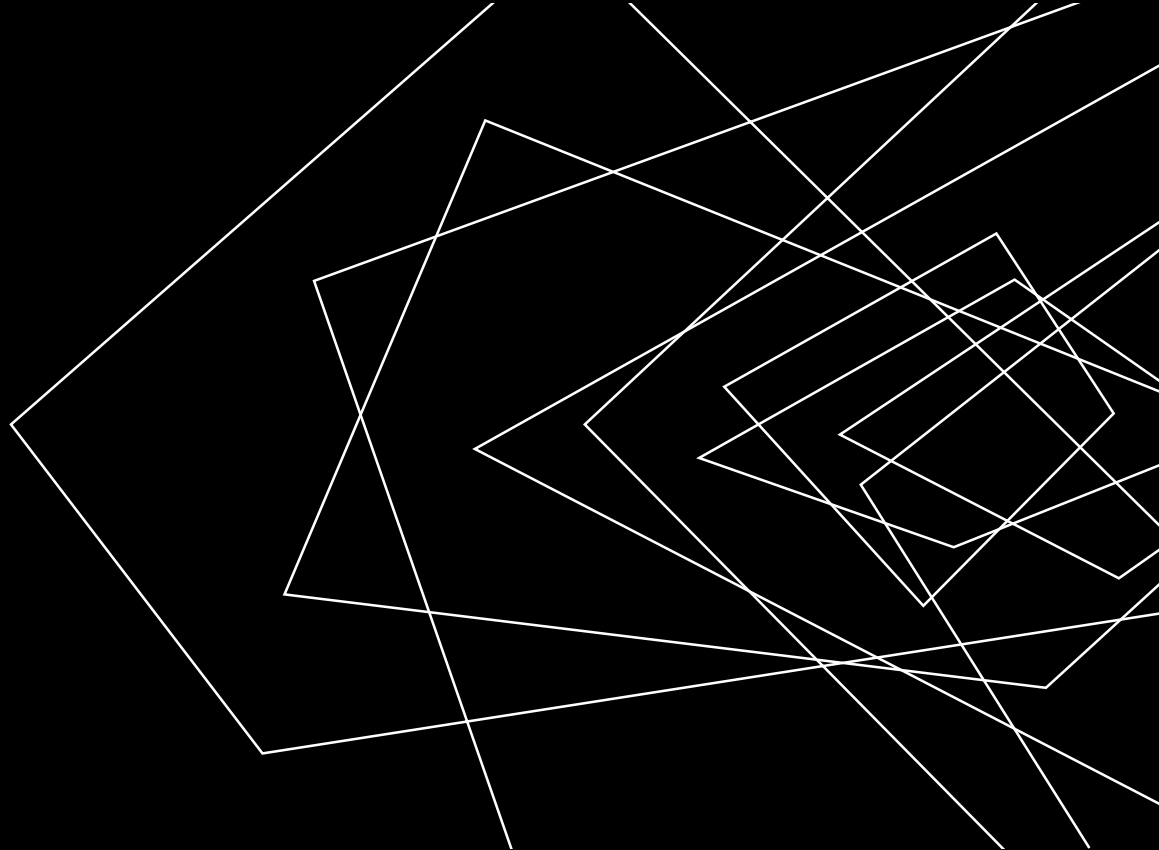
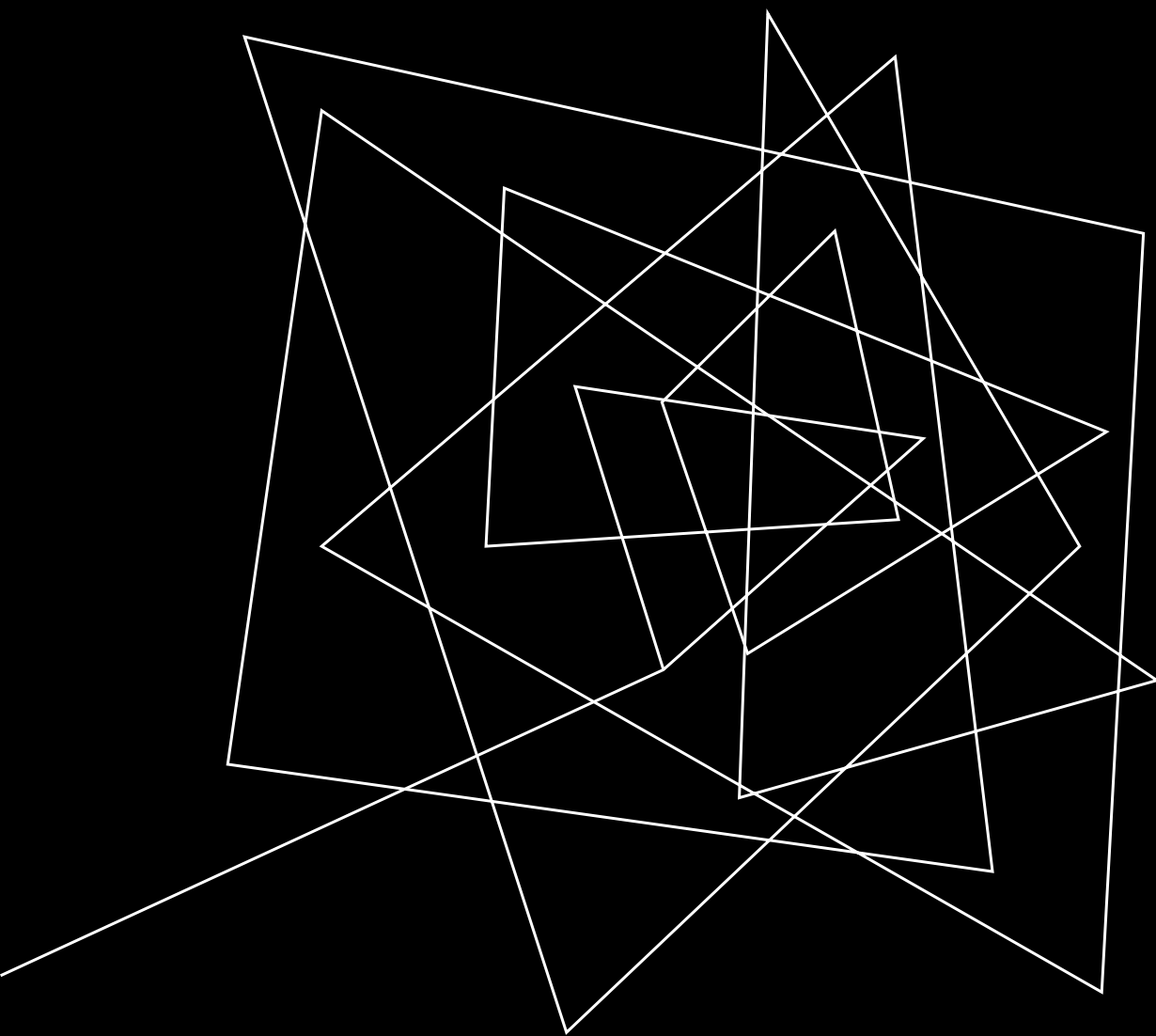


LINUX - TAG 3

AGENDA

1. Benutzer- und Gruppenmanagement
2. Einführung in Netzwerke unter Linux
3. Netzwerktools und Netzwerkbefehle
4. Einführung in den Paketmanager





1. BENUTZER- UND GRUPPENMANAGEMENT

Anlegen und Verwalten von Benutzern und Gruppen,
Befehle wie `useradd`, `usermod`, `groupadd`

1. BENUTZER- UND GRUPPENMANAGEMENT

- 1.1. Übersicht
- 1.2. Benutzerkonten
- 1.3. Der Befehl useradd
- 1.4. Der Befehl usermod
- 1.5. Benutzerverwaltung - Passwort ändern
- 1.6. Gruppenmanagement
- 1.7. Der Befehl groupadd
- 1.8. Der Befehl gpasswd
- 1.9. Der Befehl groupdel
- 1.10. Praxisbeispiele
- 1.11. Zusammenfassung

1.1. ÜBERSICHT

- Einführung in das Benutzer- und Gruppenmanagement
- Wichtige Befehle:
 - useradd
 - usermod
 - groupadd
- Benutzerrechte und Gruppenzugehörigkeit

1.2. BENUTZERKONTEN

Was ist ein Benutzerkonto?

- Ein Benutzerkonto ermöglicht den Zugriff auf das System.
- Jeder Benutzer hat:
 - Eindeutige UID (User ID)
 - Heimatverzeichnis
 - Shell

Ziel des Benutzerkontos:

- Verwalten von Zugriffsrechten und Berechtigungen.

1.3. DER BEFEHL USERADD

Verwendung:

- Erstellt ein neues Benutzerkonto.

Syntax:

- **useradd [Optionen] Benutzername**

Wichtige Optionen:

- **-m**: Erstellt ein Heimatverzeichnis
- **-s**: Legt die Shell fest

Beispiel:

- **useradd -m -s /bin/bash max** (Erstellt den Benutzer max mit Bash als Shell)

1.4. DER BEFEHL USERMOD

Verwendung:

- **Ändert bestehende Benutzereinstellungen.**

Syntax:

- **usermod [Optionen] Benutzername**

Wichtige Optionen:

- **-l: Ändert den Benutzernamen**
- **-g: Ändert die primäre Gruppe**
- **-aG: Fügt den Benutzer zu einer sekundären Gruppe hinzu**

Beispiel:

- **usermod -aG sudo max (Fügt max zur Gruppe sudo hinzu)**

1.5. BENUTZERVERWALTUNG - PASSWORT ÄNDERN

Verwendung des Befehls passwd:

- **Setzt oder ändert das Passwort eines Benutzers.**

Syntax:

- **passwd [Benutzername]**

Beispiel:

- **passwd max (Ändert das Passwort für den Benutzer max)**

1.6. GRUPPENMANAGEMENT

Was ist eine Gruppe?

- **Gruppen fassen Benutzer zusammen, um Berechtigungen einfacher zu verwalten.**
- **Jeder Benutzer kann mehreren Gruppen angehören.**

Ziel:

- **Verwaltung gemeinsamer Ressourcen durch Zuweisung von Gruppenzugriffsrechten.**

1.7. DER BEFEHL GROUPADD

Verwendung:

- **Erstellt eine neue Gruppe.**

Syntax:

- **groupadd [Gruppenname]**

Beispiel:

- **groupadd entwickler (Erstellt die Gruppe entwickler)**

1.8. DER BEFEHL GPASSWD

Verwendung:

- **Verwaltet Gruppenmitglieder.**

Syntax:

- **gpasswd -a [Benutzername] [Gruppenname] (Fügt einen Benutzer zu einer Gruppe hinzu)**
- **gpasswd -d [Benutzername] [Gruppenname] (Entfernt einen Benutzer aus der Gruppe)**

Beispiel:

- **gpasswd -a max entwickler (Fügt den Benutzer max zur Gruppe entwickler hinzu)**

1.9. DER BEFEHL GROUPDEL

Verwendung:

- **Löscht eine Gruppe.**

Syntax:

- **groupdel [Gruppenname]**

Beispiel:

- **groupdel entwickler (Löscht die Gruppe entwickler)**

1.10. PRAXISBEISPIELE

1. Benutzer anlegen:

- **useradd -m -s /bin/bash anna**

2. Benutzer zur Gruppe hinzufügen:

- **usermod -aG sudo anna**

3. Neue Gruppe erstellen:

- **groupadd marketing**

4. Benutzer zu einer Gruppe hinzufügen:

- **gpasswd -a anna marketing**

1.11. ZUSAMMENFASSUNG

- **Benutzerverwaltung:** Mit Befehlen wie `useradd` und `usermod` Benutzer erstellen und verwalten
- **Gruppenmanagement:** Mit Befehlen wie `groupadd` und `gpasswd` Gruppen erstellen und Benutzer zuweisen
- **Sicherheitsrelevanz:** Genaue Verwaltung der Benutzer- und Gruppenrechte erhöht die Systemsicherheit



2. EINFÜHRUNG IN NETZWERKE UNTER LINUX

Grundlagen der Netzwerkkonfiguration, IP-Adressen,
Netzwerkkarten

2. EINFÜHRUNG IN NETZWERKE UNTER LINUX

- 2.1. Übersicht
- 2.2. Netzwerke unter Linux
- 2.3. Netzwerkkarten (NIC)
- 2.4. IP-Adressen
- 2.5. Netzwerkkonfiguration
- 2.6. Der Befehl ifconfig
- 2.7. Der Befehl ip
- 2.8. Der Befehl ping
- 2.9. Praxisbeispiele
- 2.10. Zusammenfassung

2.1. ÜBERSICHT

- Netzwerke und ihre Rolle in Linux
- Grundlegende Netzwerkkonzepte:
 - IP-Adressen
 - Netzwerkkarten
- Konfiguration von Netzwerken unter Linux

2.2. NETZWERKE UNTER LINUX

Was ist ein Netzwerk?

- **Ein Netzwerk verbindet Computer, um Daten auszutauschen.**
- **Netzwerkschnittstelle (NIC): Ermöglicht die physische Verbindung zu einem Netzwerk.**

Ziel eines Netzwerks:

- **Datenübertragung zwischen Computern, Servern und anderen Geräten.**

2.3. NETZWERKKARTEN (NIC)

Netzwerkkarten in Linux:

- **Jede Netzwerkkarte wird als Schnittstelle behandelt.**
- **Typische Namen von Netzwerkschnittstellen:**
 - **eth0: Erste Ethernet-Schnittstelle**
 - **wlan0: Erste WLAN-Schnittstelle**

Verwaltung von Netzwerkkarten:

- **Überprüfen von Schnittstellen mit ifconfig oder ip addr.**

2.4. IP-ADRESSEN

Was ist eine IP-Adresse?

- Eine IP-Adresse identifiziert Geräte im Netzwerk eindeutig.
- Zwei Versionen:
 - IPv4: 32-Bit-Adresse (z.B. 192.168.1.1)
 - IPv6: 128-Bit-Adresse (z.B. 2001:db8::1)

IP-Adressen und Subnetzmasken:

- Subnetzmasken trennen Netzwerkteile (z.B. /24 für IPv4).

2.5. NETZWERKKONFIGURATION

Statische vs. Dynamische IP-Adressen:

- **Statisch: Fest zugewiesene IP-Adresse.**
- **Dynamisch: Vom DHCP-Server automatisch zugewiesene IP-Adresse.**

Netzwerk-Dateien:

- **/etc/network/interfaces: Manuelle Netzwerkkonfiguration.**
- **/etc/resolv.conf: DNS-Server für Namensauflösung.**

2.6. DER BEFEHL IFCONFIG

Verwendung:

- **Veraltet, aber noch gebräuchlich zur Anzeige und Konfiguration von Netzwerkinterfaces.**

Syntax:

- **ifconfig [Schnittstelle]**

Beispiel:

- **ifconfig eth0 (Zeigt Informationen über die Schnittstelle eth0)**

2.7. DER BEFEHL IP

Verwendung:

- **Moderner Ersatz für ifconfig.**

Syntax:

- **ip addr (Zeigt alle Netzwerkinterfaces und IP-Adressen)**

Beispiel:

- **ip addr show eth0 (Zeigt die IP-Adresse und Details für eth0)**

2.8. DER BEFEHL PING

Verwendung:

- **Testet die Erreichbarkeit eines Netzwerks.**

Syntax:

- **ping [IP-Adresse oder Hostname]**

Beispiel:

- **ping 192.168.1.1 (Sendet Pakete zur IP-Adresse 192.168.1.1)**

2.9. PRAXISBEISPIELE

1. Netzwerkinterfaces anzeigen:

- **ip addr**

2. Statische IP-Adresse konfigurieren (Debian-basiert):

- **Editiere /etc/network/interfaces**
- **Beispiel:**

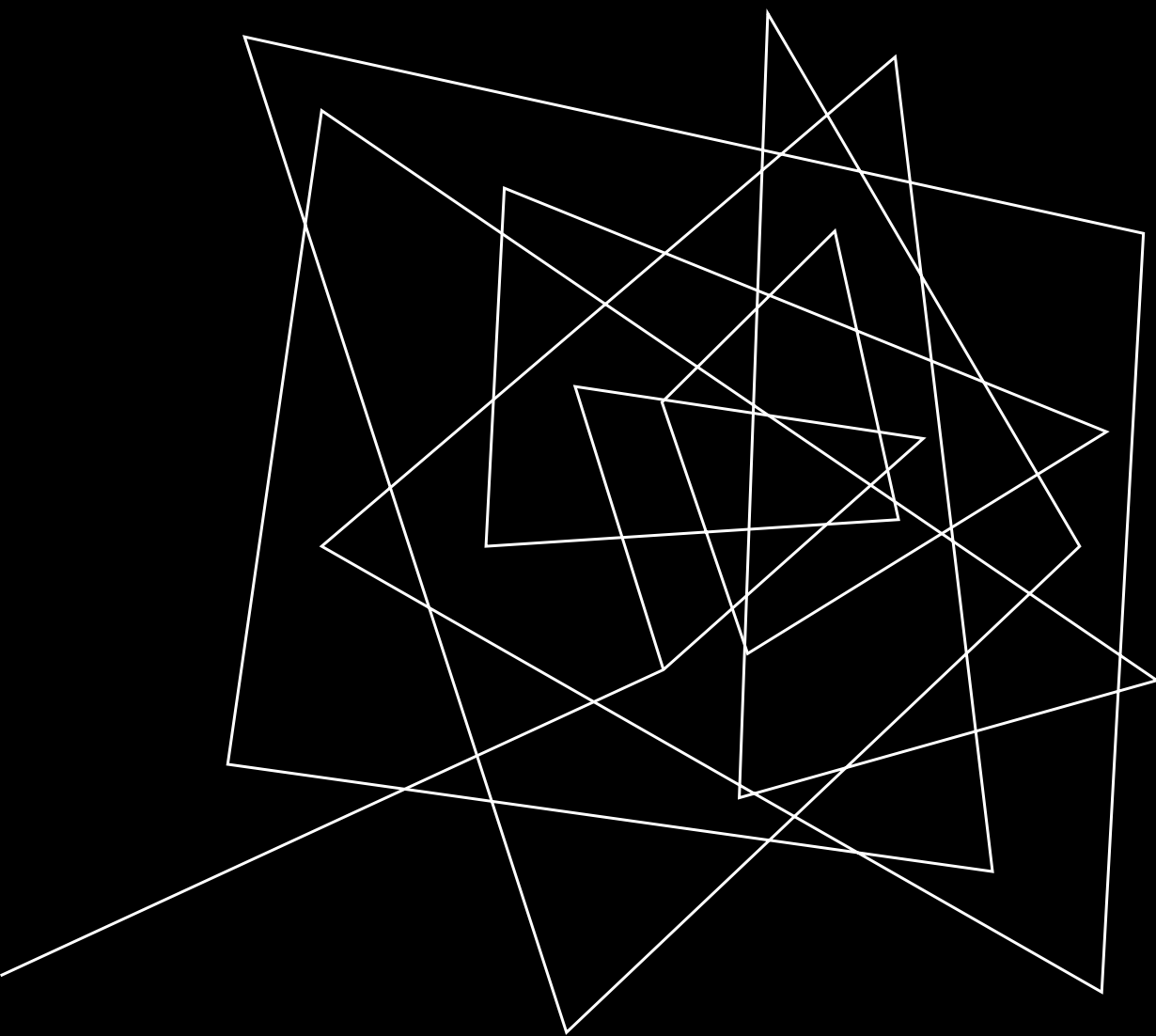
```
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
```

3. Verbindung testen:

- **ping google.com**

2.10. ZUSAMMENFASSUNG

- **Netzwerke ermöglichen die Kommunikation zwischen Geräten.**
- **IP-Adressen sind zentrale Elemente der Netzwerkidentifikation.**
- **Verwaltung von Netzwerken unter Linux erfolgt durch Befehle wie `ip`, `ifconfig`, und `ping`.**



3. NETZWERKTOOLS UND -BEFEHLE

Verwendung von ping, ifconfig, netstat, ssh,
scp

3. NETZWERKTOOLS UND - BEFEHLE

- 3.1. Übersicht
- 3.2. Der Befehl ping
- 3.3. Der Befehl ifconfig
- 3.4. Der Befehl netstat
- 3.5. Der Befehl ssh
- 3.6. Der Befehl scp
- 3.7. Praxisbeispiele
- 3.8. Zusammenfassung

3.1. ÜBERSICHT

- **Netzwerktools und ihre Funktionen**
- **Häufig genutzte Befehle zur Netzwerkdiagnose und -verwaltung**
 - **ping**
 - **ifconfig**
 - **netstat**
 - **ssh**
 - **scp**

3.2. DER BEFEHL PING

Verwendung:

- **Überprüft die Netzwerkverbindung zu einem Host.**

Funktion:

- **Sendet ICMP-Echo-Anfragen und misst die Antwortzeit.**

Syntax:

- **ping [IP-Adresse oder Hostname]**

Beispiel:

- **ping google.com (Testet die Verbindung zu google.com)**

3.3. DER BEFEHL IFCONFIG

Verwendung:

- **Zeigt und konfiguriert Netzwerkinterfaces.**

Funktion:

- **Veraltet, aber noch häufig verwendet.**

Syntax:

- **ifconfig [Schnittstelle]**

Beispiel:

- **ifconfig eth0 (Zeigt Informationen über die Schnittstelle eth0)**

3.4. DER BEFEHL NETSTAT

Verwendung:

- Zeigt Netzwerkverbindungen, Routing-Tabellen und Statistiken.

Funktion:

- Hilfreich für die Analyse aktiver Verbindungen und des Netzwerkstatus.

Syntax:

- `netstat [Optionen]`

Wichtige Optionen:

- `-a`: Zeigt alle Verbindungen
- `-t`: Zeigt TCP-Verbindungen
- `-u`: Zeigt UDP-Verbindungen

Beispiel:

- `netstat -at` (Zeigt alle TCP-Verbindungen)

3.5. DER BEFEHL SSH

Verwendung:

- Stellt eine sichere Verbindung zu einem Remote-Server her.

Funktion:

- Ermöglicht Fernzugriff auf andere Computer im Netzwerk.

Syntax:

- **ssh [Benutzername]@[Hostname oder IP-Adresse]**

Beispiel:

- **ssh max@192.168.1.100** (Stellt eine SSH-Verbindung zum Benutzer max auf dem Host 192.168.1.100 her)

3.6. DER BEFEHL SCP

Verwendung:

- **Kopiert Dateien sicher zwischen Rechnern.**

Funktion:

- **Nutzt SSH zur sicheren Übertragung von Dateien.**

Syntax:

- **scp [Quelldatei] [Zielort]**

Beispiele:

- **Datei vom lokalen Rechner auf einen Remote-Server kopieren:**
 - **scp datei.txt max@192.168.1.100:/home/max/**
- **Datei von einem Remote-Server auf den lokalen Rechner kopieren:**
 - **scp max@192.168.1.100:/home/max/datei.txt .**

3.7. PRAXISBEISPIELE

1. Verbindung testen:

- **ping 8.8.8.8**

2. Netzwerkinterface anzeigen:

- **ifconfig eth0**

3. Netzwerkverbindungen überwachen:

- **netstat -atu**

4. SSH-Verbindung herstellen:

- **ssh user@remote-server.com**

5. Datei mit scp übertragen:

- **scp dokument.txt user@remote-server.com:/home/user/**

3.8. ZUSAMMENFASSUNG

- **ping: Verbindungstest zwischen Rechnern.**
 - **ifconfig: Veraltet, zeigt und konfiguriert Netzwerkinterfaces.**
 - **netstat: Überwacht Netzwerkverbindungen und Statistiken.**
 - **ssh: Sicherer Fernzugriff auf entfernte Rechner.**
 - **scp: Sicherer Dateitransfer zwischen Rechnern im Netzwerk.**



4. EINFÜHRUNG IN DEN PAKETMANAGER

Verwalten von Software mit apt oder yum,
Paketinstallation, -aktualisierung und -entfernung.

4. EINFÜHRUNG IN DEN PAKETMANAGER

- 4.1. Übersicht
- 4.2. Der Paketmanager
- 4.3. Der Befehl apt
- 4.4. Paketinstallation mit apt
- 4.5. Paketaktualisierung mit apt
- 4.6. Pakete entfernen mit apt
- 4.7. Der Befehl yum
- 4.8. Paketinstallation mit yum
- 4.9. Paketaktualisierung mit yum
- 4.10. Pakete entfernen mit yum
- 4.11. Zusammenfassung

4.1. ÜBERSICHT

- **Was ist ein Paketmanager?**
- **Verwalten von Software auf Linux-Systemen**
- **Einführung in die Paketmanager:**
 - **apt (Debian/Ubuntu)**
 - **yum (RHEL/CentOS)**

4.2. DER PAKETMANAGER

Was ist ein Paketmanager?

- **Ein Werkzeug, um Software zu installieren, zu aktualisieren und zu entfernen.**
- **Verwaltung von Paketabhängigkeiten.**
- **Sucht, lädt und installiert Pakete aus Repositories.**

Ziele:

- **Vereinfachung der Softwareinstallation.**
- **Gewährleistung von Sicherheits- und Stabilitätsupdates.**

4.3. DER BEFEHL APT

Verwendung:

- **apt ist der Paketmanager für Debian-basierte Systeme (z.B. Ubuntu).**

Wichtige Funktionen:

- **Installation von Paketen**
- **Aktualisierung von Paketen**
- **Entfernung von Paketen**

Syntax:

- `apt [Optionen] [Paketname]`

4.4. PAKETINSTALLATION MIT APT

Installation eines neuen Pakets:

- **Syntax:**
 - **sudo apt install [Paketname]**

Beispiel:

- **sudo apt install vim (Installiert den Texteditor vim)**

4.5. PAKETAKTUALISIERUNG MIT APT

Aktualisierung der Paketliste:

- **Syntax:**
 - **sudo apt update**

Aktualisierung aller installierten Pakete:

- **Syntax:**
 - **sudo apt upgrade**

Beispiel:

- **sudo apt update && sudo apt upgrade (Aktualisiert die Paketliste und die installierte Software)**

4.6. PAKETE ENTFERNEN MIT APT

Entfernen eines Pakets:

- **Syntax:**
 - **sudo apt remove [Paketname]**

Beispiel:

- **sudo apt remove vim** (Entfernt das Paket vim)

Vollständiges Entfernen inkl. Konfigurationsdateien:

- **Syntax:**
 - **sudo apt purge [Paketname]**

4.7. DER BEFEHL YUM

Verwendung:

- **yum ist der Paketmanager für RHEL-basierte Systeme (z.B. CentOS, Fedora).**

Funktionalität:

- **Verwaltung von RPM-Paketen.**
- **Installation, Aktualisierung und Entfernung von Paketen.**

Syntax:

- **yum [Optionen] [Paketname]**

4.8. PAKETINSTALLATION MIT YUM

Installation eines neuen Pakets:

- **Syntax:**
 - **sudo yum install [Paketname]**

Beispiel:

- **sudo yum install vim (Installiert den Texteditor vim)**

4.9. PAKETAKTUALISIERUNG MIT YUM

Aktualisierung aller Pakete:

- **Syntax:**
 - **sudo yum update**

Beispiel:

- **sudo yum update (Aktualisiert alle installierten Pakete)**

4.10. PAKETE ENTFERNEN MIT YUM

Entfernen eines Pakets:

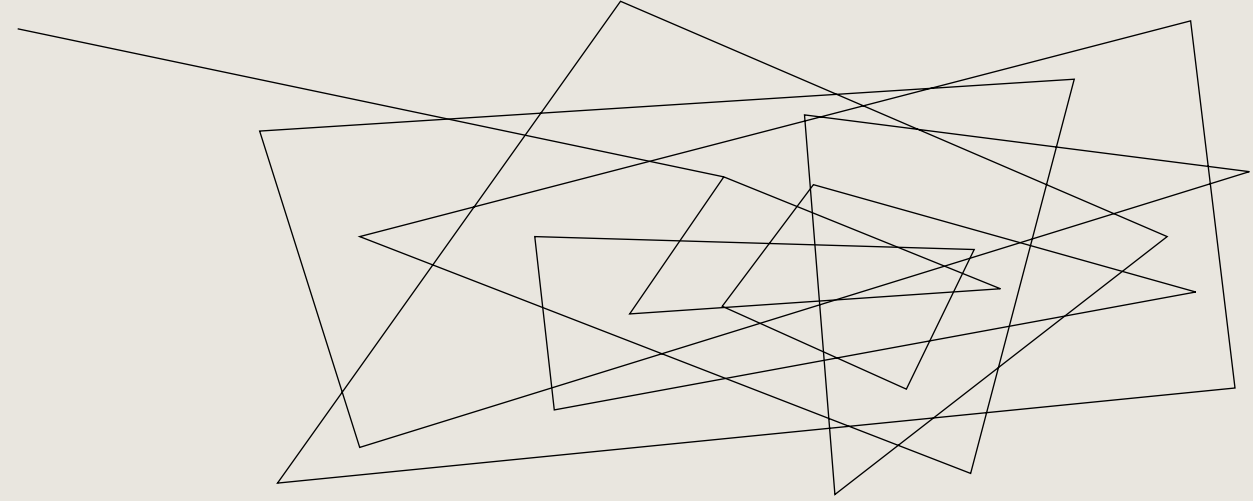
- **Syntax:**
 - **`sudo yum remove [Paketname]`**

Beispiel:

- **`sudo yum remove vim` (Entfernt das Paket vim)**

4.11. ZUSAMMENFASSUNG

- **apt für Debian-basierte Systeme (z.B. Ubuntu)**
- **yum für RHEL-basierte Systeme (z.B. CentOS)**
- **Wichtige Funktionen:**
 - **Installation: install**
 - **Aktualisierung: update / upgrade**
 - **Entfernung: remove**





DANKE

Costis Aivalis

costis.Aivalis@gmail.com

aivalis.eu