

Kommandozeileninterpreter

Was ist ein Kommandozeileninterpreter?

- Ein zeilenorientiertes Computerprogramm
 - Liest einzelne Zeilen (Kommandozeilen) als Befehle ein
 - Führt diese Befehle aus
 - Präsentiert das Ergebnis
- Auch Kommandointerpreter oder CLI (command-line interpreter) genannt
- Mensch-Maschine-Schnittstelle
 - Shell (Schale) um den Kernel (innerster Kern) eines Betriebssystems
- Ermöglicht Automatisierung durch Scripting
- Steuerung und Überwachung von Funktionen des OS

Geschichte

- Zuse Z3 verwendet Dezimaltastatur und Lampenfeld zur Ein- bzw. Ausgabe
- Fernschreiber
 - Elektrischer Schreibautomat zur Ausgabe auf Papier
 - Elektrische Tastatur zur Eingabe
 - Vom Computer getrennte Schnittstelle → ermöglicht Vernetzung
 - Ursprung des ASCII (zur Kommunikation zwischen Geräte verschiedener Hersteller)
- Glas-Fernschreiber
 - Elektrische Schreibautomaten / Drucker werden durch Röhrenbildschirme ersetzt
 - Erlaubt Cursorbewegung

Bash

- Bourne-Again Shell (Wieder eine Bourne Shell / Wiedergeborene Shell)
- Kommandointerpreter des GNU Betriebssystems
- Stammt direkt von der Bourne Shell ab
- Vereint Eigenschaften ihrer Vorgänger (z.B. ksh, csh)
 - Kompatibel zu einigen anderen Shells
- Standard-Shell in vielen Linux Distribution

Aufbau eines Befehls

\$ *<Befehlsnamen> <optionale Parameter>*

- Befehle können über ihren Namen oder Aliase aufgerufen werden
- Befehle können mit einem „&“ verkettet werden

Parameter

- Optionen (Flags)

\$ Befehl --verbose

\$ Befehl -v

\$ Befehl --input /Pfad/zur/Input-Datei

\$ Befehl -i /Pfad/zur/Input-Datei

- Argumente

\$ ls /tmp/

tmpfile

\$ rm /tmp/tmpfile

Präsentation & Ausarbeitung:

[github.com/john2ksonn/
Kommandozeileninterpreter](https://github.com/john2ksonn/Kommandozeileninterpreter)



Logische Verknüpfung von Befehlen

Logisches UND

\$ Befehl1 && Befehl2 Wenn Befehl1 erfolgreich ausgeführt wurde, wird Befehl2 ausgeführt

Logisches ODER

\$ Befehl1 || Befehl2 Wenn Befehl1 nicht erfolgreich ausgeführt wurde, wird Befehl2 ausgeführt

Datenstrom-Umleitung

\$ Befehl > Ausgabe.txt	Standardausgabe des Befehls in eine Datei umleiten
\$ Befehl >> Ausgabe.txt	Standardausgabe des Befehls einer Datei hinzufügen
\$ FehlerhafterBefehl 2> Ausgabe.txt	Standardfehlerausgabe des Befehls in eine Datei umleiten
\$ FehlerhafterBefehl 2>> Ausgabe.txt	Standardfehlerausgabe des Befehls einer Datei hinzurügen
\$ Befehl < Datei.txt	Den Inhalt der Datei als Standardeingabe in den Befehl einlesen

Piping

\$ Befehl1 Befehl2	Standardausgabe von Befehl1 als Standardeingabe von Befehl2 weiterleiten
\$ Befehl1 \$(Befehl2)	Standardausgabe von Befehl2 als Parameter für Befehl1

Scripting

```
#!/bin/bash

#iterating through all .log files
for file in $(ls *.log)
do
    #filtering all the SEVERE entries and writing them into a new file
    cat $file | grep SEVERE > severe/$file

    #print the number of SEVERE entries by counting the lines of a file
    echo "SEVERE entries count: $(wc -l severe/$file)"
done
```

Quellcode 1 Bash-Skript von der Demo

Hilfreiche Befehle

pwd	Gibt den Namen des aktuellen Verzeichnisses aus (<i>print working directory</i>)
ls	Listet den Inhalt eines Verzeichnisses auf
cd	Wechselt in das als Argument angegebene Verzeichnis. Sollte kein Argument angegeben sein, wird in das Home-Verzeichnis gewechselt. (<i>change directory</i>)
man	Stellt eine interaktive Anleitung zu einem als Argument angegebenen Befehl zur Verfügung. Viele Befehle verfügen über die <code>-help</code> -Flag und geben bei Verwendung dieser Flag ebenfalls eine kurze Erklärung aus. (<i>manuals</i>)
cat	Konkateniert (zusammenfügen) die als Argumente angegebenen Dateien und gibt das Ergebnis aus. (<i>concatenate</i>)
mkdir	Erstellt ein neues Verzeichnis mit dem als Argument angegebenen Namen (<i>make directories</i>)
echo	Gibt eine als Argument angegebene Textzeile aus. Kann mit Datenstromumleitung zum Bearbeiten von Dateien helfen.
touch	Aktualisiert den Zeitstempel einer Datei. Sollte die als Argument angegebene Datei nicht existieren, wird sie erstellt.
rm	Löscht die als Argument angegebene Datei. (<i>remove</i>)
rmdir	Löscht das als Argument angegebene Verzeichnis. Das Verzeichnis muss hierfür bereits leer sein. (<i>remove directory</i>)