

Aufgabe 1 - Linux-Kommandos

Geben Sie ein Kommando an, mit dem Sie ...

1. Handbuchseiten („man pages“) öffnen können.

`man`

2. das aktuelle Verzeichnis in der Shell ausgeben können.

`pwd`

3. ein neues Verzeichnis erzeugen können.

`mkdir`

4. in ein Verzeichnis wechseln können.

`cd`

5. den Inhalt eines Verzeichnisses in der Shell ausgeben können.

`ls`

6. eine leere Datei erzeugen können.

`touch`

7. versuchen können den Inhalt einer Datei zu bestimmen.

`file`

8. den Inhalt verschiedener Dateien verknüpfen oder den Inhalt einer Datei ausgeben können.

`cat`

9. Zeilen vom Ende einer Datei in der Shell ausgeben können.

`tail`

10. Zeilen vom Anfang einer Datei in der Shell ausgeben können.

`head`

11. Dateien oder Verzeichnisse an eine andere Stelle kopieren können.

`cp`

12. Dateien oder Verzeichnisse an eine andere Stelle verschieben können.

`mv`

13. Dateien oder Verzeichnisse löschen können.

rm

14. ein leeres Verzeichnis löschen können.

rmdir

15. eine Zeichenkette in der Shell ausgeben können.

echo

16. die Dateirechte von Dateien oder Verzeichnissen ändern können.

chmod

17. Das Password eines Benutzers ändern können.

passwd

18. die laufende Sitzung (und damit auch die Shell) beenden und den Rückgabewert eines Shell-Skripts festzulegen können.

exit

19. das System neu starten können.

reboot oder alternativ shutdown

20. das System ausschalten können.

halt oder alternativ shutdown

21. einen neuen Benutzer erstellen können.

adduser

22. einen Benutzer löschen können.

deluser

23. einen Benutzer ändern können.

usermod

24. die Gruppenzugehörigkeiten des Benutzers ausgeben können.

groups

25. eine neue Gruppe erstellen können.

groupadd

26. eine Gruppe löschen können.

groupdel

27. eine Gruppe ändern können.

groupmod

28. den Benutzer (\Rightarrow Besitzer) ändern können, der einer Datei oder einem Verzeichnis zugeordnet ist.

chown

29. die Gruppe ändern können, die einer Datei oder einem Verzeichnis zugeordnet ist.

chgrp

30. einen „Link“ erstellen können.

ln

31. eine Datei nach den Zeilen durchsuchen können, die ein Suchmuster enthalten?

grep

32. eine Liste der laufenden Prozesse in der Shell ausgeben können.

ps

33. einen im Hintergrund der Shell laufenden Prozess in den Vordergrund holen können.

fg

34. einen Prozess in den Hintergrund der Shell verschieben können.

bg

35. einen Prozess beenden können.

kill

36. eine Gruppe von Prozessen beenden können.

killall

37. die Priorität eines neuen Prozesses festlegen können.

nice

38. die Priorität eines existierenden Prozesses ändern können.

renice

39. eine Liste der existierenden Prozesse als Baumstruktur in der Shell ausgeben können.

```
pstree
```

Aufgabe 2 – Dateien und Verzeichnisse

1. Erzeugen Sie in Ihrem Benutzerverzeichnis (Home-Verzeichnis) ein Verzeichnis BTS.

```
mkdir ~/BTS
```

2. Wechseln Sie in das Verzeichnis BTS und erzeugen Sie darin eine leere Datei mit dem Dateinamen `Datei1.txt` (Verwenden Sie zum Anlegen der Datei keinen Editor, sondern einen Kommandozeilenbefehl).

```
cd ~/BTS && touch Datei1.txt
```

3. Überprüfen Sie die Dateigröße der Datei `Datei1.txt`.

```
ls -lh Datei1.txt
```

4. Ändern Sie die Modifikationszeit der Datei `Datei1.txt` auf Ihr Geburtsdatum.

```
touch -t XXXXYZZAABB Datei1.txt
```

XXXX enthält das Jahr.

YY enthält den Monat.

ZZ enthält den Tag des Monats.

AA enthält die Stunde.

BB enthält die Minute.

5. Erzeugen Sie auf der Shell eine neue Datei `Datei2.txt` und fügen Sie in die neue Datei einen beliebigen mehrzeiligen Text als Inhalt ein (Verwenden Sie für das Einfügen des Textes keinen Editor, sondern einen Kommandozeilenbefehl).

```
echo -e "Zeile1\nZeile2" > Datei2.txt
```

6. Geben Sie die erste Zeile der Datei `Datei2.txt` aus.

```
head -n 1 Datei2.txt
```

7. Hängen den Inhalt von `Datei2.txt` an `Datei1.txt` an (Verwenden keinen Editor, sondern einen Kommandozeilenbefehl).

```
cat Datei2.txt >> Datei1.txt
```

8. Erzeugen Sie in Ihrem Home-Verzeichnis ein Verzeichnis mit dem Verzeichnisnamen `BS_Vorlesung`.

```
mkdir ~/BS_Vorlesung
```

9. Kopieren Sie die Dateien `Datei1.txt` und `Datei2.txt` aus dem Verzeichnis BTS in das Verzeichnis `BS_Vorlesung`.

```
cp ~/BTS/* ~/BS_Vorlesung
```

10. Löschen Sie das Verzeichnis BTS.

```
rm -rf ~/BTS
```

Aufgabe 3 – Hello World

Schreiben Sie ein C-Programm `hello.c`, das „Hello, World“ in die Standardausgabe ausgibt.

[`hello.c`](#)

Aufgabe 4 – Argumente

Schreiben Sie ein C-Programm `args.c`, das die Argumente der Kommandozeile in der Standardausgabe ausgibt.

[`args.c`](#)

Aufgabe 5 – Translate

Schreiben Sie ein C-Programm `translate.c`, das Zeichenketten aus der Standeingeabe einliest und nach einer Übersetzung in der Standardausgabe wieder ausgibt. Die Übersetzung wird durch zwei gleichlange Argumente der Kommandozeile `set1` und `set2` definiert und geschieht wie folgt, für den jeden Buchstaben der eingelesenen Zeichenketten wird in nachdem ersten Auftreten des Buchstabens in dem Argumente `set1` gesucht. Falls der Buchstabe nicht im Argument `set1` gefunden wird, bleibt der Buchstabe unverändert. Falls der Buchstabe im Argument `set1` an der i -ten Stelle zum ersten Mal auftritt, wird der Buchstabe durch den Buchstabe an der i -ten Stelle des Arguments `set2` ersetzt.

[`translate.c`](#)

Aufgabe 6 – Fork & Exec

1. Schreiben Sie ein C-Programm `fork.c`, das einen Kindprozess erzeugt, wobei der Elternprozess und der Kindprozess jeweils ihre Prozess-Id in der Standardausgabe ausgeben sollen.

[`fork.c`](#)

2. Schreiben Sie zwei C-Pogramme `parent.c` und `child.c`. Das Programm `parent.c` soll als erstes Argument den Pfad des kompilierten Programms `child.c` und ggf. weitere Argumente übergeben bekommen und dann dieses Programm mit den weiteren Argumenten als Kindprozess starten. Das Programm `child.c` soll lediglich die Argumente der Kommandozeile in der Standausgabe ausgeben (siehe Aufgabe 4).

[parent.c](#)
[child.c](#)