

Dateisystem 2, Suchen & Finden

Linux-Kurs der Unix-AG

Benjamin Eberle

15. Januar 2015



UNIX
AG

TU Kaiserslautern

RH Regionales
Hochschul-
Rechenzentrum
Kaiserslautern RK

- ▶ `ln („link“)` legt Verknüpfungen an, Verwendung wie `cp`
- ▶ ohne Optionen wird ein zweiter Name für die gleiche Datei erzeugt (Hardlink); geht nicht mit Verzeichnissen (sonst Zyklen im Dateibaum)
- ▶ `ls -l` zeigt in der zweiten Spalte den Link-Zähler an
 - ▶ zeigt bei Dateien die Anzahl der Links an
 - ▶ bei Verzeichnissen mindestens 2 wegen `.` (Link auf das Verzeichnis selbst) und für jedes Unterverzeichnis `+1` wegen `..` (Link auf das Elternverzeichnis)
- ▶ `ls -li` zeigt die sog. Inode-Nummer an (laufende Nummer für Dateien); bei Links ist diese gleich
- ▶ wird ein Link gelöscht, kann man über die anderen noch auf die Datei zugreifen

- ▶ Hardlinks sind verwirrend (verschiedene Dateinamen, aber gleicher Inhalt)
- ▶ `ln -s` legt einen symbolischen Link (Symmlink) an: Link zeigt auf das Ziel
- ▶ geht auch mit Verzeichnissen
- ▶ wird das Ziel gelöscht, zeigt der Link ins Nichts
- ▶ `ls -l` zeigt an, wo der Link hinzeigt

find

- ▶ `find` findet Dateien im Dateisystem
- ▶ erstes Argument ist das Verzeichnis, in dem gesucht werden soll
- ▶ die weiteren Argumente sind Suchkriterien:
 - ▶ `-name`: Muster für den Dateinamen (Wildcards erlaubt, dann aber in Anführungszeichen setzen)
 - ▶ `-iname`: Wie `-name`, unterscheidet aber nicht zwischen Groß- und Kleinschreibung (case-insensitive)
 - ▶ `-type`: Dateityp (`f`: Dateien, `d`: Verzeichnisse, `l`: Symlinks)
 - ▶ `-o` zwischen zwei Kriterien verknüpft diese mit Oder (Standard ist Und)
 - ▶ `!` vor einem Kriterium kehrt dessen Bedeutung um
 - ▶ weitere Parameter: Buch, Kapitel 6.4.4

find – Beispiele

- ▶ `find ~ -size +2M` – findet alle Dateien im Home, die 2MB oder größer sind
- ▶ `find ~ -type f -mtime -1` – findet alle Dateien im Home, die vor weniger als einem Tag verändert wurden
- ▶ `find ~ -name "*.conf" -o -name "*.cfg"` – findet Dateien mit den Endungen `.conf` und `.cfg` im Home

find -exec

- ▶ standardmäßig gibt find die gefundenen Dateien nur aus
- ▶ mit `find ... -exec <befehl> "{}" \;` kann auch für jede gefundene Datei ein Befehl ausgeführt werden
- ▶ Anwendungsbeispiel: alle Dateien löschen, die älter als X Tage sind
- ▶ Beispiel: `find ~ -size +2M -exec ls -lh "{}" \;` – ruft `ls -lh` für alle Dateien größer 2MB im Home auf

locate

- ▶ `find` durchsucht den Dateibaum bei jedem Aufruf neu, kann sehr lange dauern
- ▶ `locate` hat eine Datenbank und sucht darin, geht sehr viel schneller
- ▶ Datenbank muss regelmäßig aktualisiert werden
- ▶ Dateien, die nicht in der Datenbank stehen, werden nicht gefunden
- ▶ Argument: Dateimuster (Wildcards erlaubt)
- ▶ `locate` durchsucht immer den ganzen Dateibaum und findet auch Dateien, deren Namen das Suchmuster enthalten
- ▶ Beispiel: `locate README` – findet `README`, `README.txt`, `README.gz`, ...

- ▶ grep gibt alle Zeilen einer Datei aus, die ein Suchmuster enthalten
- ▶ Wichtige Optionen:
 - ▶ -i: bei der Suche nicht auf Klein- und Großschreibung achten
 - ▶ -v: nur die Zeilen ausgeben, die das Muster **nicht** enthalten
 - ▶ -r: Verzeichnisse rekursiv durchsuchen
 - ▶ -E: erweiterte reguläre Ausdrücke können verwendet werden
- ▶ Argumente: Suchmuster und Dateien/Verzeichnisse
- ▶ alternativ kann der zu durchsuchende Text auch per Pipe übergeben werden

Reguläre Ausdrücke

- ▶ einfache Suchmuster erlauben es nur nach Zeichenketten zu suchen
- ▶ aber schon bei mehreren möglichen Schreibweisen wird es kompliziert (z. B. Potential und Potenzial)
- ▶ Lösung: Reguläre Ausdrücke (regular expression, Regex)
- ▶ Muster können Platzhalter und Wiederholungsangaben enthalten

Regex – Platzhalter und Wiederholungsangaben

Platzhalter

- ▶ `.` steht für ein einzelnes Zeichen: `a.b` passt auf `acb`, `axb`, ...
- ▶ `[...]` steht für eine Klasse von Zeichen (z. B. `[tz]` für `t` oder `z`, `[a-z]` für alle Zeichen von `a` bis `z`, enthält je nach Spracheinstellung auch Großbuchstaben)

Wiederholungsangaben

- ▶ `?`: Zeichen kommt einmal oder keinmal vor: `ab?c` passt auf `abc` und `ac`
- ▶ `*`: Zeichen kommt beliebig oft vor: `ab*c` passt auf `ac`, `abc`, `abbbbbc`, ...
- ▶ `+`: Zeichen kommt mindestens einmal vor: `ab+c` passt auf `abc`, `abbbbbc`, aber nicht `ac` (nur mit `-E` oder `egrep`)

Unterschiede zwischen Globbs und Regexes

- ▶ Globbs kennen nur Platzhalter (für ein Zeichen oder mehrere)
- ▶ Regexes kennen Platzhalter (für ein Zeichen) und Wiederholungsangaben
- ▶ Globbs werden verwendet um mehrere Dateinamen auf einmal anzugeben, Regexes um Text zu durchsuchen
- ▶ Regexes sind daher viel komplexer als Globbs

	Glob/Wildcard	Regex
Ein Zeichen	?	.
Mehrere Zeichen	*	.* / .+

grep – Beispiel

- ▶ `grep "[^:]*:[!*][^:]*:" /etc/shadow` – gibt alle gesperrten Accounts aus
- ▶ Passwortfeld beginnt mit ! oder *: Account gesperrt
- ▶ `[^...]` passt auf alle Zeichen, die **nicht** angegeben sind, also `[^:]` auf alle Zeichen außer :

Alle Befehle

Befehl	Optionen
ln	-s
find	-(i)name, -type, -size, -mtime, -exec, -o, (!)
locate	
grep / egrep	-i, -v, -r, -E

Regexes

Platzhalter: ., [...], Wiederholungsangaben: ?, *, +