Arbeiten mit der Shell Teil 3

Linux-Kurs der Unix-AG

Zinching Dang

29. November 2016

UNIX AG
TU Kaiserslautern

RH Regionales Hochschul-
Rechenzentrum Kaiserslautern RK

CC BY SA

Übersicht

Wiederholung & Vertiefung

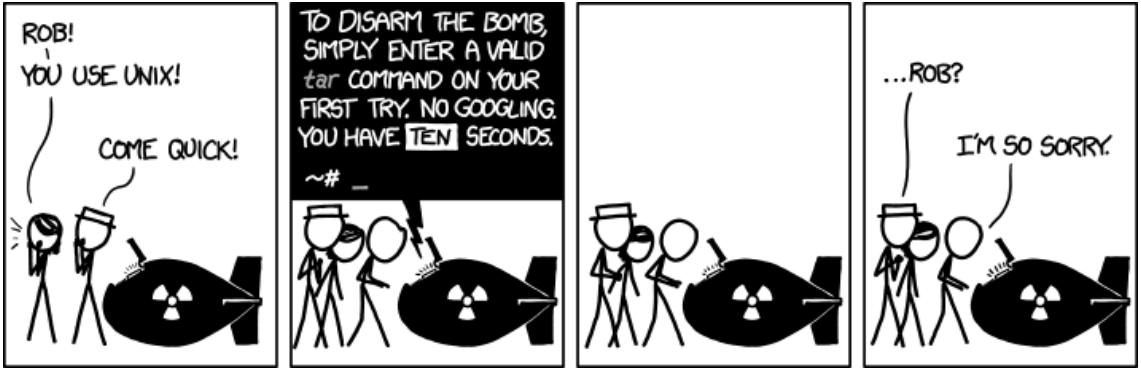
Globs

Archivieren & Komprimieren

Befehle & Optionen

Zusammenfassung & Ausblick

Übersicht



I don't know what's worse – the fact that after 15 years of using tar I still can't keep the flags straight, or that after 15 years of technological advancement I'm still mucking with tar flags that were 15 years old when I started. (`https://xkcd.com/1168`)

Wiederholung & Vertiefung: Shellbefehle & Pipes

Optionen zusammenfassen und Optionsreihenfolge

- `cut -d , -f 1-3,4-5`
  - kein Zusammenfassen möglich, da beide Optionen eigene Argumente benötigen (`,` für `-d` bzw. `1-3,4-5` für `-f`)
- `tail -fn 3`
  - Zusammenfassen nur unter Beachtung der Reihenfolge möglich
  - Option `-f` hat kein zusätzliches Argument, daher zuerst
  - `-nf 3` oder `-n3f` sind ungültig

Wiederholung & Vertiefung: Shellbefehle & Pipes

Programmausgaben umleiten und weiterverarbeiten

- ```
  head -n 3 tab1 | tail -n 1 | cut -d , -f 2-4,6-
  ```
- ```
  #tab1: Stadt Land Fluss
  az,s,l,f,n,t,b,p
  A-Z,Stadt,Land,Fluss,Name,Tier,Beruf
  C,Celle,China,Colorado,Chris,Chamälion,Chemiker
  ```

Wiederholung & Vertiefung: Shellbefehle & Pipes

Programmausgaben umleiten und weiterverarbeiten

- ```
  head -n 3 tab1 | tail -n 1 | cut -d , -f 2-4,6-
  ```
- ```
  #tab1: Stadt Land Fluss
  az,s,l,f,n,t,b,p
  A-Z,Stadt,Land,Fluss,Name,Tier,Beruf
  C,Celle,China,Colorado,Chris,Chamälion,Chemiker
  ```

  head

Wiederholung & Vertiefung: Shellbefehle & Pipes

Programmausgaben umleiten und weiterverarbeiten

- `head -n 3 tab1 | tail -n 1 | cut -d , -f 2-4,6-`
-
```
#tab1: Stadt Land Fluss
az,s,l,f,n,t,b,p
A-Z,Stadt,Land,Fluss,Name,Tier,Beruf
C,Celle,China,Colorado,Chris,Chamälion,Chemiker
```
  tail

Wiederholung & Vertiefung: Shellbefehle & Pipes

Programmausgaben umleiten und weiterverarbeiten

- `head -n 3 tab1 | tail -n 1 | cut -d , -f 2-4,6-`
-
```
#tab1: Stadt Land Fluss
az,s,l,f,n,t,b,p
A-Z,Stadt,Land,Fluss,Name,Tier,Beruf
C,Celle,China,Colorado,Chris,Chamälion,Chemiker
```
  cut

Wiederholung & Vertiefung: Shellbefehle & Pipes

Programmausgaben in Dateien umleiten

- `sort -r /etc/passwd /etc/shadow > liste 2> fehler`
- Programmausgabe `STDOUT` wird in „ `liste` “ umgeleitet
- Fehlerausgabe `STDERR` wird in „ `fehler` “ umgeleitet
- keine Ausgabe auf der Shell

Wiederholung & Vertiefung: Shellbefehle & Pipes

Besonderheiten

- standardmäßig wird bei Pipes nur `STDOUT` umgeleitet
- „ `2>&1` “ leitet `STDERR` nach `STDOUT` um
- „ `>>` “ hängt die Ausgabe an eine Datei an

Globs

Allgemeines

- spezielle Zeichen mit besonderer Bedeutung in der Shell
- Platzhalter für Zeichen
- ermöglichen einfache Suchmuster

Globs

Allgemeines

- spezielle Zeichen mit besonderer Bedeutung in der Shell
- Platzhalter für Zeichen
- ermöglichen einfache Suchmuster

Anwendungszwecke

- verarbeiten mehrerer Dateien mit ähnlichem Dateinamen
- Wörter mit Mehrfachschreibweise, z. B. Potenzial/Potential

Globs

Platzhalter

- `?` : einzelnes beliebiges Zeichen außer `.` am Anfang
  - `a?c` passt z. B. auf: `aac`, `acc`, `aBc`, `a1c`, `a.c`, `a?c`
- `*` : beliebig viele Zeichen (auch keins) außer `.` am Anfang
  - `a*c` passt z. B. auf: `ac`, `aac`, `aBac`, `a1aa1ac`, `a..a.?c`
- `[]` : anzugebende Zeichenliste
  - `a[abc]c` passt auf: `aac`, `abc`, `acc`, aber nicht auf: `aAc`, `aaac`
  - `a[ab][cd]c` passt auf: `aacc`, `aadc`, `abcc`, `abdc`, aber nicht auf: `acdc`, `aabc`

Globs

Weitere Beispiele

- `a[a-zA-Z][0-9a-z]`
  - `aA2`, `aa2`, `aWc`, `abc`
- `Ha*llo`
  - `Hallo`, `Hallihallo`, `Ha2xqf4p3bsltgllo`
- `*ll*`
  - `Hallo`, `Halle`, `Fall`, `ll`

Archivieren & Komprimieren

Archivieren

- mehrere Dateien und Verzeichnisse zusammenfassen
- historisch bedingt durch das Speichern auf Magnetbändern
- heutzutage als Dateidownload/E-Mail Anhang üblich

Archivieren & Komprimieren

Archivieren

- mehrere Dateien und Verzeichnisse zusammenfassen
- historisch bedingt durch das Speichern auf Magnetbändern
- heutzutage als Dateidownload/E-Mail Anhang üblich

Komprimieren

- Speicherbedarf von Dateien reduzieren
- ausgeklügelte Algorithmen erlauben teils hohe Einsparung

Archivieren & Komprimieren

Befehle

- `tar` zum Archivieren
- `gzip` oder `bzip2` zum Komprimieren

Archivieren & Komprimieren

Befehle

- `tar` zum Archivieren
- `gzip` oder `bzip2` zum Komprimieren

Allgemeines

- häufig zusammen als komprimierte Archive verwendet
- unter Linux/Unix `tar` mit `gzip` / `bzip2` üblich
  - `.tar.gz` / `.tgz` bzw. `.tar.bz2` -Dateien

Übersicht: Shellbefehle

Neue Befehle

- `tar` – „**t**ape **ar**chive“: Archive erstellen und entpacken
- `gzip` – „**G**NU **zip**“ und `bzip2` :
  - Programme um Dateien zu komprimieren und dekomprimieren

Dateien und Verzeichnisse archivieren – `tar`

Allgemeines

- fasst mehrere Dateien und Verzeichnisse zusammen
- Argumente: eine oder mehrere Datein oder Verzeichnisse

Dateien und Verzeichnisse archivieren – `tar`

Allgemeines

- fasst mehrere Dateien und Verzeichnisse zusammen
- Argumente: eine oder mehrere Datein oder Verzeichnisse

Wichtige Optionen

- `-f a.tar` – „**f**ile“: liest/schreibt in die Datei „ `a.tar` “
- `-c` – „**c**reate“: erstellt ein Archiv
- `-x` – „e**x**tract“: entpackt ein Archiv
- `-t` – „**t**able of contents“: listet den Inhalt eines Archivs auf

Dateien und Verzeichnisse archivieren – `tar`

Beispiele

- `tar -cf backup.tar Bilder/ Dokumente/`
  - Archiv „ `backup.tar` “ wird im akuellen Verzeichnis angelegt
  - beinhaltet die Verzeichnisse `Bilder/` und `Dokumente/`
- `tar -tf backup.tar`
  - zeigt den Inhalt des Archivs an
- `tar -xf backup.tar`
  - entpackt das Archiv in das aktuelle Verzeichnis

Dateien (de)komprimieren – `gzip` & `bzip2`

Allgemeines

- häufig zusammen mit `tar` verwendet
- `bzip2` komprimiert besser, aber langsamer als `gzip`
- Argument: eine oder mehrere Dateien
  - Originaldatei wird durch (de)komprimierte ersetzt

Dateien (de)komprimieren – `gzip` & `bzip2`

Allgemeines

- häufig zusammen mit `tar` verwendet
- `bzip2` komprimiert besser, aber langsamer als `gzip`
- Argument: eine oder mehrere Dateien
  - Originaldatei wird durch (de)komprimierte ersetzt

Wichtige Optionen

- `-d` – „**d**ecompress“: dekomprimieren
  - `gunzip` und `bunzip2` sind äquivatente Befehle zum Dekomprimieren

Dateien komprimieren – `gzip` & `bzip2`

Beispiele

- `gzip backup.tar`
  - komprimiert „ `backup.tar` “ zu „ `backup.tar.gz` “
- `bzip2 backup.tar`
  - komprimiert „ `backup.tar` “ zu „ `backup.tar.bz2` “
- `bunzip2 backup.tar.bz2` oder `gzip -d backup.tar.gz`
  - dekomprimiert das jeweilige Archiv

`gzip` & `bzip2` mit `tar` verknüpfen

Allgemeines

- Verknüpfung mit Hilfe von Pipes möglich, jedoch aufwendig
- `tar` hat Optionen zum Komprimieren eingebaut
  - `-z` für um `gzip` zum Komprimieren zu verwenden
  - `-j` für um `bzip2` zum Komprimieren zu verwenden
- Beispiel:
  - `tar -cjf backup.tar.bz2 Bilder/ Dokumente/`
  - erstellt ein `bzip2` -komprimiertes Archiv

Wichtige Befehle & Optionen

Befehl	Optionen	Funktion
`tar`	`-c`, `-f`, `-j` (`bzip2`), `-t`, `-x`, `-z` (`gzip`)	Archiv erstellen
`gzip`		`gzip` komprimieren
`bzip2`		`bzip2` komprimieren
`gunzip`		`gzip` dekomprimieren
`bunzip2`		`bzip2` dekomprimieren

Zusammenfassung & Ausblick

Zusammenfassung

- einfache Suchmuster mit Globs
- Dateien und Verzeichnisse archivieren und komprimieren

Zusammenfassung & Ausblick

Zusammenfassung

- einfache Suchmuster mit Globs
- Dateien und Verzeichnisse archivieren und komprimieren

Nächstes Mal

- weitere Shellbefehle
- kompliziertere Suchmuster („Regular Expressions“)
- suchen und finden