Betriebssysteme

Seminar 2

Inhalt

- Wiederholung
- Regulärer Ausdruck
- grep
- sed

Wiederholung

- Shell-Skript und Befehlszeile
- Liste der Befehle und Zusammengesetzte Befehle
- Umleiten
- UNIX-Kontrollstrukturen

Regulärer Ausdruck

- = ist eine Folge von Zeichen, in dem einige Zeichen besondere Bedeutung haben

- das Zeichen "\" vermeidet die spezielle Bedeutung eines Zeichens, "..." vermeidet jedes Zeichen außer $ und '...' und '...' vermeidet jedes Zeichen

- sehr seltsame und hässliche, aber kompakte und flexible Sprache für passenden Text

Regeln für reguläre Ausdrücke

- **.**: entspricht einem einzelnen Zeichen
- ****: Escape ändert die Bedeutung des darauf folgenden Zeichens zwischen normal und speziell
- **[abc]**: entspricht jedem einzelnen Zeichen, das in der Liste angezeigt wird (in diesem Fall a oder b oder c)
- **[a-z]**: entspricht einem einzelnen Zeichen, das zum Bereich gehört (in diesem Fall einem Kleinbuchstaben)
- **[^0-9]**: entspricht einem einzelnen Zeichen, das nicht zum Bereich gehört (in diesem Fall alles, was keine Ziffer ist)
- **^**: Zeilenanfang

Regeln für reguläre Ausdrücke

- **$**: Ende der Linie
- **\<**: Wortanfang
- **\>**: Ende des Wortes
- **( )**: Gruppierung mehreren Zeichen in einem Ausdruck
- ***:** vorheriger Ausdruck null oder mehrmals
- **+**: vorheriger Ausdruck einmal oder mehrmals
- **?**: vorheriger Ausdruck null oder einmal
- **{m,n}**: vorheriger Ausdruck mindestens m und höchstens n Mal
- **|**: logisches ODER zwischen Teilen des regulären Ausdrucks

Aufgabe

Geben Sie ausgehend von den oben dargestellten Aspekten die unten dargestellten Ausdrücke an:

1. .*
2. [a-zA-Z02468]
3. [,]
4. ^[^0-9]+$

GREP

- steht für **g**lobal **r**egular **e**xpression **p**rint
- sucht in einer oder mehreren Dateien nach einer bestimmten Zeichenfolge und zeigt das Ergebnis bei der Standardausgabe an.

grep [-[[AB]]num] [-[CEFGVBchilnsvwx]] [-e] regulären_Ausdruck [-f Name-Szenario] [Dateiliste]

Optionen

- **-c (--count):**
  - Trefferanzahl anzeigen
  - die Anzahl aller Zeilen, die das Suchmuster beinhalten, kann man mit -c oder --count ausgeben lassen.
  - damit liefert grep die reine Anzahl, also z.B. “4” für 4 Treffer zurück, sonst nichts.

Beispiel:

```
grep -c 'abc' datei.txt

grep --count 'abc' datei.txt
```

Optionen

- **-i (ignore case)**
  - Groß- und Kleinschreibung ignorieren
  - standardmässig sucht der **grep-Befehl** *case-sensitive*. Das heisst, die Groß- und Kleinschreibung wird beachtet. Will man dies, also *case-insensitive*, suchen, muss man die Option -i hinzufügen.

Beispiel:

```
grep -i 'abc' bsp1
```

Optionen

- **-l:**
  - nur Dateinamen ausgeben lassen
  - manchmal braucht man nur den/die Dateinamen, in denen der Suchstring gefunden wurde. Das kann man über die Option -l (kleines “L”) erreichen.

Beispiel:

```
grep -l 'a' *.txt
```

Optionen

- **-L:**
  - zeige gefundene, NICHT matchende Dateinamen
  - auch gibt es Anwendungsfälle, in denen man nur die Dateinamen aufgelistet haben will, in denen der Suchstring NICHT gefunden wurde. Dafür nutzt man die Option -L.

Beispiel:

```
grep -l 'a' *.txt
```

Optionen

- **-n:**
  - Zeilennummer für jedes Match
  - wird vor jeder Match die Zeilennummer des Matches geschrieben

Beispiel:

```
grep -n 'abc' datei.txt
```

Optionen

- **-v:**
  - NICHT betroffene Zeilen anzeigen
  - zeigt alle Zeilen an, die die angegebene Zeichenfolge nicht enthalten

Beispiel:

```
grep -v 'abc' datei.txt
```

Optionen

- **-w:**
  - nur ganze Wörter finden
  - per default findet **grep** alle Vorkommen des Suchstrings, sowohl in Wortteilen als auch in ganzen Wörtern. Nur auf Vorkommen in ganzen Wörtern beschränkt man per -w.

Beispiel:

```
grep -w 'abc' datei.txt
```

Optionen

- **-h (hide):**
  - zeigt den Dateinamen nicht an

  Beispiel:

  grep -h 'a' *

Optionen

- **-E:**
- verwenden erweiterte reguläre Ausdrücke
- ist wie grep, nur mit dem Unterschied, dass man auch +, ?, | und () als Regular Expression nutzen kann.

Aufgaben

Man betrachtet eine zufällige Textdatei seminar2.txt:

1. Zeigen Sie alle nicht leeren Zeilen an.
2. Zeigen Sie alle leeren Zeilen an.
3. Zeigen Sie alle Zeilen an, die eine ungerade Anzahl von Zeichen enthalten.

SED

- bezeichnet auch als nichtinteraktiver Texteditor
- anstatt Änderungen direkt an der Datei vorzunehmen, erzeugt man zunächst eine temporäre Datei, deren Inhalt anschließend an die Ausgangsdatei übergeben wird. Jede Zeile einer Datei wird einzeln eingelesen, bearbeitet und dann wieder ausgegeben. (Standardmäßig wird die Datei nicht geändert, sondern das Ergebnis der Verarbeitung der Eingabedatei angezeigt)
- *Die wichtigste Funktion:* bestimmte Zeichenketten in der Datei zu suchen und dann durch andere Zeichen zu ersetzen.

sed [-n] [-g] [-e Szenario] [-f Skriptdatei] [Dateiliste] ...

Bedingungen

- Ein sed-Szenario besteht aus Linien der Form:

  Bedingung Anweisung

  - **keine Bedingung:** wahr für alle Zeilen in der Datei
  - **n:** wahr für die Zeilen mit dem Zeilennummer n (Zeilen sind in der Dateiliste kumulativ nummeriert)
  - **$:** wahr für die letzte Zeile in der Datei
  - **/ regulären Ausdruck /** : wahr für die Zeilen, die mindestens einen Teilstring enthalten, der dem regulären Ausdruck entspricht
  - **expr1,expr2:** wahr für die Zeilen, die zwischen der Zeile, die mit Ausdruck1 übereinstimmt, und der Zeile, die mit Ausdruck2 übereinstimmt

Beispiele:

$ sed 1,10 Anweisung fis1

(*führt die Anweisung in den Zeilen 1 bis 10 aus*)

$ sed 10,$ Anweisung fis1 fis2

(*führt die Anweisung in den Zeilen von 10 bis zum Ende der Datei aus, die durch Verketten von fis1 mit fis2 erhalten wird*)

Suchen / Ersetzen

sed -E "s/regex/replacement/flags" datei.txt

- **s** - ist der Such- / Ersetzungsbefehl
- **/** ist das Trennzeichen und kann ein beliebiges anderes Zeichen sein. Das erste Zeichen nach dem Befehl s wird als Trennzeichen betrachtet
- Die Flags am Ende können g, i oder beides sein
  - **g** - führt den Ersatz überall in der Linie. Ohne sie wird nur das erste Erscheinungsbild ersetzt
  - **i** - Führt eine Suche ohne Berücksichtigung der Groß- und Kleinschreibung durch
- Die Ersetzung kann Verweise auf die Ausdrücke enthalten, die in der Regex als \ 1, \ 2 usw. gruppiert sind, wobei die Nummer die Reihenfolge ist, in der die Gruppen in der Regex erscheinen

Transliterate

sed -E "y/characters/replacement/" a.txt

- **y** - ist der Transliterationsbefehl
- **/** ist das Trennzeichen und kann ein beliebiges anderes Zeichen sein. Das erste Zeichen nach dem Befehl y wird als Trennzeichen betrachtet
- Die *characters* und der *replacement* müssen gleich lang sein

Zeilen löschen, die einem regulären Ausdruck entsprechen

```
sed -E "/regex/d" a.txt
```

- **/** ist das Trennzeichen
- **d** - ist der Befehl zum Löschen von Zeilen

Aufgaben

Lassen Sie uns mit dem Inhalt von / etc / passwd arbeiten:

1. Zeigen Sie alle Zeilen an und ersetzen Sie alle Selbstlaute (Vokale) durch Leerzeichen.
2. Zeigen Sie alle Zeilen an und konvertieren Sie alle Selbstlaute (Vokale) in Großbuchstaben
3. Zeigen Sie alle Zeilen an und löschen Sie die Zeilen mit fünf oder mehr Ziffern: