

Arbeiten mit der Shell Teil 2

Linux-Kurs der Unix-AG

Benjamin Eberle

12. Mai 2015



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 mit Bild-Auf/-Ab oder 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

Pipes

- ▶ Unix-Philosophie: Viele kleine Programme, die eine Aufgabe gut erfüllen
- ▶ Komplexere Aufgaben durch Kombination mehrerer Programme
- ▶ Pipes (engl. für Rohre) ermöglichen Umleitung der Ausgabe eines Programms in ein anderes Programm
- ▶ „Pipelines“

Pipes – Beispiele

- ▶ `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)

- ▶ Ausgabe von Befehlen kann in Dateien umgeleitet werden
- ▶ Eingabe für Befehle kann aus Dateien gelesen werden
- ▶ „Trichter“: „<“ und „>“ bzw. „>>“
- ▶ „<“ liest die Eingabe aus einer Datei
- ▶ „>“ überschreibt den Inhalt einer Datei mit der Ausgabe
- ▶ „>>“ hängt die Ausgabe an eine Datei an

Ein-/Ausgabe in Dateien (2)

Beispiele

- ▶ `ls -a > liste`
 - ▶ Datei `liste` enthält Auflistung des Inhalts des aktuellen Verzeichnisses
 - ▶ Wenn die Datei bereits existiert, wird sie überschrieben
- ▶ `ls -a / >> bar`
 - ▶ Datei `bar` enthält Auflistung des Inhalts des Verzeichnisses `/`
 - ▶ Auflistung wird an das Ende der Datei angehängt
- ▶ `cat < foo`
 - ▶ `cat` erhält als Eingabe die Datei `foo`
 - ▶ Macht das selbe wie `cat foo`
 - ▶ „`<`“ wird eher selten benutzt

Standard-Kanäle

- ▶ Verschiedene Kanäle für Ein- und Ausgabe
- ▶ Standardausgabe (stdout): Normale Ausgabe
- ▶ Standardfehlerausgabe (stderr): Fehlermeldungen
- ▶ Standardeingabe (stdin): Eingabe
- ▶ Kein Unterschied zwischen stdout und stderr bei Ausgabe in Terminals
- ▶ Normale Ausgabe und Fehlermeldungen können getrennt werden
- ▶ Nützlich bei Programmen mit sehr viel Ausgabe
- ▶ `programm > /dev/null`: Leitet normale Ausgabe nach `/dev/null` um (wird verworfen), nur Fehler werden ausgegeben

Standard-Kanäle – Umleitungen

- ▶ „Normale“ Umleitungen (>, >>, |) leiten nur Standardausgabe um
- ▶ Fehlerausgabe in Datei umleiten: 2>
- ▶ Fehlerausgabe in Standardausgabe umleiten: 2>&1
- ▶ Notwendig, um Fehlerausgabe in Pipelines umzuleiten:
2>&1 |

Umleitungen – Reihenfolge

- ▶ Standard- und Fehlerausgabe in Datei umleiten: Erst Standardausgabe in Datei, dann Fehlerausgabe in Standardausgabe
- ▶ Beispiel: `ls /etc/ /root > liste 2>&1`
- ▶ `ls /etc/ /root 2>&1 > liste` gibt Fehler im Terminal aus
- ▶ Fehlerausgabe in Pipe: Erst Fehlerausgabe in Standardausgabe umleiten, dann Pipe
- ▶ Beispiel:
`cat /etc/shadow /etc/passwd 2>&1 | grep root`
- ▶ Ausgabedateien werden vor Ausführen des Programms geleert (außer bei `>>`):
- ▶ `befehl < datei > datei`: Datei ist danach leer

sort

- ▶ sort: sortiert Text abhängig von der Spracheinstellung des Systems
- ▶ Wichtige Optionen:
 - ▶ -u („unique“): vereint mehrfach vorkommende Zeilen
 - ▶ -r („reverse“): dreht die Sortierreihenfolge um
- ▶ Argumente: eine oder mehrere Dateien
 - ▶ Bei mehreren Dateien werden diese zusammen sortiert

cut (1)

- ▶ cut: Schneidet Spalten aus Texten heraus
- ▶ 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 | Funktion |
|--------|--------------------|-----------------------------------|
| cat | -A | Dateiinhalte ausgeben |
| less | (/muster, n, N, q) | Dateiinhalte seitenweise anzeigen |
| head | -n | Anfang einer Datei ausgeben |
| tail | -n, -f | Ende einer Datei ausgeben |
| sort | -u, -r | Text sortieren |
| cut | -c, -f, -d | Spalten aus Text ausschneiden |
| wc | -m, -l, -w | Zählt Wörter od. Zeilen |