemctl start/stop/restart* : Dienste Starten und Anhalten
- *systemctl enable/disable* : Dienste Autostart

Dienste Loggen

Logs wurden traditionell in eine Datei unter */var/logs/* geschrieben. `systemd` führt zentrales Logmanagement über `journald` ein.

- *journalctl -xe*
- *journalctl -u $dienst*

In den Einstellungen von `journald` kann festgelegt werden was mit Logs passiert, z.B. wie lange diese behalten werden.

SSH Erweitere Funktionen

Um die Sicherheit von SSH zu erhöhen, sollte die Passwortauthentifizierung abgeschaltet werden. Zugriff ist dann nur durch autorisierte Public Keys möglich.

Key kann im Client hinterlegt werden, keine weitere Authentifizierung notwendig (unsicherer). Alternative: Mit Passwort schützen oder auf Hardware Key kopieren.

SSH kann dann ohne Bedenken durch die Firewall offen gestellt werden. SSH ist sicher!

SSH Port Forwarding

Der sichere Tunnel kann auch für andere Kommunikation genutzt werden. Ideal um z.B. auf eine Datenbank für Notfalloperationen zuzugreifen.

```
ssh user@server -L LOCAL_PORT:localhost:REMOTE_PORT
```

Baut eine Verbindung zum Server auf und erstellt einen Tunnel zum REMOTE_PORT auf dem Server. Dieser ist dann auf dem client unter LOCAL_PORT zu erreichen.

Datei Transfers über SSH

- Simpel über **scp** : scp test.txt server:/tmp/test.txt
- Erweitert über **rsync**: Viele Optionen, vorallem aber Delta Transfers.

Firewall

- Low level tool ist **iptables**, mit dem man sehr detailiert Netzwerkanfragen filtern oder routen kann. Achtung: Trennung in *iptables* und *ip6tables* für ipv6
- Besseres Frontend: **ufw**. Vereinfacht Befehle und Management. *ufw allow 5050* erlaubt Zugriffe auf Port 5050.