

## Aufgabe 1 – Einfache Befehle der Shell

**Teil 1:** Machen Sie sich (auch unter Nutzung des Hilfe-Systems) klar, was folgende Befehle bewirken und wie diese parametrisiert werden können.

**man:**

Der Befehl man gibt Informationen über andere Shell Befehle aus. Mit der „e“ und „y“ taste kann man nach unten und oben scrollen.

**pwd:**

Dieser Befehl gibt die aktuelle Working Directory aus.

**ls:**

Gibt den Inhalt des aktuellen Verzeichnisses aus. Der LS Befehl hat eine Vielzahl von Parametern.

**ls -l**

Gibt wie ls das Verzeichnis aus mit zusätzlichen Inhalten wie Änderungsdatum oder Besitzer

**ls -al:**

Der Parameter gibt zusätzlich versteckte Verzeichnisse und Dateien an.

**cd:**

Mit dem cd Befehl kann man in ein anderes Verzeichnis wechseln, durch Angabe des lokalen Pfades oder des globalen Pfades.

**mkdir:**

Dieser Befehl erstellt ein Verzeichnis in der aktuellen Work Direktion. Der Befehl kann mit -m, -p -v und -z parametrisiert werden.

**rm:**

Mit dem „rm“ Befehl können Dateien oder Verzeichnisse gelöscht werden. Der Befehl rm kann mit -f, -i, -l, -r, -d, -v parametrisiert werden.

**cp:**

Kopiert Dateien und Verzeichnisse in ein angegebenes Verzeichnis. Cp kann z.B. mit -a, --attributs-only, --backup, -b, --copy-contents, -d, -f, -i, -h, -l, und -L parametrisiert werden.

**mv:**

**more:**

Mit more kann der Inhalt einer Datei ausgegeben werden, wenn der Inhalt größer ist als das Terminal ausgeben kann, wird der Inhalt in teile eingeteilt. Die einzelnen Teilausgaben können mit der Leertaste durchlaufen werden können. More kann mit -d, -l, -f, -p, -c, -s, -u, -number, +number und +/string parametrisiert werden.

**tar:**

Mit dem tar Befehl können Dateien in ein Archiv gepackt werden oder aus einem Archiv ausgepackt werden. Tar kann z.b. mit -cf, -tvf und -xf parametrisiert werden.

### gzip:

Mit gzip können Dateien verpackt oder ausgepackt werden. Gzip kann z.B. mit -a, -c, -d, -f und -h parametrisiert werden.

### top:

Der top Befehl gibt alle laufenden Prozesse aus und wie viel Ressourcen sie verbrauchen. Top hat viele Parametrisierung Möglichkeiten z.B. -b, -c, -d, -H, -i und -n.

### uname:

Der uname Befehl gibt den Namen des Betriebssystems aus mit den Parametern -a, -s, -n, -r, -v, -m, -p, -i, und -o können zusätzliche Informationen ausgegeben werden.

### Was bewirkt der folgende Befehl?

`find . | xargs grep testString`

Der Befehl durchsucht alle Dateien nach dem String "testString". Dabei gibt der Befehl alle durchsuchten Verzeichnisse aus, sowie die Datei in der „testString“ drin steht.

**Teil 2:** Legen Sie eine sinnvolle Verzeichnisstruktur für das Praktikum in Ihrem Home-Verzeichnis an und wechseln Sie in Ihr Arbeitsverzeichnis für diese Übung. Kopieren Sie die Datei file.tar.gz aus Ilias in Ihr Arbeitsverzeichnis und entpacken Sie die Datei dort. Machen Sie sich anhand der entpackten Dateien mit den Metazeichen vertraut. Bestimmen Sie mit dem ls-Kommando die oberste Datei in Ihrem Arbeitsverzeichnis und löschen Sie diese mit rm (verwenden Sie für die Bestimmung der obersten Datei den Befehl head).

Anlegen der Verzeichnisstruktur und entpacken:

- `cd ~`
- `mkdir praktikum`
- `cd praktikum`
- `mkdir p01`
- `cd p01`
- `tar -xvzf file.tar.gz` (an sich reicht `tar -xf file.tar.gz`)
  - `tar -xf file.tar.gz -C /path/path` (ginge auch)
  - x: extract
  - v: verbose -> zeigt alle extrahierten Dateien an
  - z: dekomprimiert gzip
  - f: muss last flag sein, tar-Datei muss folgen

Die wichtigsten Meta-Zeichen sind:

- \* Eine Folge von keinem, einem oder mehreren Zeichen
- ? Ein einzelnes Zeichen
- [abc] Übereinstimmung mit einem beliebigen Zeichen in der Klammer
- [a-q] Übereinstimmung mit einem beliebigen Zeichen aus dem angegebenen Bereich

- [!abc] Übereinstimmung mit einem beliebigen Zeichen, das nicht in der Klammer ist
- ~ Home-Verzeichnis des aktuellen Benutzers
- ~name Home-Verzeichnis des Benutzers name
- ~+ Aktuelles Verzeichnis
- ~- Vorheriges Verzeichnis

⇒ Das ähnelt sehr stark den regulären Ausdrücken!

Der letzte Aufgabenteil war etwas merkwürdig, da mit Head nicht die oberste Datei bestimmt wird, sondern der Kopf des Inhalts einer Datei angezeigt wird. Es ließ sich allerdings das Ergebnis der ls Abfrage an head weiterleiten (*ls -a | head*).

Hier wird angezeigt, dass „1file1“ die oberste Datei im directory ist, welche mit *rm 1file1* gelöscht werden kann. Man muss aufpassen: Datei *\*file1* alleine lässt sich zwar mit *rm \*file1* löschen, allerdings werden auch die anderen Dateien mit file1 im Namen gelöscht. Alleine löscht man diese Datei mit *rm '1file1'*.

**Teil 3:** Laden Sie die Datei *dirStruct.tar.gz* aus Ilias und entpacken Sie die Datei. Löschen Sie mit *rm* alle entpackten Dateien und Verzeichnisse. Entpacken Sie die *dirStruct.tar.gz* erneut und wechseln Sie in das Verzeichnis *./rootTest/dir1*. Was passiert, wenn Sie alle Dateien mit der Endung *.txt* löschen wollen und als Befehl versehentlich *rm \* .txt* (Leerzeichen zwischen *\** und *.txt*) eingeben? Ist das Ergebnis rückgängig zu machen?

- Entpacken: *tar -xf dirStruct.tar.gz*
- Löschen: *rm -r rootTest* oder *rm -rf rootTest* (-f force-Flag)
- Entpacken: *tar -xf dirStruct.tar.gz*
- Wechseln: *cd rootTest/dir1*

Mit dem Befehl *rm \*.txt* sollen alle txt-Dateien gelöscht werden. Fügt man an den Stern (Metazeichen) nun eine Leertaste, werden aber stattdessen alle Dateien gelöscht, gefolgt von dem Hinweis, dass es keine Datei oder Ordner namens *.txt* gibt.

Das Ergebnis ist an sich nicht leicht rückgängig zu machen, da gelöschte Dateien per *rm* in keinen Trash abgelegt werden. Möchte man so eine Möglichkeit bietet sich das „trash-cli“ package an.