

Partnerarbeit:

(Lest euch die Aufgaben gemeinsam gut durch und bearbeitet innerhalb von Linux folgende Fragestellungen)

1. Dateien und Verzeichnisse:

- 1.1. Lies die Manual-Page des Befehls **ls** und finde heraus:
 - Wie kannst Du Dir ein langes Listing anzeigen lassen?
 - Wie kannst Du Dir die Dateien der Größe/der Modifikationszeit nach anzeigen lassen?Anschließend wechsele in das Verzeichnis **/etc/** und lass Dir ein langes Listing der Dateien in aufsteigender Größe anzeigen.
- 1.2. Führe folgenden Aktionen in der Konsole aus:
 - Erzeuge ein neues Verzeichnis mit dem Namen **verzeichnis**, und wechsele hinein.
 - Erzeuge in dem Verzeichnis **verzeichnis** eine Datei namens **datei** und lass Dir ein langes Listing anzeigen.
 - Schreibe mit einem Konsoleneditor (z.B. vi oder nano) in die Datei **datei** Deinen Namen hinein.
 - Benenne nun die Datei **datei** in **neue datei** um (achte auf die richtige Syntax wegen des Leerzeichens!).
 - Versuche das erzeugte Verzeichnis **verzeichnis** mit Hilfe des Befehls **rmdir** zu löschen – warum geht es nicht? Mit welchem Befehl funktioniert es?
- 1.3. Erzeuge vier Dateien mit den Namen **datei1** bis **datei4** und führe folgende Aktionen aus:
 - Setze die Berechtigungen auf **datei1** so, dass Du sie lesen und schreiben kannst, aber sonst niemand Rechte hat.
 - Setze die Berechtigungen auf **datei2** so, dass sie alle lesen, aber niemand schreiben kann. Wer kann trotzdem in die Datei schreiben?
 - Setze die Berechtigungen auf **datei3** so, dass Benutzer aus derselben Gruppe Lese- und Schreibrechte haben, alle anderen nur Leserechte.
 - Setze die Berechtigungen auf **datei4** so, dass sie alle Benutzer ausführen können, aber nur Du die Datei auch lesen und schreiben kannst.
- 1.4. Wie groß ist das größte Unterverzeichnis von **/usr/include**?
- 1.5. Wie voll ist die Partition, auf der das Root-Verzeichnis gemountet ist? (d.h. wieviel Plattenplatz ist dort noch frei?)

2. Harte und symbolische Links:

- 2.1. Erstelle in Deinem Homeverzeichnis eine Datei namens **file**. Lege sowohl einen Hardlink **hardlink** als auch einen Symlink **symlink** auf diese Datei an. Woran erkennst Du, wie viele harte Links auf die Datei **file** zeigen (es sollten inzwischen 2 sein)?
- 2.2. Schreibe das heutige Datum (ohne Zuhilfenahme eines Editors) in die Datei **file**. Zeige den Inhalt von **file** an. Dann zeige sowohl den Inhalt von **hardlink** als auch von **symlink** an. Was passiert?
- 2.3. Entferne nun die Datei **file** und zeige wiederum den Inhalt von **hardlink** als auch von **symlink** an. Erkläre, warum das für **hardlink** geht und für **symlink** nicht.

3. Arbeiten mit Prozessen:

3.1. Lies die Manual-Page des Befehls **top** und finde heraus:

- Was macht der Befehl?
- Welche Informationen werden angezeigt?
- Wie kann man die Prozesse nach Speicherverbrauch in Prozent/nach CPU-Verbrauch in Prozent sortieren?

3.2. Lies die Manual-Page des Befehls **ps** und finde heraus:

- Wie kannst Du Dir alle Prozesse anzeigen lassen (nicht nur deine)?
- Wie kannst Du Dir alle Prozesse mit dem Namen **getty** anzeigen lassen?
- Wie kannst Du Dir mit ASCII-Art die Prozesshierarchie anzeigen lassen?

4. Shell-Programmierung:

4.1. Rufe das Skript **linuxscript1** auf. Es erstellt ein Verzeichnis **~/linux1** und darin einen Verzeichnisbaum. Wechsle in dieses Verzeichnis und finde in dem erstellten Verzeichnisbaum alle Dateien des Dateityps ***.txt**, die den Text **ba** beinhalten. (Hinweis: Für das Ausführen eines Kommandos für jedes Argument, das als Standardausgabe eines anderen Kommandos auftritt, eignet sich u.a. der Befehl **xargs**).

4.2. Rufe das Skript **linuxscript2** auf. Es erstellt ein Verzeichnis **~/linux2** und darin einige Dateien mit dem Namen **linux2file_***. Wechsle in dieses Verzeichnis und kopiere mit einem neuen Script **filemove** diese Dateien in das Unterverzeichnis **target**.