

ARBEITSBLATT – LÖSUNG

BETRIEBSSYSTEME – LINUX GRUNDLAGEN

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: **Konsolenbefehl „man ls“**

- Wie kannst Du Dir ein langes Listing anzeigen lassen? **„ls -l“ aka long listing**
- Wie kannst Du Dir die Dateien der Größe/der Modifikationszeit nach anzeigen lassen?
Nach Größe sortiert: „ls -S“ / nach Modifikationszeit „ls -u“
-u = nach Zugriffszeit / -t = nach Änderungszeit

Anschließend wechsele in das Verzeichnis **/etc/** und lass Dir ein langes Listing der Dateien in aufsteigender Größe anzeigen.

Verzeichnis wechseln: cd /etc **Langes Listing nach Größe: ls -l -S**

1.2. Führe folgenden Aktionen in der Konsole aus:

- Erzeuge ein neues Verzeichnis mit dem Namen **verzeichnis**, und wechsele hinein.
Verzeichnis erstellen: mkdir verzeichnis **hineinwechseln: cd /verzeichnis**
- Erzeuge in dem Verzeichnis **verzeichnis** eine Datei namens **datei** und lass Dir ein langes Listing anzeigen.
Erzeuge Datei: touch datei **langes Listing: ls -l**
- Schreibe mit einem Konsoleneditor (z.B. vi oder nano) in die Datei **datei** Deinen Namen hinein.
Start von Nano um Datei zu bearbeiten: nano datei **CTRL-X = Speichern**
- Benenne nun die Datei **datei** in **neue datei** um (achte auf die richtige Syntax wegen des Leerzeichens!).
Datei mit Leerzeichen umbenennen: mv datei neue\ datei
- Versuche das erzeugte Verzeichnis **verzeichnis** mit Hilfe des Befehls **rmdir** zu löschen – warum geht es nicht? Mit welchem Befehl funktioniert es?
Weil es nicht leer ist. Wenn doch verhindern Prozesse wie Spotlight das löschen.
Funktionierender Befehl: rm -rf verzeichnis

1.3. Erzeuge vier Dateien mit den Namen **datei1** bis **datei4** und führe folgende Aktionen aus:

Befehl: touch datei1-4

- Setze die Berechtigungen auf **datei1** so, dass Du sie lesen und schreiben kannst, aber sonst niemand Rechte hat.
Befehle: su (von Benutzer auf root wechseln, PW eingeben)
chmod u+w, u+r, g-w, g-r, o-w, o-r datei1
- Setze die Berechtigungen auf **datei2** so, dass sie alle lesen, aber niemand schreiben kann. Wer kann trotzdem in die Datei schreiben?
Befehle: chmod u-w, u+r, g-w, g+r, o-w, o+r datei2
root kann in die Datei schreiben
- Setze die Berechtigungen auf **datei3** so, dass Benutzer aus derselben Gruppe Lese- und Schreibrechte haben, alle anderen nur Leserechte.
Befehle: chmod u+w, u+r, g+w, g+r, o-w, o+r datei3
- Setze die Berechtigungen auf **datei4** so, dass sie alle Benutzer ausführen können, aber nur Du die Datei auch lesen und schreiben kannst.
Befehle: chmod u+w, u+r, u+rwx, g-w, g-r, g+rwx, o-w, o-r, o+rwx datei4

1.4. Wie groß ist das größte Unterverzeichnis von **/usr/include**?

Befehle: cd /usr/include **ls -l -S** **größtes Unterverzeichnis: GL 4,096KB**

- 1.5. Wie voll ist die Partition, auf der das Root-Verzeichnis gemountet ist? (d.h. wieviel Plattenplatz ist dort noch frei?)

Befehle: `df`

benutzt: 3728780, verfügbar: 15864384, Ben%: 20%, Eingehängt auf: /

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)?

Befehle: `touch file`

`ln file hardlink`

`ln -s file symlink`

Durch befehl ls Blauer hintergrund der Dateien

- 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?

Befehle: `echo 21.10.2010 > file`

`cat file`

`cat hardlink`

`cat symlink`

Hard- und Symlink haben Inhalt von File übernommen

- 1.1. 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.

Befehle: `rm file`

`cat hardlink`

`cat symlink`

In Symlink steht deswegen nichts mehr, weil es wie unter Windows ein normaler Shortcut ist und eine Quelle benötigt. Der Hardlink funktioniert deswegen, weil er eine Verzeichnis Referenz der Datei ist und viel näher der Datei steht als ein Shortcut