# Zusammenfassung Tag 19

## Standardausgabe, Standardfehler

- **Programmausgabe in eine Datei schreiben**
  - über > können ausgaben eines Programm in eine Datei geschrieben werden
    - *date > ausgabe.txt*
      - erstellt die Datei ausgabe.txt im aktuellen Ordner
        und schreibt die Ausgabe von date dort hinein
      - Falls in der Datei schon etwas geschrieben steht wird dies überschrieben
  - über >> können ausgaben eines Programm an eine Datei angehangen werden.
    - *date >> ausgabe.txt*
      - die Ausgabe von date wird in die Datei ausgabe.txt geschrieben, ohne das der vorhandene Inhalt gelöscht wird
  - > oder >> (kurzschreibweiße) kann auch 1> oder 1>> geschrieben werden
- **Fehlerausgabe in eine Datei schreiben**
  - über 2> können Fehler in eine Datei geschrieben werden
    - *cat dateiexistiertnicht.txt 2> ausgabe.txt*
      - Der fehler das eine Datei nicht gefunden wurde wird in ausgabe.txt geschrieben
- **Programmausgabe und Fehlerausgabe können auch verknüpft werden**
  - *cat asdafasd.txt 1> programm-out.txt 2> programm-err.txt*
    - schreibt die ausgabe des Programms in programm-out.txt und falls ein Fehler auftritt wird dieser in programm-err.txt geschrieben
- **1> steht für den ersten ”Kanal“ die Programmausgabe**
- **2> steht für den zweiten ”Kanal“ die Fehlerausgabe**

## Stderr nach Stdout umleiten

- ***(date +“%Y“ && cat s.txt) 1> ausgabe.txt 2> ausgabe.txt***
  - schreibt das die Ausgabe des Programms in die Datei ausgabe.txt, welches aber überschrieben wird da ein Fehler auftritt.
  - Dies ließe sich wie folgt vermeiden:
    (wird aber so nicht genutzt da es bessere Möglichkeiten gibt)
  - *(date +“%Y“ && cat s.txt) 1>> ausgabe.txt 2>> ausgabe.txt*
- **mit 2>&1 kann die Fehlerausgabe in die Programmausgabe umgeleitet werden**
- ***(date +“%Y“ && cat s.txt) > ausgabe.txt 2>&1***
  - schreibt die Programmausgabe und den Fehler in die Datei ausgabe.txt

## Stderr nach Stdout umleiten (Teil 2)

- **Reihenfolge ist wichtig**
  - Richtig:
    *(date +"%Y" && cat s.txt) > ausgabe.txt 2>&1*
  - Falsch:
    *(date +"%Y" && cat s.txt) 2>&1 > ausgabe.txt*
- Ansonsten wird die Programmausgabe überschrieben und es wird nur der Fehler in ausgabe.txt geschrieben.

## Das Gerät dev null

- ***/dev/null***
  - für Linux ein gerät (device)
  - ausgabe eines Befehls wird verworfen
    - *echo "Test" > /dev/null*
- ***cat /dev/random***
  - gibt Zufallsdaten (Binär) aus
- ***cat /dev/urandom***
  - gibt unendlich pseudo Zufallsdaten(Binär) aus

## Der Exit-Code von Programmen

- ***$?***
  - Exit Variable
  - 0 Programm wurde korrekt ausgeführt
  - 1 (selten auch 2 oder 3) heißt es ist ein Fehler aufgetreten
  - Variable wird nach jedem Abgesetzten Befehl überschrieben
- ***echo $?***
  - gibt die Exit Variable aus
  - es kann überprüft werden ob der letzte Befehl erfolgreich ausgeführt wurde

## Standardeingabe

- ***sort***
  - es können Eingaben getätigt werden die dann sortiert werden
  - STRG + D beendet die Eingabe
- ***sort < name.txt***
  - schickt die Datei name.txt an das Programm sort
  - name.txt wird als Standardeingabe für sort verwendet
    sort bekommt name.txt nicht als Datei sondern als normale Eingabe
- ***sort < name.txt > name-sorted.txt***
  - gibt die Datei name.txt and sort weiter
    und schreibt die Ausgabe des Programms in name-sorted.txt
- **< ermöglicht Verkettungen von Befehlen mit Übergabe**

## Der Pipe-Operator

- **Der Pipe Operator |**
  - *ls | sort*
    gibt die Ausgabe von ls direkt an sort weiter
- ***ls | sort -r***
  - gibt die Ausgabe von ls an sort weiter
  - sort dreht durch -r die Sortierung um
- **Durch | kann man auch die Ausgabe von Programmen weitergeben**
  - < nimmt nur den Inhalt einer Datei
  - *sort < script.sh*
    - sricpt.sh wird als Datei betrachtet
  - *./scipt.sh | sort*
    - die ausgabe von sricpt.sh wird weitergegeben
- ***cat /proc/cpuinfo | grep "model name"***
  - die Ausgabe von cat wird an grep weitergegeben und nach model name durchsucht

## Das Programm tee

- ***ls | xargs echo***
  - xargs wandelt die Standard eingaben von ls in Parameter um damit echo diese verwenden kann
  - ls | echo würde nichts ausgeben
- ***ls /etc | tee output.txt | grep "cron"***
  - Ausgabe von ls wird an tee übergeben welches diese dann in eine Datei speichert diese Ausgabe wird dann weiter an grep übergeben
    welches diese nach cron durchsucht
- ***tee -a output.xt***
  - Ausgabe wird an die Datei an gehangen durch den Parameter -a

## Nützliche Befehle:

| | |
|---|---|
| *clear* | Bereinigt die Konsole |
| *strg+c* | Beendet ein Programm / unterbricht einen Befehl |
| *cat* | Erzeugt eine Ausgabe z.B. von einer Datei |
| *nano* | Einfacher Editor zum bearbeiten von Dateien |
| *commandname –help* | Öffnet meistens die Hilfe eines Programm |
| *man commandname* | Öffnet das Manual eines Programm falls vorhanden |
| *type commandname* | gibt aus worum es sich handelt (Befehl/Funktion/Programm) |
| *sort* | nimmt die Eingabe und sortiert diese |