

# Archivieren und Komprimieren, Globbs

## Linux-Kurs der Unix-AG

Benjamin Eberle

23. November 2015



UNIX  
AG

TU Kaiserslautern

**RH** Regionales  
Hochschul-  
Rechenzentrum  
Kaiserslautern **RK**

# Sonderzeichen in Dateinamen

- ▶ bestimmte Zeichen haben für die Shell eine Sonderbedeutung (z. B. Globs oder das Leerzeichen)
- ▶ können nicht einfach so als Dateinamen verwendet werden
- ▶ müssen „escaped“ werden: `\` vor das Sonderzeichen stellen
- ▶ oder: den Dateinamen in Anführungszeichen setzen
- ▶ `ls Komischer\ Date\*name` oder `ls "Komischer Date*name"`
- ▶ Verwendung in Dateinamen nicht zu empfehlen!

# Globs

- ▶ Wildcard/Platzhalter, z.B. `?`, `*`, `[...]`
- ▶ spezielle Zeichen, die mit mehreren Zeichen übereinstimmen
- ▶ nützlich, um mehrere Dateien gleichzeitig zu verarbeiten (kopieren, löschen, etc.)
- ▶ `?`: genau ein beliebiges Zeichen (außer `.` als erstes Zeichen)
- ▶ `*`: beliebig viele Zeichen, auch keins (außer `.` als erstes Zeichen)
- ▶ `[...]`: die gewünschten Zeichen können angegeben werden (z. B. `[avx]` oder `[a-z]`)

# Beispiele zu Globbs

- ▶ `a?c`: passt z. B. auf `abc`, `a.c`, `a?c`, `a0c`, `aDc` aber nicht auf `adec`
- ▶ `a*c`: passt z. B. auf `abc`, `a?c`, `a0c`, `adec`, `aaaaaac`
- ▶ `a[xyz]c`: passt nur auf `axc`, `ayc` und `azc`
- ▶ `a[a-z]c`: passt z. B. auf `aac`, `axc`, je nach Spracheinstellung auch Großbuchstaben
- ▶ `a[0-9]c`: passt z. B. auf `a0c`, `a3c`, `a9c`
- ▶ Kombination mehrerer Globbs auch möglich:
- ▶ `a[bc][de]f`: passt auf `abdf`, `acdf`, `abef`, `acef`

# Archivieren

- ▶ Zusammenfassen mehrerer Dateien zu einer einzigen
- ▶ komplette Verzeichnisbäume mit Dateinamen und Dateirechten auch möglich
- ▶ früher: Speichern auf Magnetband
- ▶ heute: Versand per E-Mail, Download von Webseiten, Backups

# tar

- ▶ `tar` (tape archive) archiviert Dateien
- ▶ `-c` (create) erzeugt ein neues Archiv
- ▶ `-x` (extract) entpackt das Archiv
- ▶ `-t` (table of contents) zeigt den Inhalt des Archivs an
- ▶ `-f <DATEI>` (file) schreibt oder liest das Archiv von `<DATEI>`, statt die Standard-Ein- oder Ausgabe zu benutzen
- ▶ `-f` muss immer die letzte Option sein (danach kommt die Datei)

# tar, Beispiele

- ▶ `tar -cf desktop.tar Desktop` archiviert das Verzeichnis `Desktop` und alles darin in die Datei `desktop.tar`
- ▶ `tar -tf desktop.tar` zeigt den Inhalt von `desktop.tar` an
- ▶ `tar -xf desktop.tar` packt den Inhalt von `desktop.tar` aus

# Komprimieren

- ▶ Komprimieren reduziert den Speicherbedarf von Daten
- ▶ meist als Option von `tar` verwendet
- ▶ alternativ Verknüpfung mit der Archivierung über Pipes



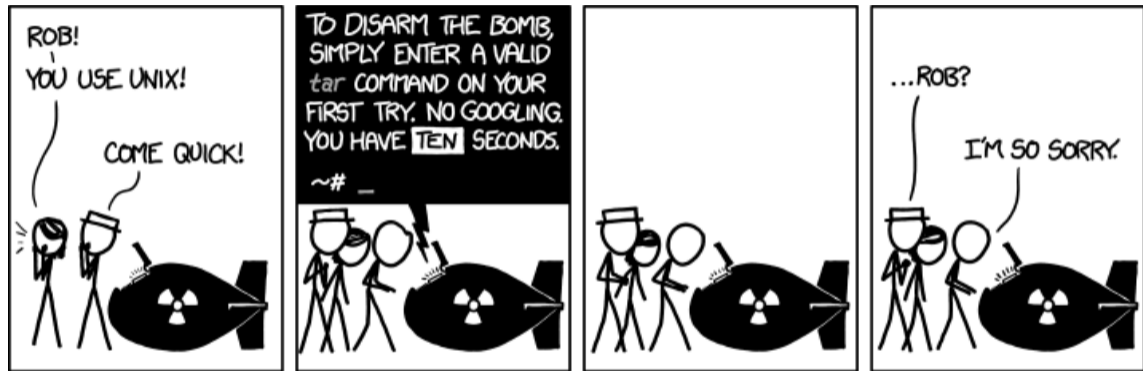
# gzip, bzip2

- ▶ `gzip` und `bzip2` sind Komprimierungsprogramme
- ▶ `bzip2` komprimiert etwas kleiner, braucht aber ein wenig mehr Rechenleistung beim Komprimieren
- ▶ Wichtige Optionen:
  - ▶ `-d` (decompress) entpackt eine Datei (dasselbe wie die Befehle `gunzip` und `bunzip2`)
  - ▶ `-l` (list, nur `gzip`) zeigt Informationen der komprimierten Datei an, z. B. die Dateigröße komprimiert und unkomprimiert
- ▶ Argumente: eine oder mehrere Dateien
- ▶ mehrere Dateien werden einzeln komprimiert

# Archivierung und Kompression

- ▶ `tar` kann `gzip` und `bzip2` direkt verwenden
- ▶ `tar -czf desktop.tar.gz Desktop`: packen mit `gzip`
- ▶ `tar -cjf desktop.tar.bz2 Desktop`: packen mit `bzip2`
- ▶ `tar -xf desktop.tar.gz`: entpacken, das Format wird automatisch erkannt
- ▶ `tar -xf desktop.tar.bz2`: entpacken, das Format wird automatisch erkannt
- ▶ manche Versionen von `tar` erkennen das Format nicht automatisch, in diesem Fall `-z` bzw. `-j` angeben

# tar Befehle



<https://xkcd.com/1168/>

# zip, unzip

- ▶ `zip` ist ein auch unter Windows verbreitetes Kompressionsformat
- ▶ kann direkt mehrere Dateien packen, `tar` wird nicht benötigt
- ▶ Option: `-r`: rekursiv packen
  
- ▶ `unzip` zum Entpacken
- ▶ Option: `-l` (list): Inhalt des Archivs anzeigen

# zip, unzip, Beispiele

- ▶ `zip -r desktop.zip Desktop` packt das Verzeichnis Desktop und alles darin in die Datei `desktop.zip`
- ▶ `unzip desktop.zip` packt `desktop.zip` aus
- ▶ `unzip -l desktop.zip` zeigt den Inhalt von `desktop.zip` an

# SSH

- ▶ `ssh` („Secure Shell“) startet eine Shell auf einem anderen Rechner über das Netzwerk
- ▶ Verbindung wird verschlüsselt aufgebaut
- ▶ Verwendung: `ssh benutzer@rechner`
- ▶ ohne Angabe des Benutzers wird der aktuelle Benutzer angenommen
- ▶ Option `-X`: Auf dem entfernten Rechner gestartete grafische Programme werden lokal angezeigt
- ▶ Anmerkung: beim erstmaligen Verbinden muss der Fingerprint des entfernten Rechners akzeptiert werden

# scp

- ▶ `scp` („Secure Copy“) kopiert Dateien auf einen anderen Rechner mittels `ssh`
- ▶ Verwendung (vom lokalen zum entfernten Rechner):  
`scp quelledatei benutzer@rechner:/pfad/zur/zieldatei`
- ▶ Verwendung (vom entfernten zum lokalen Rechner):  
`scp benutzer@rechner:/pfad/zur/quelledatei zieldatei`
- ▶ Optionen: `-r` (rekursiv)

# wget

- ▶ `wget` dient dazu Dateien via http aus dem Internet herunterzuladen
- ▶ Es können auch ganze Ordner oder Webseiten mit `wget` heruntergeladen werden -> Optionen siehe Manpage
- ▶ `wget URL` lädt die angegebene Datei herunter
- ▶ Beispiel: `wget https://www.unix-ag.uni-kl.de/_media/linuxeinsteiger_15ss_uebung4.pdf`



# Spickzettel

Befehl	Optionen	Funktion ( Argument )
tar	-c, -x, -t, -f	Dateien packen
	-z (gzip), -j (bzip2)	Komprimieren
zip	-r	Zip-Archive erstellen
unzip	-l	Zip-Archive entpacken
ssh	-X	[benutzer]@rechner
scp		Dateien via ssh kopieren
wget		Dateien herunterladen

## Globs

? genau ein beliebiges Zeichen

\* beliebig viele Zeichen