

spline_

Shell-Scripting
Teil 2 - Reguläre Ausdrücke

Alexander Gordon
Freie Universität Berlin

5. Vorlesung WS17/18

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

Was sind reguläre Ausdrücke?

Ein regulärer Ausdruck ist in der theoretischen Informatik eine Zeichenkette, die der Beschreibung von Mengen von Zeichenketten mit Hilfe bestimmter syntaktischer Regeln dient.

Es gibt zwei Arten von regulären Ausdrücken:

- ▶ **BRE** (Basic Regular Expressions) - Einfache Reguläre Ausdrücke
- ▶ **ERE** (Extended Regular Expressions) - Erweiterte Reguläre Ausdrücke

In der GNU-Implementierung regulärer Ausdrücke verstehen beide Dialekte jeweils alle Sprachelemente des anderen - sie werden lediglich ein wenig anders geschrieben.

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

BRE (grep)	ERE (egrep)	Bedeutung
<code>xy</code>	<code>xy</code>	Ein 'x' gefolgt von einem 'y'
<code>.</code>	<code>.</code>	Ein beliebiges Zeichen
<code>[xyz]</code>	<code>[xyz]</code>	Ein 'x' oder ein 'y' oder ein 'z'
<code>[a-z]</code>	<code>[a-z]</code>	Ein beliebiges Zeichen von 'a' bis 'z'
<code>[^xyz]</code>	<code>[^xyz]</code>	Ein beliebiges Zeichen außer 'x', 'y' und 'z'
<code>a{2,5}</code>	<code>a{2,5}</code>	zwei bis fünf mal 'a' (hintereinander)
<code>a{2,}</code>	<code>a{2,}</code>	'a' mindestens 2 mal
<code>a{2}</code>	<code>a{2}</code>	'a' genau 2 mal

BRE (grep)	ERE (egrep)	Bedeutung
*	*	Beliebig viele Zeichen. Auch kein mal
+	+	Das dem Pluszeichen vorangehende Zeichen einmal oder öfter
?	?	Das dem Fragezeichen vorangehende Zeichen null oder einmal
\(...\)	(...)	Gruppe von Ausdrücken
a b	a b	Entweder 'a' oder 'b'
^	^	Zeilenanfang
\$	\$	Zeilenende

BRE (grep)	ERE (egrep)	Bedeutung
<code>\<</code>	<code><</code>	Wortanfang
<code>\></code>	<code>></code>	Wortende
<code>\b</code>	<code>\b</code>	Wortanfang oder Ende
<code>\B</code>	<code>\B</code>	Weder Wortanfang oder Ende
<code>\1...</code>	<code>\1...</code>	Rückbezug

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit `grep/egrep`

Ersetzen mit `sed`:

Schneiden von Text mit `cut`

Wörterausgabe mit `awk`

Suchen mit grep/egrep

grep „regulärer Ausdruck“ testdatei
 egrep „regulärer Ausdruck“ testdatei

Beispiel:

Listing 1: text.txt

```
last@spline.de
@last
last@
linuxkurs@spline.de
last.spline.de
last@@spline.de
```

Ziel: Die richtigen E-Mail Adressen rauszufiltern.

Lösung: egrep "^([@]+@{1})[(@)]+\$"text.txt

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

sed 's/text1/text2/g' testdatei

Parameter:

-n	Verhindert das automatische Ausgeben des Ergebnisses
-e skript	Angeben eines sed-Skriptes, mehrere sed-Skripte sind möglich. Bei nur einem sed-Skript kann -e weggelassen werden
-f skriptfile	Das sed-Skript steht in der Skriptdatei, nicht auf der Kommandozeile
-i	Die Textdatei wird verändert, anstatt das Ergebnis auf Standardausgabe auszugeben

Beispiel:

Listing 2: text.txt

```
text1  
text2  
text4  
Superman  
Spiderman  
Batman  
man is man
```

Ziel: text4 durch text3 zu ersetzen, man durch woman zu ersetzen nur bei den Heldennamen.

Lösung:

```
sed -i 's/text4/text3/g' text.txt  
sed -i 's/(Super|Spider|Bat)man/\1woman/g' text.txt
```

Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

Schneiden von Text mit cut

cut <paramete1,...,parameter n> testdatei

Parameter:

-c	Zeilensegmente Zeichenweise
-f	Zeilensegmente Felderweise
-d	Trennzeichen

Beispiel:

Listing 3: text.txt

```
| one:two:three:four:five:six:seven:eight:nine:ten
```

Ziel: Wort „one“ ausgeben, Wort „five“ ausgeben, alle Wörter ab „six“ ausgeben

Lösung:

```
| cut -c 1-3 text.txt  
| cut -d : -f 5 text.txt  
| cut -d : -f 6- text.txt
```


Reguläre Ausdrücke

Was sind reguläre Ausdrücke?

Syntax

Suchen mit grep/egrep

Ersetzen mit sed:

Schneiden von Text mit cut

Wörterausgabe mit awk

Wörterausgabe mit awk:

awk ist wie sed eine eigenständige Sprache. Hier werden wir nur eine Möglichkeit betrachten wie man einzelne Wörter nach ihren Feldnummer ausgeben kann.

```

      .o+`
     `ooo/
      +oooo:
     +ooooo:
    -+ooooo+:
   `/:-!++oooo+:
  `/+////+////////:
 `/+////////+////////:
`/+++ooooooooooooo/`
./ooooSSSSO++OSSSSSO+`
.ooSSSSSO-`"/OSSSSS+`
 -OSSSSSO.      :SSSSSSO.
 :OSSSSSS/      OSSSSO+++
 /OSSSSSSS/     +SSSSOOO/-
`/OSSSSSO+/:-   -:/+OSSSSO+-
`+SSO+:`-`      `.-/+OSO:
`++:..          `.-/+/
`

```

```

centzilius@megumin
OS: Arch Linux
Kernel: x86_64 Linux 4.13.12-1-ARCH
Uptime: 0m
Packages: 823
Shell: zsh 5.4.2
Resolution: 1366x768
WM: i3
CPU: Intel Atom Z3735F @ 4x 1.8326GHz
GPU: intel
RAM: 358MiB / 1933MiB

```

Jede Distro hat sein eigenes Logo und Namen. Wir können jetzt ein Skript schreiben, das und z.B. den Namen (OS:) ausgibt:

```
screenfetch | grep OS | awk '{print $5}'
```

Hier kommt „Arch“ raus.

Wir können das noch schöner machen:

```
screenfetch | grep OS | awk '{print $5" "$6}'
```

Ausgabe: „Arch Linux“