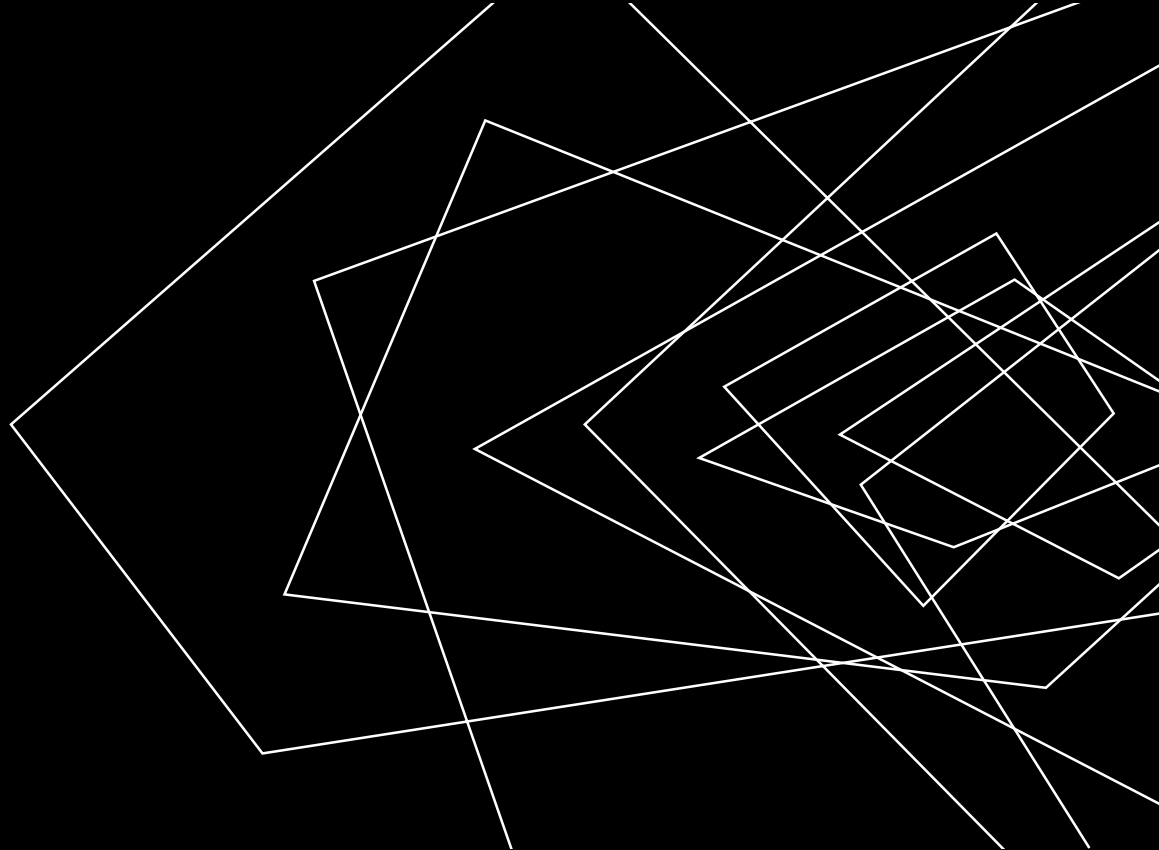
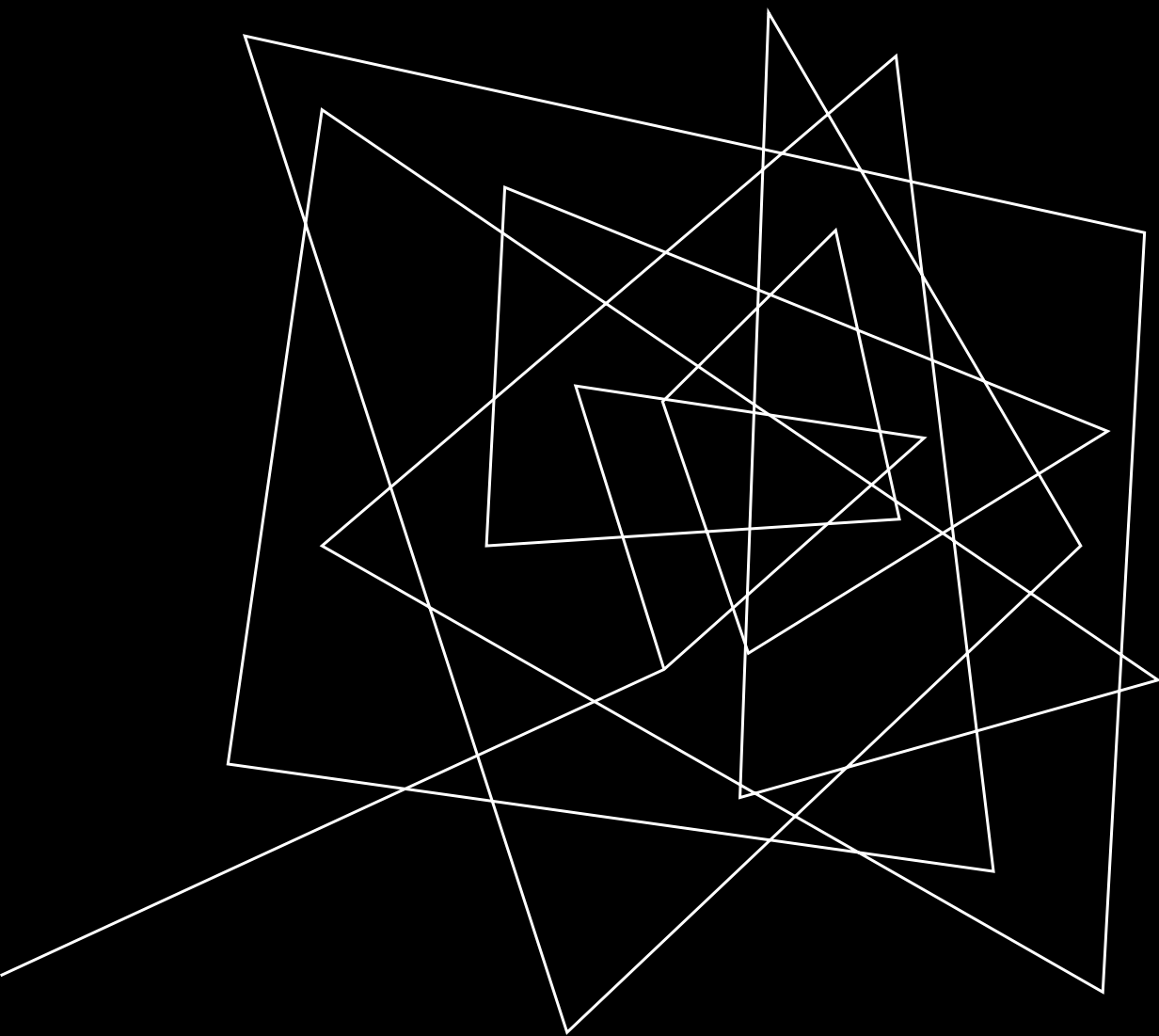


LINUX - TAG 2

AGENDA

1. Dateimanagement
2. Textbearbeitung mit Nano und Vim
3. Einführung in Dateisystemberechtigungen
4. Komprimierung und Archivierung





1. DATEIMANAGEMENT

Grundlegende Befehle und Dateiberechtigungen

1. DATEIMANAGEMENT IN LINUX

- 1.1. Übersicht
- 1.2. Der Befehl cp
- 1.3. Der Befehl mv
- 1.4. Der Befehl rm
- 1.5. Der Befehl touch
- 1.6. Dateiberechtigungen
- 1.7. Dateiberechtigungen ändern
- 1.8. Dateieigenschaften
- 1.9. Zusammenfassung

1.1. ÜBERSICHT

- **cp**: Dateien kopieren
- **mv**: Dateien verschieben oder umbenennen
- **rm**: Dateien löschen
- **touch**: Leere Dateien erstellen oder Zeitstempel ändern
- **Dateiberechtigungen**: Zugriffsrechte für Dateien
- **Dateieigenschaften**: Größe, Besitzer, Zeitstempel

1.2. DER BEFEHL CP

Verwendung:

- **cp [Quelle] [Ziel]**

Beispiel:

- **cp datei.txt /home/benutzer/** (Kopiert datei.txt nach /home/benutzer/)

Optionen:

- **-r:** Rekursiv, kopiert auch Verzeichnisse
- **-i:** Fragt vor dem Überschreiben
- **-v:** Zeigt Details des Kopiervorgangs

1.3. DER BEFEHL MV

Verwendung:

- **mv [Quelle] [Ziel]**

Beispiel:

- **mv datei.txt dokumente/** (Verschiebt datei.txt in den Ordner dokumente/)
- **mv datei.txt neue_datei.txt** (Benennt datei.txt in neue_datei.txt um)

Optionen:

- **-i:** Warnt vor dem Überschreiben
- **-v:** Zeigt Details des Verschiebevorgangs

1.4. DER BEFEHL RM

Verwendung:

- **rm [Datei]**

Beispiel:

- **rm datei.txt** (Löscht datei.txt)

Optionen:

- **-r**: Rekursiv, löscht Verzeichnisse
- **-f**: Erzwingt das Löschen ohne Bestätigung
- **-i**: Fragt vor jedem Löschen

1.5. DER BEFEHL TOUCH

Verwendung:

- **touch [Dateiname]**

Beispiel:

- **touch neue_datei.txt** (Erstellt eine leere Datei neue_datei.txt)

Zusätzliche Funktion:

- Ändert den Zeitstempel einer vorhandenen Datei

1.6. DATEIBERECHTIGUNGEN

Drei Ebenen von Berechtigungen:

- **Lesen (r):** Datei oder Verzeichnis anzeigen
- **Schreiben (w):** Datei ändern oder in Verzeichnis schreiben
- **Ausführen (x):** Datei ausführen oder Verzeichnis betreten

Benutzergruppen:

- **Besitzer:** Der Ersteller der Datei
- **Gruppe:** Zugehörige Benutzergruppe
- **Andere:** Alle anderen Benutzer

1.7. DATEIBERECHTIGUNGEN ÄNDERN

Der Befehl chmod:

- **chmod [Modus] [Datei]**

Beispiel:

- **chmod 755 datei.txt (Setzt Berechtigungen auf rwxr-xr-x)**

Bedeutung:

- **7: Lesen, Schreiben, Ausführen**
- **5: Lesen, Ausführen**

1.8. DATEIEIGENSCHAFTEN

Der Befehl ls -l: Zeigt Dateieigenschaften:

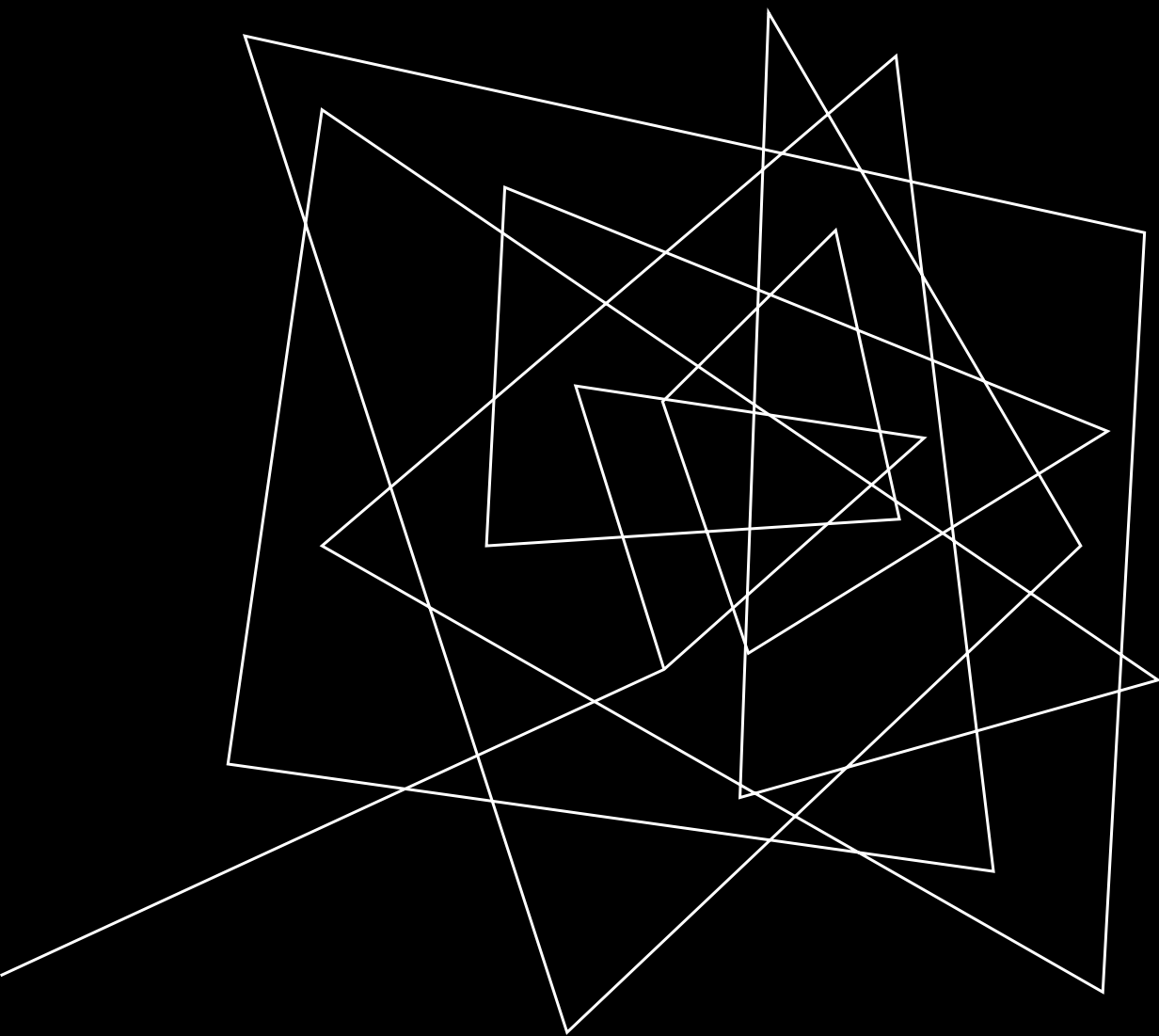
- **Berechtigungen**
- **Anzahl der Links**
- **Besitzer**
- **Gruppe**
- **Dateigröße**
- **Zeitstempel der letzten Änderung**

Beispiel:

-rw-r--r-- 1 benutzer gruppe 1024 Sep 17 12:00 datei.txt

1.9. ZUSAMMENFASSUNG

- **cp: Dateien kopieren**
- **mv: Dateien verschieben/umbenennen**
- **rm: Dateien löschen**
- **touch: Dateien erstellen/aktualisieren**
- **chmod: Dateiberechtigungen ändern**
- **ls: Dateieigenschaften anzeigen**



2. TEXTBEARBEITUNG MIT NANO UND VIM

Grundlagen der Texteditoren Nano und Vim, einfache
Textbearbeitung

2. TEXTBEARBEITUNG MIT NANO UND VIM

- 2.1. Übersicht
- 2.2. Der Texteditor Nano
- 2.3. Grundlegende Nano-Befehle
- 2.4. Nano - Weitere Funktionen
- 2.5. Der Texteditor Vim
- 2.6. Vim - Modi
- 2.7. Grundlegende Vim-Befehle
- 2.8. Vim - Weitere Funktionen
- 2.9. Unterschiede zwischen Nano und Vim
- 2.10. Zusammenfassung

2.1. ÜBERSICHT

- Einführung in Nano und Vim
- Grundlegende Befehle in beiden Editoren
- Unterschiede und Anwendungsbereiche
- Praktische Beispiele für einfache Textbearbeitung

2.2. DER TEXTEDITOR NANO

Einführung:

- **Einfache und benutzerfreundliche Bedienung**
- **Ideal für Einsteiger**

Starten von Nano:

- **nano [Dateiname]**

Beispiel:

- **nano textdatei.txt**

2.3. GRUNDLEGENDE NANO-BEFEHLE

Navigation:

- **Pfeiltasten: Bewegen im Text**

Textbearbeitung:

- **Strg + O: Datei speichern**
- **Strg + X: Editor beenden**
- **Strg + K: Zeile ausschneiden**
- **Strg + U: Zeile einfügen**

2.4. NANO - WEITERE FUNKTIONEN

- **Strg + W: Suchen im Text**
- **Strg + G: Hilfemenü anzeigen**
- **Strg + C: Aktuelle Cursorposition anzeigen**
- **Alt + A: Text markieren**

Praktisches Beispiel:

- Datei öffnen, bearbeiten, speichern und beenden mit den erlernten Befehlen.

2.5. DER TEXTEDITOR VIM

Einführung:

- **Leistungsfähiger, modal aufgebauter Editor**
- **Hat eine steilere Lernkurve, aber viele fortgeschrittene Funktionen**

Starten von Vim:

- **vim [Dateiname]**

Beispiel:

- **vim textdatei.txt**

2.6. VIM - MODI

Normale (Befehls-)Modus:

- **Standardmodus nach dem Start**
- **Zum Navigieren und Befehle ausführen**

Einfügemodus:

- **Textbearbeitung, wie in einem klassischen Editor**
- **i: In den Einfügemodus wechseln**

Beispiel:

- **i: Text schreiben**
- **Esc: Zurück in den normalen Modus**

2.7. GRUNDLEGENDE VIM-BEFEHLE

Navigation:

h, j, k, l: Bewegung im Text (links, runter, hoch, rechts)

:w Datei speichern

:q Editor beenden

:wq Speichern und beenden (alternativ: **ZZ**)

:q! Beenden ohne Speichern Aenderungen werden nicht gespeichert

:q Beenden ohne Speichern wenn es keine Aenderungen gibt

2.8. VIM - WEITERE FUNKTIONEN

- **dd: Zeile löschen**
- **yy: Zeile kopieren**
- **p: Einfügen**
- **Strg-G: info**
- **/suchbegriff: Text durchsuchen**

Praktisches Beispiel:

- **Textdatei öffnen, bearbeiten, speichern und navigieren mit den Vim-Befehlen.**

2.9. UNTERSCHIEDE ZWISCHEN NANO UND VIM

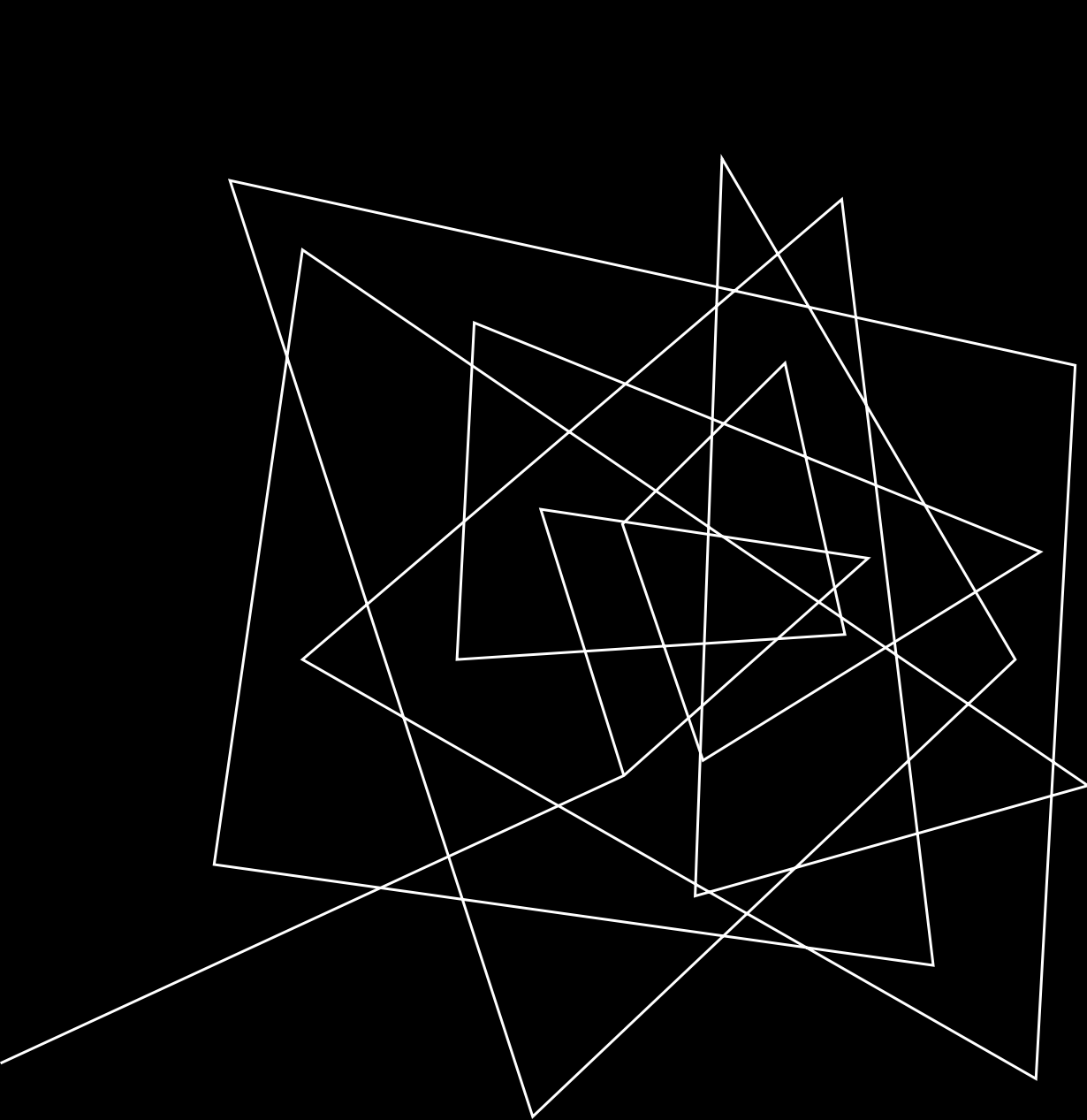
- **Nano:** Einfach zu bedienen, für schnelle Änderungen ideal
- **Vim:** Mächtig und flexibel, für komplexere Bearbeitungen geeignet
- Nano ist **modal** (kein Wechsel zwischen Modi), Vim nutzt Modi zur Bearbeitung.

Empfehlung:

- **Nano für Anfänger und schnelle Änderungen**
- **Vim für Benutzer, die komplexe Aufgaben effizient erledigen wollen**

2.10. ZUSAMMENFASSUNG

- **Nano:** Einfache, intuitive Bedienung mit grundlegenden Bearbeitungsfunktionen
- **Vim:** Mächtiger, modal aufgebauter Editor mit fortgeschrittenen Funktionen
- **Textbearbeitung:** In beiden Editoren leicht möglich, abhängig vom Erfahrungsgrad und den Anforderungen



3. EINFÜHRUNG IN DATEISYSTEM- BERECHTIGUNGEN

Berechtigungen, Eigentum, Befehle wie chmod, chown, chgrp

3. EINFÜHRUNG IN DATEISYSTEMBERECHTIGUNGEN

- 3.1. Übersicht
- 3.2. Dateiberechtigungen
- 3.3. Dateiberechtigungen anzeigen
- 3.4. Der Befehl `chmod`
- 3.5. Der Befehl `chown`
- 3.6. Der Befehl `chgrp`
- 3.7. Praxisbeispiele
- 3.8. Zusammenfassung

3.1. ÜBERSICHT

- **Dateiberechtigungen und Eigentum in Linux**
- **Zugriffsebenen: Benutzer, Gruppe, Andere**
- **Befehle zur Verwaltung von Berechtigungen**
 - **chmod, chown, chgrp**

3.2. DATEIBERECHTIGUNGEN

Grundkonzept:

- **Jede Datei hat bestimmte Berechtigungen:**
 - **Lesen (r):** Datei anzeigen
 - **Schreiben (w):** Datei ändern
 - **Ausführen (x):** Datei ausführen oder Verzeichnis betreten

Zugriffsebenen:

- **Benutzer (Owner):** Der Dateibesitzer
- **Gruppe:** Benutzer in der gleichen Gruppe wie der Besitzer
- **Andere:** Alle anderen Benutzer

3.3. DATEIBERECHTIGUNGEN ANZEIGEN

Der Befehl ls -l:

- Zeigt Dateiberechtigungen in einer langen Liste an.
- Beispiel:
`-rw-r--r-- 1 benutzer gruppe 1024 Sep 17 12:00 datei.txt`

Interpretation:

- **rw**x für den Besitzer: Lesen, Schreiben, Ausführen
- **r--** für die Gruppe: Nur lesen
- **r--** für Andere: Nur lesen

3.4. DER BEFEHL CHMOD

Verwendung:

- **Ändert die Berechtigungen von Dateien und Verzeichnissen.**

Symbolische Methode:

- **chmod [Wer]+[Berechtigung] [Datei]**
 - **Beispiel: chmod u+x datei.txt (Fügt dem Benutzer die Ausführberechtigung hinzu)**

Oktale Methode:

- **chmod [Modus] [Datei]**
 - **Beispiel: chmod 755 datei.txt (Berechtigungen rwxr-xr-x)**

Oktale Werte:

- **7: Lesen, Schreiben, Ausführen**
- **6: Lesen, Schreiben**
- **5: Lesen, Ausführen**
- **4: Nur Lesen**

3.5. DER BEFEHL CHOWN

Verwendung:

- **Ändert den Eigentümer einer Datei oder eines Verzeichnisses.**
- **chown [neuerBesitzer] [Datei]**

Beispiel:

- **chown benutzername datei.txt**

Zusätzlich:

- **Man kann sowohl den Besitzer als auch die Gruppe ändern:**
 - **chown benutzername:gruppe datei.txt**

3.6. DER BEFEHL CHGRP

Verwendung:

- **Ändert nur die Gruppe einer Datei oder eines Verzeichnisses.**
- **chgrp [gruppe] [Datei]**

Beispiel:

- **chgrp entwickler datei.txt**

3.7. PRAXISBEISPIELE

1. Dateiberechtigungen setzen:

- **chmod 644 dokument.txt** (Nur der Besitzer kann schreiben, alle anderen können nur lesen)

2. Besitzer ändern:

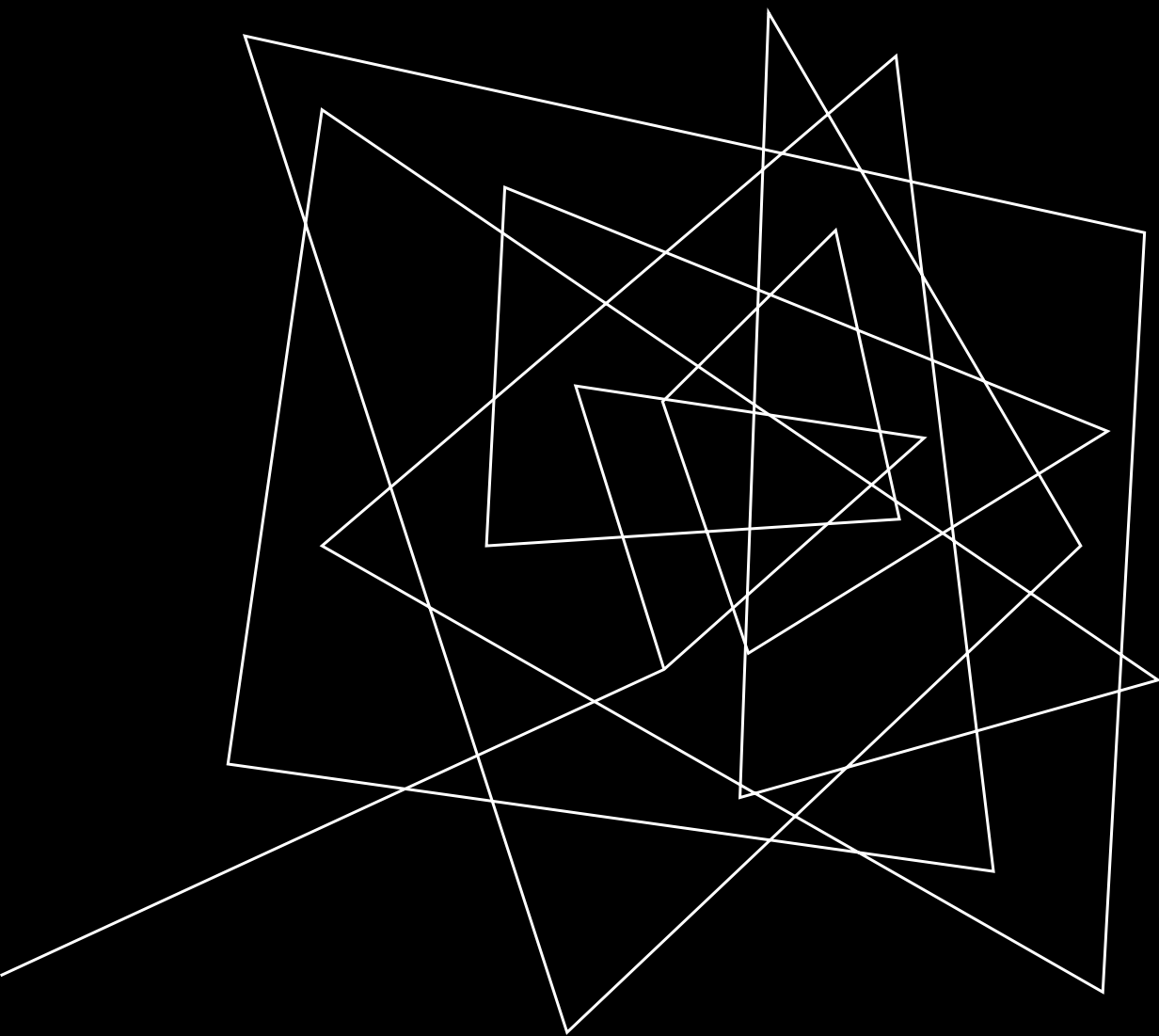
- **chown max dokument.txt** (Ändert den Besitzer auf "max")

3. Gruppe ändern:

- **chgrp team dokument.txt** (Weist die Gruppe "team" zu)

3.8. ZUSAMMENFASSUNG

- **Berechtigungen: Kontrollieren den Zugriff auf Dateien**
- **Befehle:**
 - **chmod: Berechtigungen ändern**
 - **chown: Besitzer ändern**
 - **chgrp: Gruppe ändern**
- **Sicherheitsrelevanz: Richtige Berechtigungen erhöhen die Sicherheit**



4. KOMPRIMIERUNG UND ARCHIVIERUNG

Befehle wie tar, gzip, zip, unzip, Arbeiten mit
komprimierten Dateien

4. KOMPRIMIERUNG UND ARCHIVIERUNG

- 4.1. Übersicht
- 4.2. Der Befehl tar
- 4.3. Der Befehl tar mit Komprimierung
- 4.4. Der Befehl gzip
- 4.5. Der Befehl zip
- 4.6. Der Befehl unzip
- 4.7. Vergleich von tar, gzip, und zip
- 4.8. Praxisbeispiele
- 4.9. Zusammenfassung

4.1. ÜBERSICHT

- **Komprimierung: Reduzierung der Dateigröße**
- **Archivierung: Mehrere Dateien in einer Datei zusammenfassen**
- **Wichtige Befehle:**
 - **tar, gzip, zip, unzip**

4.2. DER BEFEHL TAR

Verwendung:

- **Archiviert mehrere Dateien und Verzeichnisse in einer Datei.**
- **Syntax:**
 - **tar -cvf [Archivname.tar] [Dateien/Verzeichnisse]**

Optionen:

- **-c: Erstellen eines neuen Archivs**
- **-v: Zeigt den Fortschritt an**
- **-f: Gibt den Dateinamen des Archivs an**

Beispiel:

- **tar -cvf archiv.tar verzeichnis/ (Erstellt ein Archiv von verzeichnis/)**

4.3. DER BEFEHL TAR MIT KOMPRIMIERUNG

Verwendung mit Komprimierung:

- **Kombiniert Archivierung mit Komprimierung, oft mit gzip oder bzip2.**

Syntax:

- **tar -czvf [Archivname.tar.gz] [Dateien/Verzeichnisse]**

Optionen:

- **-z: Mit gzip komprimieren**
- **-j: Mit bzip2 komprimieren**

Beispiel:

- **tar -czvf archiv.tar.gz verzeichnis/ (Erstellt und komprimiert das Archiv)**

4.4. DER BEFEHL GZIP

Verwendung:

- **Komprimiert einzelne Dateien.**

Syntax:

- **gzip [Dateiname]**

Beispiel:

- **gzip dokument.txt (Komprimiert dokument.txt zu dokument.txt.gz)**

Entkomprimierung:

- **gunzip [Dateiname.gz]**

4.5. DER BEFEHL ZIP

Verwendung:

- **Erstellt ZIP-Archive und komprimiert mehrere Dateien.**

Syntax:

- **zip [Archivname.zip] [Dateien/Verzeichnisse]**

Beispiel:

- **zip archiv.zip datei1.txt datei2.txt (Erstellt ein ZIP-Archiv)**

Optionen:

- **-r: Rekursiv, fügt alle Dateien in Unterverzeichnissen hinzu**

4.6. DER BEFEHL UNZIP

Verwendung:

- **Entpackt ZIP-Archive.**

Syntax:

- **unzip [Archivname.zip]**

Beispiel:

- **unzip archiv.zip (Entpackt den Inhalt von archiv.zip)**

4.7. VERGLEICH VON TAR, GZIP, UND ZIP

- **tar: Nur Archivierung; oft in Kombination mit gzip oder bzip2 für Komprimierung**
- **gzip: Effiziente Komprimierung einzelner Dateien**
- **zip: Archivierung und Komprimierung in einem Schritt**

4.8. PRAXISBEISPIELE

1. Erstellen eines Archivs mit tar:

- **tar -cvf archiv.tar /pfad/zum/verzeichnis/**

2. Komprimieren mit gzip:

- **gzip datei.txt**

3. Erstellen eines ZIP-Archivs:

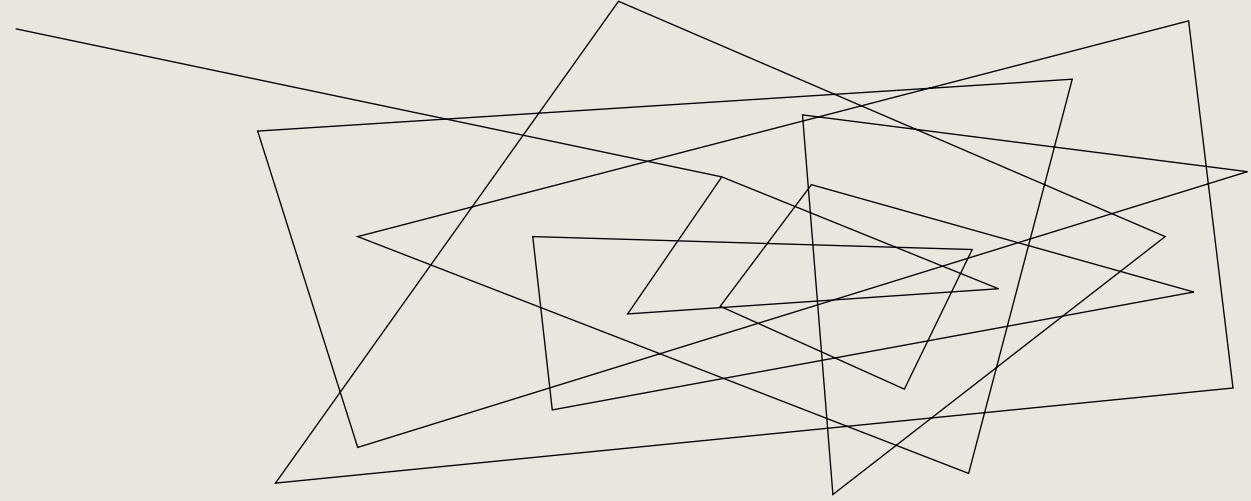
- **zip archiv.zip datei1.txt datei2.txt**

4. Entpacken eines ZIP-Archivs:

- **unzip archiv.zip**

4.9. ZUSAMMENFASSUNG

- **tar: Archivieren und Komprimieren mehrerer Dateien**
- **gzip: Komprimiert einzelne Dateien**
- **zip und unzip: Archivierung und Komprimierung von Dateien**
- **Komprimierung spart Speicherplatz und erleichtert den Dateitransfer.**





DANKE

Costis Aivalis

costis.Aivalis@gmail.com

aivalis.eu