



**MDLUG e.V.**

# UNIX Power Tools

Helferlein für den Alltag

Maximilian Haupt  
mhaupt@cs.uni-magdeburg.de

Fakultät für Informatik,  
Otto-von-Guericke-Universität Magdeburg

Magdeburg, 28. Oktober 2008

1 Reguläre Ausdrücke

2 grep

3 find

4 sed

5 awk

6 Kombinationen

7 Schluss

# Reguläre Ausdrücke

- Syntaktische Regeln zur Beschreibung von Mengen
- Als Filter interpretierbar, der Zeichenketten matcht
- Auch als Textersetzung und -erzeugung

# Beschreibungselemente

- Normale Zeichen
- Beliebiges Zeichen: `.`
- Zeichengruppen: `[abc]`, `[a-z]`, `[ : digit : ]`
- Klammern: `()`, `{}`, `[]`, `<>`
- Quantoren: `?`, `+`, `*`, `{n}`, `{min,max}`
- Gruppierungen: `(abc)`
- Alternativen: `abc|ABC`
- Sondersymbole: `^`, `$`, `\`, `\b`, `\B`, `\<`, `\>`, `\n`
- Bedingungen: ...

# Beispiele

- $a$  → irgendwo im Wort ist ein  $a$
- $a^+b^*$  → mind. ein  $a$ , mind 0  $b$
- $[ab]^+$  → beliebige Folge von  $a$  oder  $b$
- $[a-z]^+$  → beliebige Folge von Kleinbuchstaben
- $[m-zA-Z]^+$  → alle Buchstaben ab  $m$  bzw  $M$
- $a^+\$$  → alle Wörter, die auf beliebig viele  $a$  enden
- $abc|ABC$  → entweder  $abc$  oder  $ABC$
- $[0-9]\{2, 5\}$  → 2 bis 5 mal eine Zahl
- $[^a]^*$  → alles ohne  $a$

# grep

- Textsuch-Programm für UNIX
- Statt vom Kommando `g/re/p` von `ed`  
***g**lobal search for a **r**egular **e**xpression and **p**rint out matched lines*
- Input von Dateien oder Streams
- Output immer Stream

# Beispiele

- **grep max /etc/passwd** - nach max in /etc/passwd suchen
- **cat /etc/passwd | grep max** - das gleiche, nur als Stream
- **grep "^ma[nx]:" /etc/passwd** - user man oder max suchen
- **grep "nologin\$" /etc/passwd** - alle user mit nologin
- **grep -i NVIDIA /var/log/Xorg.0.log** - nvidia im log suchen
- **grep -il studio \*** - nach studio suchen in . suchen
- **grep -rsniH studio .** (GNU grep) - rekursiv nach studio suchen

# find

- Suchen nach Dateien und Ordnern
- Aufruf: *find [Ordner] [Argumente]*
- Die wichtigsten Argumente
  - **-name** - name der Datei/Ordner
  - **-type** - z.B. f oder d
  - **-exec** - Befehl auf match ausführen
  - **-print** - match ausgeben (in der Regel standard)
  - **-ls** - ls auf match ausführen
  - **-perm** - nach Rechten suchen
  - ... - man find

# Beispiele

- **find . -type d -empty**  
rekursiv alle leeren Ordner listen
- **find . -type d -name "CVS"**  
alle CVS Ordner suchen
- **find . -type d -name "CVS" -or -name ".svn"**  
alle CVS und svn Ordner
- **find . -type d -name "CVS" -exec rm -rf {} \;**  
alle CVS Ordner löschen
- **find . -type f -name "\*\.(sh|ksh|ish)" -exec sh {} \;**  
alle sh Dateien ausführen
- **find /proc/acpi -type f -print -exec cat {} \;**  
alle Dateien in /proc/acpi mit Namen ausgeben

# sed

- sed = **s**tream **e**ditor
- Stream oder File als Input  
⇒ mit RegExp suchen und ersetzen
- Aufruf: *sed [Optionen] [File]*
- Die wichtigsten Argumente
  - **-e [script]** - Bearbeitungsscript ausführen
  - **-i[suffix]** - Files inline bearbeiten
  - **-f [scriptfile]** - Befehle aus scriptfile ausführen
  - ... - man sed

# Beispiele

- Allgemein: `s/RegExp/NeuerAusdruck/Optionen`
- **`sed -e 's/[Mm][Dd]/Magdeburg/'`**  
einmal MD durch Magdeburg ersetzen
- **`sed -e 's/[Mm][Dd]/Magdeburg/g'`**  
alle MD durch Magdeburg ersetzen
- **`sed -e 's,/home/max,/home/peter,'`**  
statt / auch andere Zeichen möglich

# Suchen und Ersetzen in vim

- **`:%s/reg/abc/optionen`**
- Als Option noch `c` möglich zum einzelnen commit

# awk

- General Purpose Programmiersprache
- 1977 an den Bell Labs entstanden
- Aus *sed* und *awk* ist dann *Perl* entstanden

# Beispiele

- **awk '{print "mv " \$1 " backup/" \$1}'**  
Befehl zum Verschieben ausgeben

# Kombinationen

- UNIX Prinzip: jedes Tool macht genau eine Aufgabe → Kombination ermöglicht sehr mächtige Aktionen

# Beispiele

- **alias wo find . -type f -exec grep -il \!^ {} \;**
- **find . -type d | grep -v 'privat'**
- **find . -name \*.h -or -name \*.c -exec sed -i -e 's/AlteFkt/NeueFkt/g' {} \;**
- **find src -name \*.h -exec grep -l Q\_OBJECT {} \; | sed -e 's/\.h/.cpp/g' -e 's,(.\*)/, \1/moc.,'**
- **ls -l | awk '/Oct/{print \$5,\$9}'**  
Größe und Name aller Dateien im aktuellen Verzeichnis, die im Oktober geändert wurden

# Was man mitnehmen sollte

- Reguläre Ausdrücke kann man immer gebrauchen, nicht nur in der Konsole
- **grep** für einfache Textsuche
- **find** zum durchforsten von Ordnerstrukturen
- **sed** zum einfachen Ersetzen von Pattern
- **awk**, wenns ein schwerer Fall ist ;)

# Ab ins Moonlight

- Vielen Danke für die Aufmerksamkeit