# Arbeiten mit der Shell Teil 2

## Linux-Kurs der Unix-AG

Sebastian Weber

22. November 2013

# cat

- `cat` („concatenate“): gibt den Inhalt von Dateien aus
- Wichtige Optionen:
  - `-A`: nicht darstellbare Zeichen (Tabs, Zeilenumbrüche, etc.) werden durch Ersatzzeichen dargestellt
- Argumente: eine oder mehrere Dateien
  - Mehrere Dateien werden zusammenhängend ausgegeben

# less

- `less` ist ein Pager
- Inhalt einer Datei wird passend auf Bildschirm-/Terminalgröße ausgegeben
- Ermöglicht zeilenweises scrollen mit den Pfeiltasten und seitenweises Bild-Auf/-Ab bzw. `b`/Leertaste
- Pos1 und Ende springt zum Anfang bzw. zum Ende
- Suchen mit `/muster` (nach unten) oder `?muster` (nach oben)
- Weitersuchen mit `n` (gleiche Richtung) oder `N` (andere Richtung)
- Beenden mit `q`

# head

- `head`: gibt den Anfang einer Datei aus
- Wichtige Optionen:
  - `-n Zahl`: gibt soviele Zeilen aus wie angegeben, standardmäßig 10 Zeilen
- Argumente: eine oder mehrere Dateien

# tail

- `tail`: gibt das Ende einer Datei aus
- Wichtige Optionen:
  - `-n Zahl`: gibt soviele Zeilen aus wie angegeben, standardmäßig 10 Zeilen
  - `-f`: wartet darauf, dass neue Zeilen in die Datei geschrieben werden und gibt diese dann fortlaufend aus
  - Abbruch mit `STRG-C`
- Argumente: eine oder mehrere Dateien

# Ein-/Ausgabekanäle

- Programme verfügen über drei Standardkanäle
- Können sowohl lesbarer Text als auch codierte Daten sein
- Bei Shell-Befehlen normalerweise Text
- Kanäle können umgeleitet werden

| Nummer | Bezeichnung | Name | Gerät |
|---|---|---|---|
| 0 | Standard-Eingabe | `stdin` | Tastatur |
| 1 | Standard-Ausgabe | `stdout` | Konsole |
| 2 | Standard-Fehlerausgabe | `stderr` | Konsole |

# Pipes (1)

- Pipes („|“) leiten die Standard-Ausgabe (`stdout`) von einem Programm als Eingabe (`stdin`) in ein anderes Programm
- Wird typischerweise genutzt, um die Ausgabe eines Programms mit einem anderen weiterzuverarbeiten
- Grundprinzip: Einfache Befehle werden verknüpft um komplexere Aufgaben zu erledigen
- Es können (fast) beliebig viele Befehle mit Pipes verknüpft werden

# Pipes (2)

## Beispiel

- `ls -lhaR | less`
  - Zeigt alle Dateien, Verzeichnisse und deren Unterverzeichnisse mit Zusatzinformationen an
  - Die Ausgabe kann mit `less` seitenweise betrachtet und durchsucht werden
- `tail -n 42 /var/log/syslog | head -n 12 | less`
  - Zeigt von den letzten 42 Zeilen der Datei „`/var/log/syslog`“ nur die ersten 12 an
  - Die Ausgabe kann mit `less` betrachtet werden

# Ein-/Ausgabe in Dateien (1)

- Die Kanäle können in Dateien umgeleitet werden
- Dazu dienen die Operatoren „<“ und „>“ bzw. „>>“
- „<“ liest die Eingabe `stdin` aus einer Datei
- „>“ überschreibt den Inhalt einer Datei mit der Ausgabe `stdout`
- „>>“ hängt die Ausgabe `stdout` an eine Datei an

# Ein-/Ausgabe in Dateien (2)

## Beispiele

- `ls -a > liste`
  - Die Datei `liste` enthält Auflistung aller Dateien und Verzeichnisse des aktuellen Verzeichnisses
  - Wenn die Datei bereits existiert, so wird diese überschrieben
- `ls -a / >> bar`
  - Die Datei `bar` enthält Auflistung aller Dateien und Verzeichnisse der Verzeichnisse `/` und `/etc`
  - Neue Inhalte werden an das Ende der Datei angehangen
- `cat < foo`
  - Das Programm `cat` erhält als Eingabe die Datei `foo`
  - Macht das selbe wie `cat foo`
  - Der Operator „<“ wird eher selten benutzt

# Kanäle umleiten (1)

- Die Kanäle können mit dem Operator „`>&`“ aufeinander umgeleitet werden
- Um Kanal 2 (`stderr`) auf Kanal 1 (`stdout`) umzuleiten, wird „`2>&1`“ an das Ende der Befehlszeile hinzugefügt
- „`2>&1`“ wird verwendet, um die Fehlerausgabe von Programmen mit Pipes oder in Dateien weiterzuleiten

# Kanäle umleiten (2)

## Beispiele

- `ls -l /bin /root /tmp 2>&1 > list`
  - Die Standard-Fehlerausgabe wird auf die Konsole umgeleitet
  - Die Standard-Ausgabe wird auf die Datei „`list`“ umgeleitet
  - Die Fehlermeldungen erscheinen trotzdem auf der Konsolenausgabe und nicht in der Datei
- `ls -l /bin /root /tmp > list 2>&1`
  - Die Standard-Ausgabe wird auf die Datei „`list`“ umgeleitet
  - Die Standard-Fehlerausgabe wird auf die Standard-Ausgabe umgeleitet
  - Die Fehlermeldungen erscheinen in der Datei

# sort (1)

- `sort`: sortiert Text abhängig von der Spracheinstellung des Systems
- Wichtige Optionen:
  - `-d` („dictionary“): ignoriert alles außer Buchstaben, Ziffern und Leerzeichen
  - `-i` („ignore“): ignoriert alle nicht druckbaren Zeichen
  - `-b` („blank“): ignoriert führende Leerzeichen
  - `-u` („unique“): vereint mehrfach vorkommende Zeilen

# sort (2)

- Wichtige Optionen:
  - `-n` („numeric“): sortiert nach dem Zahlenwert (nicht zusammen mit `-h`)
  - `-h` („human-numeric“): sortiert nach menschenlesbaren Zahlen, z.B. bei Dateigrößen (123 < 22.2K < 2M < 3.14G)
  - `-k N` („key“): sortiert nach der `N`-ten Spalte
  - `-r` („reverse“): dreht die Sortierreihenfolge um
  - `-t<TRENN>` („seperator“): `<TRENN>` gibt das Trennzeichen zwischen Spalten an
- Argumente: eine oder mehrere Dateien
  - Bei mehreren Dateien werden diese zusammen sortiert

# sort (3)

## Beispiele

- `sort -k 3 -t: -n /etc/passwd`
  - Sortiert die Benutzerlogins nach ihren UIDs aufsteigend
- `dmesg | sort -b -k 3`
  - Sortiert die Systemmeldungen alphabetisch aufsteigend
- `ls -la | sort -h -k 5 -r`
  - Sortiert die Dateien und Verzeichnisse im aktuellen der Größe nach absteigend

# cut (1)

- `cut`: Schneidet Spalten aus Texten herauszuschneiden
- Wichtige Optionen:
  - `-c <LISTE>` („character“): schneidet die in `LISTE` angegebenen Zeichen heraus
  - `-f <LISTE>` („field“): schneidet die in `LISTE` angegebenen Felder heraus
  - `-d <TRENN>` („delimiter“): `TRENN` gibt das Trennzeichen zwischen den Spalten an
- `LISTE`: kann eine Zahl oder ein Zahlenbereich sein
- Argumente: eine oder mehrere Dateien

# cut (2)

## Beispiele

- `cut -d : -f 5 /etc/passwd`
  - Gibt die langen Namen aller Benutzer des Systems aus
- `ls -lh | cut -c 55-`
  - Gibt alles ab dem 55. Zeichen von `ls -lh` aus

# wc (1)

- `wc` („word count“) zählt Zeilen, Worte und Bytes
- Anzahl der Bytes entspricht in etwa der der Zeichen
- Wichtige Optionen:
  - `-m` („characters“): zählt nur Zeichen
  - `-l` („lines“): zählt nur Zeilen
  - `-w` („words“): zählt nur Worte
- Argumente: eine oder mehrere Dateien

# wc (2)

## Beispiele

- `ls -A | wc -l`
  - Gibt die Anzahl der Dateien und Verzeichnisse im aktuellen Verzeichnis an
- `wc -w /usr/share/common-licenses/GPL-3`
  - Gibt die Anzahl der Wörter der GPLv3-Lizenz aus

# Spickzettel

## Alle Befehle

| Befehl | Optionen |
|---|---|
| `cat` | `-A` |
| `less` | (`/muster`, `?muster`, `n`, `N`, `q`) |
| `head` | `-n` |
| `tail` | `-n`, `-f` |
| `sort` | `-d`, `-i`, `-b`, `-u`, `-n`, `-k`, `-r`, `-t` |
| `cut` | `-c`, `-f`, `-d` |
| `wc` | `-m`, `-l`, `-w` |