

Praktikum 1: Linux Shell

Die Lernziele in diesem Praktikum sind

- Kennenlernen der Linux Shell als Tool.
- Problemlösung mittels Shell-Kommandos.

Hinweise:

Weitere Infos finden sie in den Folien zur ersten Praktikumsaufgabe. Nutzen sie die angegebenen Tastenkombinationen zur Arbeitserleichterung in der Shell und die Namensergänzung mit der TAB-Taste.

Aufgabe 1 (Linux Shell)

Die traditionelle Benutzerschnittstelle zu einem Unix-basierten Betriebssystem ist Text-Konsole, die Shell. Im Linux finden sie dort typischerweise die `bash`. Starten sie ein Linux-System (im SWT-Pool, über X2GO, als VM, oder nativ) und öffnen Sie dort ein Terminal-Fenster mit der Shell (z.B.: CTRL-ALT-T).

- Ein Kommando um einen String auszugeben ist `echo`. Testen sie `echo "Hello World"`. Eine Übergabe von Argumenten an ein Programm per Kommandozeilenargumenten kennen sie noch aus OOP. Um einen String mit Leerzeichen als ein einzelnes Argument zu übergeben verwenden sie die Anführungszeichen.
- Neben den Argumenten gibt es auch Optionen, die mit `-` oder `--` beginnen. Mal dienen die Optionen nur als Flag, und mal benötigen sie zusätzlich ein Argument. Testen Sie `echo -e "Hello\nWorld"`. Was bewirkt `-e`, was passiert, wenn sie es auslassen?
- Eine Anleitung zu den Kommandos finden sie in den *Manpages*. Lesen sie nach: `man echo`

Aufgabe 2 (Grundlegende Linux Befehle: Dateien und Verzeichnisse)

Schreiben Sie für jede der folgenden Teilaufgaben eine einzeilige Shell-Anweisung, die die angeforderten Operationen ausführt.

1. Legen Sie im Home-Verzeichnis ein Verzeichnis mit dem Namen `uebung_bs` an.
2. Lassen Sie sich mit `ls` den Inhalt des aktuellen Verzeichnisses anzeigen. Der Ordner `uebung_bs` sollte angezeigt werden.
3. Wechseln Sie in das von Ihnen erstellte Verzeichnis `uebung_bs` und lassen Sie sich den Inhalt anzeigen.
4. Erzeugen Sie mit Hilfe des Befehls `echo > test.txt` eine neue Datei.
5. Erstellen Sie mittels `cp` einige Kopien Ihrer Datei. Geben Sie diesen Dateien die Namen `test_1.txt`, `test_2.txt`, `test_3.txt`, usw.
6. Erstellen Sie in Ihrem Home-Verzeichnis ein neues Unter-Verzeichnis mit dem Namen `bs_uebung`.
7. Verschieben Sie nun alle Dateien, die mit `test_` beginnen und mit `.txt` enden aus `uebung_bs` nach `bs_uebung`. (Das soll mit einem einzigen Befehl erreicht werden.)
8. Verifizieren Sie, dass nur noch die Datei `test.txt` in `uebung_bs` liegt und die übrigen Dateien in `bs_uebung`.

Hinweise:

Für die Schritte in dieser Aufgabe benötigen Sie zusätzlich die Befehle `cp`, `mv`. Verwenden Sie die Manpage der Befehle. Um mehrere Dateien zu kopieren oder zu verschieben können Sie Wildcards bzw. reguläre Ausdrücke verwenden. Sie finden Beispiele in den Praktikumsunterlagen.

Aufgabe 3 (Grundlegende Linux Befehle: Ausgabe umleiten, Pipes, Reguläre Ausdrücke)

In dieser Aufgabe bewegen sie sich mit der Shell durch den Verzeichnisbaum. Beachten Sie, wann sie einen absoluten Pfad brauchen (beginnend mit /) und wann sie einen relativen Pfad, ausgehen von dem Arbeitsverzeichnis, in dem sie sich befinden, benutzen. Mit `pwd` können sie das aktuelle Arbeitsverzeichnis der Shell abfragen, mittels `cd` bewegen sie sich im Verzeichnisbaum.

Schreiben Sie für jede der folgenden Teilaufgaben eine einzeilige Shell-Anweisung, die die angeforderten Operationen ausführt. Die Schritte in dieser Aufgabe können mit den Befehlen `ls`, `grep`, `cat`, `sort` und `mkdir` gelöst werden. Schauen sie in den Manpages nach, wenn Ihnen die Verwendung der Kommandos unklar ist. Bei Bedarf können Sie die Ein-/Ausgabe umleiten oder eine Pipe verwenden (siehe Folien).

- Listen Sie alle Dateien im Verzeichnis `/var/lib` im Standardformat auf (keine speziellen Parameter erforderlich).
- Listen Sie alle Dateien in `/usr/share/dict` im langen Format (`-l`-Flag) auf.
- Finden Sie alle Zeilen in `/usr/share/dict/ngerman`, die ein doppeltes kleines a (`,aa'`) enthalten.
- Finden Sie diese gleichen Zeilen wieder und speichern Sie die Wörter (als Teil derselben einzeiligen Shell-Anweisung) in der Datei `aa.txt`.
- Finden Sie alle Zeilen in derselben Datei, die ein doppeltes a haben, wobei das a entweder klein oder groß geschrieben sein kann (d.h. alle vier Kombinationen `,aa'`, `,aA'`, `,Aa'` und `,AA'` stimmen überein) und speichern Sie sie in der Datei `aa1.txt`.
- Wieviele Zeilen hat `aa.txt`, wieviele `aa1.txt`?
- Erstellen Sie ein Unterverzeichnis mit dem Namen `aufgabe1.2`.
- Erstellen Sie in diesem Unterverzeichnis eine Datei `oo.txt`, die alle Zeilen in `/usr/share/dict/ngerman` enthält, die ein doppeltes kleines o enthalten.
- Erstellen Sie in diesem Unterverzeichnis eine Datei `ooaa.txt`, die alle Zeilen aus `oo.txt` enthält, gefolgt von allen Zeilen aus `aa.txt`.
- Geben Sie in alphabetischer Reihenfolge alle Zeilen in `ooaa.txt` aus.
- Lösen Sie das Problem mittels Pipes und geben sie die verknüpften Kommandos an, mit denen sie direkt aus `/usr/share/dict/ngerman` alle Wörter mit `oo` oder `aa` sortiert ausgeben können.

Hinweise:

Sie dürfen Dateien nur in ihrem Home-Verzeichnis ablegen! Ein Shortcut für Ihr Home-Verzeichnis ist `~`.

Aufgabe 4 (Berechtigungen)

Informieren Sie sich über die Datei-Rechte in Unix/Linux.

1. Erstellen Sie eine `txt`-Datei in Ihrem Home-Verzeichnis und lassen Sie sich anschließend mittels `ls`-Befehl die Berechtigungsinformationen dazu anzeigen. Welche Rechte wurden vergeben?
2. Ändern Sie nun die Berechtigung für diese Datei auf `r-r-r-`.
3. Verifizieren Sie, dass der Inhalt der Datei nicht geändert werden kann (z.B. mit dem Editor `vi`).
4. Was müssen Sie tun, um die Datei wieder editieren zu können?