

Arbeiten mit der Shell Teil 2

Linux-Kurs der Unix-AG

Zinching Dang

22. November 2015

UNIX
AG

TU Kaiserslautern

RH Regionales
Hochschul-
Rechenzentrum
Kaiserslautern RK



Übersicht

Ausgaben umleiten

Wiederholung & Vertiefung

Shellbefehle

Befehle & Optionen

Zusammenfassung & Ausblick

```
linux-kurs@linuxkurs:~$ fortune -s | cowsay
```

```
/ You will be attacked by a beast who has \  
| the body of a wolf, the tail of a lion, | \  
\ and the face of Donald Duck. /
```

```
-----  
 \      ^ ^  
  (oo)\_____)\/\  
  (__)\/       )\/\  
      ||----w |  
      ||     ||
```

```
linux-kurs@linuxkurs:~$ █
```

Beispielanwendung zu Pipes

Programm-Ausgaben in Dateien umleiten

Datei-Umleitungen

- ▶ „ `<` “ liest die Eingabe für ein Programm aus einer Datei
- ▶ „ `>` “ leitet die Ausgabe eines Programms in eine Datei
- ▶ „ `>>` “ hängt die Ausgabe eines Programms an eine Datei an

Programm-Ausgaben in Dateien umleiten

Datei-Umleitungen

- ▶ „ `<` “ liest die Eingabe für ein Programm aus einer Datei
- ▶ „ `>` “ leitet die Ausgabe eines Programms in eine Datei
- ▶ „ `>>` “ hängt die Ausgabe eines Programms an eine Datei an

Funktionsweise

- ▶ `Programm > Datei` (wird überschrieben falls vorhanden)
- ▶ `Programm >> Datei` (Inhalt wird angehängen)

Programme miteinander verknüpfen

„Pipes“ (vertikaler Balken: |)

- ▶ leiten die Ausgabe eines Programms in ein anderes Programm weiter
- ▶ lösen komplexe Aufgaben durch geschickte Kombination mehrerer Programme

Programme miteinander verknüpfen

„Pipes“ (vertikaler Balken: |)

- ▶ leiten die Ausgabe eines Programms in ein anderes Programm weiter
- ▶ lösen komplexe Aufgaben durch geschickte Kombination mehrerer Programme

Funktionsweise

- ▶ `Programm1 | Programm2`

Standardkanäle

Programm-Ein- und -Ausgaben aufgeteilt in

- ▶ Standardausgabe („ `STDOUT` “) für normale Ausgabe
- ▶ Standardfehlerausgabe („ `STDERR` “) für Fehlermeldungen
- ▶ Standardeingabe („ `STDIN` “) für Eingaben

Standardkanäle

Programm-Ein- und -Ausgaben aufgeteilt in

- ▶ Standardausgabe („ `STDOUT` “) für normale Ausgabe
- ▶ Standardfehlerausgabe („ `STDERR` “) für Fehlermeldungen
- ▶ Standardeingabe („ `STDIN` “) für Eingaben

Standardausgabe & Standardfehlerausgabe

- ▶ werden standardmäßig zusammen ausgegeben
- ▶ können getrennt voneinander verarbeitet werden

Umleitung der Standardkanäle

Umleitungsoperatoren

- ▶ `>`, `>>` und `|` leiten nur `STDOUT` um
 - ▶ Fehlermeldungen werden noch auf der Shell ausgegeben

Umleitung der Standardkanäle

Umleitungsoperatoren

- ▶ `>`, `>>` und `|` leiten nur `STDOUT` um
 - ▶ Fehlermeldungen werden noch auf der Shell ausgegeben
- ▶ `2>` leitet `STDERR` in eine Datei um
 - ▶ `ls . /root 2> foo` leitet nur Fehlermeldung um
 - ▶ `ls . /root > foo 2> bar` leitet Fehlermeldung und Ausgabe separat um

Umleitung der Standardkanäle

Umleitungsoperatoren

- ▶ `>`, `>>` und `|` leiten nur `STDOUT` um
 - ▶ Fehlermeldungen werden noch auf der Shell ausgegeben
- ▶ `2>` leitet `STDERR` in eine Datei um
 - ▶ `ls . /root 2> foo` leitet nur Fehlermeldung um
 - ▶ `ls . /root > foo 2> bar` leitet Fehlermeldung und Ausgabe separat um
- ▶ `2>&1` leitet `STDERR` in `STDOUT` um
 - ▶ Umleitung muss am Ende stehen
 - ▶ `ls . /root > foo 2>&1` leitet beide Kanäle in eine Datei um

Wiederholung & Vertiefung: Shellbefehle

Bereits kenengelernte Befehle

- ▶ `cp` – „**copy**“: Dateien und Verzeichnisse kopieren
- ▶ `mv` – „**move**“: Dateien und Verzeichnisse verschieben
- ▶ `rm` – „**remove**“: dateien löschen
- ▶ `man` – „**man**ual“: Handbuch zu Befehlen

Besonderheiten und Eigenheiten

Verzeichnisse mit Inhalt kopieren

- ▶ `cp -r` – die Option ist dazu zwingend notwendig

Besonderheiten und Eigenheiten

Verzeichnisse mit Inhalt kopieren

- ▶ `cp -r` – die Option ist dazu zwingend notwendig

Dateien oder Verzeichnisse umbenennen

- ▶ `mv alt neu` – Umbenennung entspricht dem Verschieben

Besonderheiten und Eigenheiten

Verzeichnisse mit Inhalt kopieren

- ▶ `cp -r` – die Option ist dazu zwingend notwendig

Dateien oder Verzeichnisse umbenennen

- ▶ `mv alt neu` – Umbenennung entspricht dem Verschieben

Verzeichnisse mit Inhalt löschen

- ▶ `rm -r` – funktioniert analog zu `cp -r`

Übersicht: Shellbefehle

Neue Befehle

- ▶ `less` : Dateiinhalt seitenweise darstellen
- ▶ `cat` – „con**cat**enate“: Dateiinhalt ausgeben
- ▶ `head` : Anfang einer Datei ausgeben
- ▶ `tail` : Ende einer Datei ausgeben
- ▶ `sort` : Dateiinhalt sortiert ausgeben
- ▶ `cut` : Spalten einer Datei ausgeben
- ▶ `wc` – „**w**ord **c**ount“: Wörter einer Datei zählen

Dateiinhalte seitenweise darstellen – less

Funktionsweise

- ▶ Datei mit `less meineDatei` öffnen¹
- ▶ Dateiinhalt wird passend zur Terminalgröße angezeigt
- ▶ zeilenweises Scrollen mit Pfeiltasten
- ▶ seitenweises Scrollen mit Bild-Auf/Bild-Ab
- ▶ `</muster>` sucht vorwärts, `<?muster>` rückwärts
 - ▶ `<n>` zeigt den nächsten Treffer an, `<N>` den vorherigen
- ▶ beenden mit `<q>`

¹Man-Pages werden auch mit less dargestellt

Dateiinhalte ausgeben – cat

Allgemeines

- ▶ gibt den gesamten Dateiinhalt auf die Shell aus
- ▶ Argumente: eine oder mehrere Dateien
 - ▶ mehrere Dateien können zusammenhängend ausgegeben werden

Dateiinhalte ausgeben – cat

Allgemeines

- ▶ gibt den gesamten Dateiinhalt auf die Shell aus
- ▶ Argumente: eine oder mehrere Dateien
 - ▶ mehrere Dateien können zusammenhängend ausgegeben werden

Wichtige Optionen

- ▶ `-A` : zeigt nicht-darstellbare Zeichen (Tabs, Zeilenumbrüche, etc.) mit Hilfe von Ersatzzeichen an

Anfang einer Datei ausgeben – head

Allgemeines

- ▶ gibt den Anfang einer Datei auf die Shell aus
- ▶ standardmäßig werden die ersten 10 Zeilen ausgegeben
- ▶ Argumente: eine oder mehrere Dateien

Anfang einer Datei ausgeben – head

Allgemeines

- ▶ gibt den Anfang einer Datei auf die Shell aus
- ▶ standardmäßig werden die ersten 10 Zeilen ausgegeben
- ▶ Argumente: eine oder mehrere Dateien

Wichtige Optionen

- ▶ `-n <X>` : die ersten <X> Zeilen ausgeben
- ▶ `-n -<Y>` : alles, bis auf die letzten <Y> Zeilen ausgeben

Ende einer Datei ausgeben – tail

Allgemeines

- ▶ gibt das Ende einer Datei auf die Shell aus
- ▶ standardmäßig werden die letzten 10 Zeilen ausgegeben
- ▶ Argumente: eine oder mehrere Dateien

Ende einer Datei ausgeben – tail

Allgemeines

- ▶ gibt das Ende einer Datei auf die Shell aus
- ▶ standardmäßig werden die letzten 10 Zeilen ausgegeben
- ▶ Argumente: eine oder mehrere Dateien

Wichtige Optionen

- ▶ `-n <X>` : die letzten `<X>` Zeilen ausgeben
- ▶ `-f` : fortlaufend Änderungen ausgeben
 - ▶ Abbruch mit `<STRG>+<C>`

Dateiinhalte sortiert ausgeben – sort

Allgemeines

- ▶ gibt den sortierten Inhalt einer Datei auf die Shell aus
- ▶ Sortierung ist in der Regel alphabetisch
- ▶ Argumente: eine oder mehrere Dateien

Dateiinhalte sortiert ausgeben – sort

Allgemeines

- ▶ gibt den sortierten Inhalt einer Datei auf die Shell aus
- ▶ Sortierung ist in der Regel alphabetisch
- ▶ Argumente: eine oder mehrere Dateien

Wichtige Optionen

- ▶ `-r` – „**r**reverse“: Sortierung umdrehen
- ▶ `-u` – „**u**nique“: doppelte Einträge ausblenden

Spalten einer Datei ausgeben – cut

Allgemeines

- ▶ gibt Spalten aus einer Datei auf die Shell aus
- ▶ Argumente: eine oder mehrere Dateien

Spalten einer Datei ausgeben – cut

Allgemeines

- ▶ gibt Spalten aus einer Datei auf die Shell aus
- ▶ Argumente: eine oder mehrere Dateien

Wichtige Optionen

- ▶ `-c <Liste>` – „**c**haracter“: schneidet Zeichen aus <Liste> aus
- ▶ `-f <Liste>` – „**f**ield“: schneidet die angegebenen Felder aus
- ▶ `-d <Trenner>` – „**d**elimiter“: Trennsymbol zwischen Spalten

Wörter einer Datei zählen – `wc`

Allgemeines

- ▶ zählt Zeilen, Wörter und Bytes einer Datei
- ▶ Argumente: eine oder mehrere Dateien

Wörter einer Datei zählen – `wc`

Allgemeines

- ▶ zählt Zeilen, Wörter und Bytes einer Datei
- ▶ Argumente: eine oder mehrere Dateien

Wichtige Optionen

- ▶ `-m` : zählt die tatsächlichen Zeichen
- ▶ `-c` : zählt nur Bytes
- ▶ `-l` : zählt nur Zeilen
- ▶ `-w` : zählt nur Wörter

Wichtige Befehle & Optionen

Befehl	Optionen	Funktion
<code>less</code>		Datei seitenweise darstellen
<code>cat</code>	<code>-A</code>	Datei ausgeben
<code>head</code>	<code>-n</code>	Dateianfang ausgeben
<code>tail</code>	<code>-n</code> , <code>-f</code>	Dateiende ausgeben
<code>sort</code>	<code>-r</code> , <code>-u</code>	Datei sortieren
<code>cut</code>	<code>-c</code> , <code>-f</code> , <code>-d</code>	Spalten einer Datei ausgeben
<code>wc</code>	<code>-m</code> , <code>-c</code> , <code>-l</code> , <code>-w</code>	Wörter einer Datei zählen

Beispiele zu Pipes und Umleitungen

Pipes

- ▶ Zeilen 24 bis 42 einer Datei ausgeben (insgesamt 19 Zeilen)
 - ▶ `head -n 42 <datei> | tail -n 19`
- ▶ Datei sortieren und Zeichen 3 bis 7 ausgeben
 - ▶ `sort <datei> | cut -c 3-7`
- ▶ Zeichen 3 bis 7 einer Datei umgedreht sortiert ausgeben
 - ▶ `cut -c 3-7 <datei> | sort -r`
- ▶ einmalige Zeilen einer Datei zählen
 - ▶ `sort -u <datei> | wc -l`

Beispiele zu Pipes und Umleitungen

Umleitungen

- ▶ Ausgabe in eine Datei schreiben
 - ▶ `ls Dokumente > dateiliste`
- ▶ Ausgabe und Fehlermeldungen in separate Dateien schreiben
 - ▶ `ls Dokumente /root/ > dateiliste 2> fehler`
- ▶ Ausgabe und Fehlermeldungen in eine Datei schreiben
 - ▶ `ls Dokumente /root/ > dateilisteMitFehlern 2>&1`

Beispiele zu Pipes und Umleitungen

Pipes und Umleitungen gemeinsam

- ▶ Fehlermeldung in eine Datei schreiben, Ausgabe anzeigen
 - ▶ `ls Dokumente /root/ 2> fehler | less`
- ▶ Ausgabe und Fehlermeldungen anzeigen
 - ▶ `ls Dokumente /root/ 2>&1 | less`

Zusammenfassung & Ausblick

Zusammenfassung

- ▶ Umleitungen und Ausgabekanäle von Programmen
- ▶ Shellbefehle zur Verarbeitung von Dateiinhalten

Zusammenfassung & Ausblick

Zusammenfassung

- ▶ Umleitungen und Ausgabekanäle von Programmen
- ▶ Shellbefehle zur Verarbeitung von Dateiinhalten

Nächstes Mal

- ▶ weitere Shellbefehle
- ▶ einfache Suchmuster („Globs“)
- ▶ archivieren und komprimieren