

# ITS1-Bericht:

## Übungen und Details

Tools:.....	1
Ls -a -> list all direcorys .....	1
Wget.....	1
head / tail:.....	1
Übungen stdin/stdout: .....	2
Übung 1:.....	2
Übung 2:.....	2
Übungen Tools: .....	3
Übung 1:.....	3
Übung 2:.....	3
Übung 3:.....	3
Übung 4:.....	4

### Tools:

Ls -a -> list all direcorys

Long option: -- (zb. --all)

Short option: - (zb. -a)

### Wget

wget ist ein Tool zum Downloaden von Dateien

### head / tail:

- head<Dateiname>: ersten 10 Zeilen einer Datei werden angezeigt
- head -n3 <Dateiname>: gibt man die ersten 3 Zeilen an.
- mit head -nx <Dteiname>gewünschte anzahl an zeichen werden angeben
- long option: head --lines=5 <Dateiname>
- tail <Dateiname>: letzten 10 Zeilen einer datei werden angezeigt

## Übungen stdin/stdout:

## Übung 1:

## Übung (Textdatei erstellen)

Mit Hilfe von mehreren `echo` Befehlen können einfache Textdateien erstellt werden.

Erzeuge nur unter Verwendung von `echo` Befehlen die Datei `made_by_echoing.txt` mit Inhalt:

```
=====
=   HTL BRAUNAU   =
=====
= 2AHITS Gruppe n =
= x Schülerinnen  =
= y Schüler       =
=====
```

```
~/wildcards$ echo ===== >made_by_echoing.txt
~/wildcards$ echo '=   HTL Braunau   =' >>made_by_echoing.txt
~/wildcards$ echo ===== >>made_by_echoing.txt
~/wildcards$ echo '= 2AHITS Gruppe 2 =' >>made_by_echoing.txt
~/wildcards$ echo '= 2Schülerinnen =' >>made_by_echoing.txt
~/wildcards$ echo '=   14Schüler   =' >>made_by_echoing.txt
~/wildcards$ echo ===== >>made_by_echoing.txt
~/wildcards$ cat made_by_echoing.txt
=====
=   HTL Braunau   =
=====
= 2AHITS Gruppe 2 =
= 2Schülerinnen  =
=   14Schüler   =
=====
```

## Übung 2:

## Übung (C Hallo Welt)

Erstelle mit Hilfe von `echo` Befehlen eine Datei `hello.cpp` mit folgendem Inhalt:

```
#include <iostream>

int main() {
    printf("\n\t*** Hallo Welt ***\n");
    return 0;
}
```

```
~/wildcards$ echo '#include <iostream>'>hello.cpp
~/wildcards$ echo 'int main() { '>>hello.cpp
~/wildcards$ echo ' printf("\n\t*** Hallo Welt ***\n");'>>hello.cpp
~/wildcards$ echo ' return 0;'>>hello.cpp
~/wildcards$ echo '}'>>hello.cpp
~/wildcards$ cat hello.cpp
#include <iostream>
int main() {
    printf("\n\t*** Hallo Welt ***\n");
    return 0;
}
```

## Übungen Tools:

### Übung 1:

#### Übung (tools)

- Was ist die longhand Option für `-1` bei `ls`?
- Was macht `-q` bei `ls` und was ist die dazugehörige longhand Option.
- Was passiert wenn bei `head` eine negative Zahl für die Anzahl der Zeilen verwendet wird? Was steht dazu in der manpage von `head`?
- Welche Option verhindert bei `echo` die Ausgabe eines Zeilenumbruchs?
- `wget` zeigt sehr viele Informationen zum Download-Vorgang, finde die Option die diese Ausgaben unterdrückt.
- Welche andere Art der Fortschrittsanzeige (außer dem Default) unterstützt `wget`? Welche Option ist dafür zu setzen?

- 
- Keine
- alle Dateien mit Anführungszeichen, `ls --quote-name`
- alle Zeilen außer die letzten 'n' Zeilen ausgegeben
- `Echo -n` keine zeilenumbrüche
- `wget -q`
- `wget -nv` gibt nur Fehler und wichtige Infos aus.

### Übung 2:

#### Übung (cat)

- Verwende `cat` um mit einem einzigen Kommandozeilenbefehl den Inhalt von drei Textdateien darzustellen.
- Verwende die beiden `cat` Optionen zum Nummerieren der Zeilen. Wie unterscheiden sich diese im Ergebnis?
- Was macht die Option `-E`? Was könnte der Sinn dieser Option sein?
- Experimentiere mit den Auswirkungen der `-s` Option

- ```
~/wildcards$ cat test.txt test2.txt test3.txt
```
- `cat <Dateiname1><Dateiname2><Dateiname3>2`
- `cat -n <Dateiname1><Dateiname2><Dateiname3>`.

Der einzige Unterschied sind die Zahlen

- `cat -e` gibt ein \$ am Ende jeder Zeile aus

Zum anzeigen von leerzeichen

- `cat -s` sorgt dafür, dass immer nur eine Zeile hintereinander leer ist
- `cat -s` : immer nur eine leere Zeile zwischen Texten angegeben.

### Übung 3:

### Übung (date)

Verwende Optionen von `date` damit die Ausgabe des Datums und der Uhrzeit in der Form `30.10.2009:15` erfolgt.

```
~/wildcards$ date '+%d.%m.%Y %R'
13.12.2022 14:08
```

### Übung 4:

#### Übung (date epoch)

Finde per Recherche heraus was die folgende Zeitdarstellung bedeutet:

```
$ date +%s
1639927239
```

- gibt die vergangene Zeit seit 1.1.1970 an