9.Hilfsprogramme für die Arbeit mit Files

9.1 Stream-Editor "sed"
9.2 Familie von Zeichenkettensuchprogrammen "grep", "egrep", "fgrep"
9.3 File-Such-Programm "find"

9.1. Der Stream-Editor sed

Ein kleiner Übersetzer für alle Fälle.

Grundätzliches

sed wurde vom zeilenorientierten, dialogfähigem Editor ed abgeleitet.

sed ist nicht dialogorientiert!!!!

sed liest die Editorkommandos von der Kommandozeile oder aus einer Datei.

sed liest die zu editierenden Daten aus einem File oder von der Standardeingabe.

sed gibt die editierten Daten immer auf der Standardausgabe aus!!!!!!!
Achtung!!! Nicht die zu editierende Datei gleichzeitig als Standardausgabe benutzen!!! Totaler Datenverlust!!!!

sed ist ein kleiner leistungsfähiger Editor für Modifizierung von Dateien oder zur Extrahierung von Daten. Er wird häufig in Shellscripten eingesetzt. Er ist schneller als awk, hat aber eine geringer Leistungsfähigkeit.

```
Prinzipielle Funktionsweise des sed
-----------------------------------

1. Als erstes werden alle Editierkommandos eingelesen
   (Kommanodzeile oder Skript-Datei)

2. Dann wird die erste Zeile aus der Eingabedatei in einen
   Eingabepuffer gelesen (Standardeingabe oder File)

3. Jetzt werden der Reihe nach alle Editorkommandos der
   Reihe nach geprüft, ob sie anwendbar sind und wenn ja
   sofort ausgeführt.                                     b1

4. Das Ergebnis aller Editorkommandos wir auf die
   Standardausgabe ausgegeben.

5. Der Eingabepuffer wird gelöscht.

6. Eine neü Zeile wird in den Eingabepuffer gelesen
   und bei 3. fortgesetzt. Bei Erkennung von EOF beendet
   sed die Arbeit.
```



```
Der Aufruf von sed
------------------
 sed  [-nV]  [--quiet]  [--silent] [--version]
      [-e <editorkommando>] [-f <skriptdatei>]
      [-i[<SUFFIX>] [--in-place[=<SUFFIX>]]
      [--expression=<editorkommando>]  [-r]
      [--file=< skriptdatei > ] {<dateiname>}

   -n, --quiet, --silent
          Zeile wird nur ausgegeben, wenn dies explizit durch ein
          p-Kommando innerhalb der Editorkommando gefordert wird.
   -V,--version
          Ausgabe der Verionsnummer
   --help
          Ausgabe einer kurzen Hilfeinformation für den
          Aufruf von sed
   -f <Skript-file>, --file=<Skript-file>
          Definition eines Files, das die Editorkommando enthält.
   -i[<SUFFIX>]
          Editieren im aktuellen File
   -e <editorkommando>
          Definition eines Editorkommandos. Kann mehrfach vorkommen.
          Wird nur ein Editorkommando benötigt, kann diese auch ohne
          "-e" angegeben werden.
   <dateiname>
          Name des zu editierenden Files. Mehre Filenamen sind zulässig.
          Ist kein Filename spezifiziert, wird Standardeingabe benutzt.
   -r
          Benutzung von erweiterten (fast vollständigen) regulären
          Ausdrücken.
```

```
Beispiele:
     # einige sed-Kommandos
     #   p  - ausgeben
     #   s  - ersetzen

     sed -V
     sed --version               # Version ausgeben
     sed -e "p"  d1 > d1-edit                                b0
     sed "p"  d1 > d1-edit       # Ausgbe aller Zeilen doppelt
     sed -e "s/a/aa/" -e p d1
     sed  "s/a/aa/" d1           # Substitution
     sed -e s/a/aa/ -e p d1
     sed -n -e s/a/aa/ d1
     sed -n -e s/a/aa/ -e p d1
     sed --quiet -e s/a/aa/ -e p d1
     sed --silent -e s/a/aa/ -e p d1  # nur p-Zeilen ausgeben
     sed -n --expression=s/a/aa/ -e p d1
     sed -f s1 -e p d1
     sed -e p -f s1 d1
     sed -e p -f s1 -e p d1
     sed --file=s1 -e p d1
```



```
Editorkommando
--------------

[<adresse1> [","<adresse2>]]<kommando>[<argument>]

[<adresse1>","[<adresse2>]]
     Durch diese Adressen wird ein Bereich von hintereinander
     stehenden Zeilen beschrieben. Dabei bedeutet:
   keine Adresse -   alle Zeilen des Eingabefiles
   eine Adresse  -   alle Zeilen, auf die die Adresse zutrifft
   Bereich von Eingabezeilen
                 -   alle Zeilen dieses Bereiches, die Grenzen
                     gehören zum Bereich.

<adresse1> und <adresse2> stellen Adressen dar. Eine Adresse
ist:

  ganze Zahl  - Nummer einer Zeile, bei mehreren Eingabedateien
                wird fortlaufend gezählt.

    $         - Dollarzeichen steht für die letzte Eingabezeile,
                letzte Zeile der letzten Eingabedatei.
                                                             b2
  /<regulärer Ausdruck>/
              - adressiert alle Zeilen, die einen String enthalten,
                der auf den regulären Ausdruck passt.
```

Reguläre Ausdrücke im sed (eingeschränkt)

Metazeichen:

```
\       - Maskierung eines Metazeichens
^       - Zeilenanfang
$       - Zeilenende
.       - beliebiges Zeichen, ausser <NL>
[... ]  - Klasse von Zeichen
          z.B.  [abc]    - eines der Zeiche a, b oder c
                [a-fs-z] - eines der Zeichen a bis f
                                     oder s bis z
[^...]  - Komplementklasse von Zeichen
          z.B. [^A-Z] - alle Zeichen, ausser Grossbuchstaben
*       - Null oder beliebig häufiges Auftreten des
          vorangegangenen Zeichens.
          z.B.   a*b   - bedeutet b, ab ,aab ,aaab, ...
\(<str>\)      - String <str>
\{<nr>\}       - genau <nr>-maliges Vorkommen
\{<nr,\}       - mindestens <nr>-maliges Vorkommen
\{<nr1,<nr2\} - mindestens <nr1>-maliges Vorkommen, höchstens
                <nr2>-maliges
\<nr>   - entspricht dem <nr>-ten Ausdruck der zuvor
          durch  \(<str>\) in dem selben String auftrat.
          z.B.  "^\(.*\)\1$" - entspricht allen Zeilen, die
                aus zwei gleichen Zeichenketten bestehen.
```

b3

Beispiele:

Einfache Zeichenketten:

```
/bin/sed -n -e /o/p d2
/bin/sed -n -e /er/p d2
/bin/sed -n -e /o/,$p d2
/bin/sed -n -e 3,/o/p d2
/bin/sed -n -e /o/,/i/p d2
```

b4

Zeilenanfang und Zeilenende:

```
^<String> - <String> am Zeilenanfang
<String>$ - <String> am Zeilenende
^<String>$ - Zeile enthält nur <String>
^$         - Leerzeile
^.$        - Zeile mit genau einem Zeichen
^....$     - Zeile mit genau vier Zeichen

/bin/sed "/^#/d" d2   - entfernt alle Kommentarzeilen aus b1
```

Zeichenklassen - Komplementklassen:

```
/bin/sed -n "/[ö]/p" d2
/bin/sed -n "/^[^ö]*$/p" d2
```

b5

```
  Wiederholungen:

    /bin/sed -n '/^.\{4\}$/p' d2
    /bin/sed -n '/^.\{5,8\}$/p' d2
                                                                  b6
  Wiederholungen von Buchstaben

    /bin/sed -n '/ir*/p' d2
    /bin/sed -n '/irr*/p' d2
                                                                  b7
  n-ter Teilausdruck

    /bin/sed -n '/\(alle\) \1/p' d3
    /bin/sed -n '/\(....\) \1/p' d3
    /bin/sed -n '/\(....\).*\1/p' d3
                                                                  b8
```



```
Übersicht sed-Kommandos
-----------------------

Zeilenorientierte Kommandos
    a\   - append lines (Anfügen)
    c\   - change line  (Ersetzen der Zeile)
    d    - delete line  (löschen einer Zeile)
    i\   - insert line  (einfügen vor der Zeile)
    n    - next line  (Ausgabe des Eingbepuffers)

Ersetzungskommandos
    s    - substitute (ersetzen eines regulären Ausdrucks)
    y    - translate (übersersetzen von mehreren Zeichen )

E/A-Kommandos
    l    - list (Ausgabe des Eingabepuffers, nicht druckbare
           Zeichen werden übersetzt)
    p    - print (Ausgabe des Eingabepuffers)
    r    - read (Lesen einer Datei und auf ausgeben)
    w    - write (Eingabepuffer in Datei schreiben)

Mehrzeilen-Kommandos
    D    - delete first part of pattern (Löschen)
    N    - next line (nächste Eingabezeile an Eingabepuffer anhängen)
    P    - print (erster Teil des Eingabepuffers ausgeben)
```

```
Kommandos für Zwischenspeicherung
    g    - Eingabepuffer wird durch Zwischenspeicher überschrieben
    G    - Zwischenspeicher am Ende des Eingabepuffers
           anhängen, durch <NL> getrennt
    h    - Zwischenspeicher durch Eingabepuffer überschreiben
    H    - Inhalt des Eingabepuffers am Ende des Zwischspeichers
           anhängen, durch <NL> getrennt
    x    - exchange (austauschen Zwischenspeicher und Eingabepuffer)

Kommandos für die Ablaufsteürung
    b    - branch (Sprung zur Marke)
    t    - test jump (bedingter Sprung im sed-Script)
    !    - Verneinung der Adressangabe
    :ma  - Marke im sed-Script

Sonstige Kommandos
    q    - quit (Ende)
    =    - Ausgabe der Zeilennummer als eigene Zeile
    #    - Kommentar
```



```
Zeilenorientierte Kommandos
---------------------------

    a\   - append lines

    Die nachfolgende Zeile wird nach Ausgabe des Eingabepuffers
    ausgeschrieben. Sollen mehrere Zeilen angefügt werden, so sind
    die <NL>-Zeichen durch "\" zu maskieren.

      Beispiel:

           /bin/sed '/^#/a\
           #------------------------#
           ' d2
                                                          b9,b10

    c\   - change line

    Ersetzen des Eingabepuffers durch die nachfolgende Zeile.
    Der Eingabepuffer wird dann ausgegeben. Mehrere Zeilen können
    wie bei a\ ersetzt werden.

      Beispiel:

           /bin/sed '/^#/c\
           #----------hier war ein Kommentar--------------#
           ' d2
                                                         b11,b12
```

```
    i\   - insert line

    Einfügen von nachfolgenden Zeilen vor der aktuellen Zeile aus
    dem Eingabepuffer.

      Beispiel:

           /bin/sed '/^#/i\
           #--------es folgt ein Kommentar-------------#' d2
                                                        b13,b14

    d    - delete line

    Löschen einer Zeile aus dem Eingabepuffer. Die Arbeit wird bei
    der nächsten Zeile am Beginn des sed-Scripts fortgesetzt.

      Beispiel:

           /bin/sed '/^#/d' d2
                                                        b15

    n    - next line

    Ausgabe des Eingbepuffers. Einlesen der nächsten Zeile und
    fortsetzen des Scriptes an der aktuellen Position.

      Beispiel:

           /bin/sed -e '/^#/n' -e '/^#/d' b16
                                                        b16
```



```
Ersetzungskommandos
-------------------

    s/<regulärer Ausdruck>/<String>/[<flag>]    - substitute

    Ersetzen eines regulären Ausdrucks durch einen String.
    <flag> - g        - global - ersetzen aller möglichen Teilstrings
             p        - print - Ausgabe des Eingabepuffers, falls
                                eine Ersetzung stattfand
             w <file> - write - Ausgabe des Eingabepuffers, falls
                                eine Ersetzung stattfand in das
                                File <file>

      Beispiel:

           /bin/sed -e 's/o.$/xxxxxx/' -e '/^#/d' d2
                                                        b17

    y/<String1>/<String2>/    - translate

    Ersetzen der in <String1> vorkommenden Zeichen durch die
    an der entsprechenden Position in <String2> vorkommenden
    Zeichen. <String1> und <String2> müssen gleich lang sein.

      Beispiel:

           /bin/sed -e 's/äiou/AEIOU/' -e '/^#/d' d2
                                                        b18
```

```
E/A-Kommandos
-------------

    p    - print

    Ausgabe des Eingabepuffers. Diese Operation findet zu dem
    Zeitpunkt statt, zu dem das p-Kommando im sed-Script steht.
    Die bis dahin durchgeführten Veränderungen am Ausgabepuffer
    werden mit ausgegeben.

      Beispiel:
         /bin/sed -e /^#/d  -e 'y/äiou/AEIOU/' \
             -e p \
             -e 'y/bcdfghjklmnpqrstvwxyz/BCDFGHJKLMNPQRSTVWXYZ/' d2

                                                              b19

    l    - list

    Ausgabe des Eingabepuffers. Dabei werden nicht druckbare
    Zeichen als ASCII-Werte (drei Oktalziffern mit vorangestelltem
    "\") oder, wenn möglich, in C-Notation ausgegeben. Überlange
    Zeilen werden in mehrere Zeilen zerlegt. Für das Zeilenende wird
    ein "$" ausgegeben.

      Beispiel:

         /bin/sed -n -e l d5      # nicht druckbare Zeichen werden
                                 # sichtbar                      b20
```



```
    r <file>   - read

    Weiterbearbeiten der aktuellen Eingabepuffers. Ausgabe der fertigen
    Eingabepuffers. Lesen der Datei <file> und Ausgabe dieses Files.
    Achtung: Zwischen dem Buchstaben "r" und dem Filenamen steht
    genau ein Leerzeichen!!!

      Beispiel:

           /bin/sed  -e '/worf/r d5' -e 's/orf/xxxxx/' d2
                                                                 b21
    w <file>   - write to file

    Der aktuelle Eingabepuffer wird an die Datei <file> angehängt.
    Vor Beginn der Arbeit wird eine leere Datei <file> erzeugt.
    Nach dem w-Kommando darf kein weiteres Kommando folgen.
    Achtung: Zwischen dem Buchstaben "w" und dem Filenamen steht
    genau ein Leerzeichen!!!

      Beispiel:

           /bin/sed  -e '/o/w d6' d2
                                                                 b22
```

Mehrzeilen-Kommandos

N - next line

Die nächste Eingabezeile an Eingabepuffer angehängt. Das <NL>-Zeichen bleibt im Eingabepuffer erhalten. Nachfolgende Kommandos können jetzt dieses <NL>-Zeichen ersetzen.

Beispiel:

```
/bin/sed  -e '/-$/N' -e 's/-\n//' d7
  Achtung: In diesem Beispiel können nie mehr als zwei
           Zeilen miteinander verbunden werden!!
```

b23

D - delete first part of pattern

Löschen des ersten Teils des Eingabepuffers bis zum <NL>-Zeichen. Danach wird das gesammte sed-Script noch einmal gestartet. Wenn kein <NL>-Zeichen im Eingabepuffer vorhanden ist entspricht dieses Kommando dem d-Kommando. Dieses Kommando macht nur in Verbindung mit dem N-Kommando sinn!!!

P - print

Erster Teil des Eingabepuffers wird bis einschliesslich eines <NL>-Zeichens ausgegeben. Wenn kein <NL>-Zeichen im Eingabepuffer vorhanden ist entspricht dieses Kommando dem p-Kommando.

Beispiel:

```
    /bin/sed  -f s24 d8
 s24:
    /Raum/P
    /Raum/D
    N
    N
    N
    s/[12] [a-zA-Z][a-zA-Z]*//g
    s/\([34]\) \([a-zA-Z]\)/\2/g
    s/\n/ /g
    s/^/   /
    P
    D
```

b24

```
Kommandos für Zwischenspeicherung
---------------------------------
    Neben dem Eingabepuffer wird im sed auch noch ein Zwischenspeicher
    verwaltet. Zwischen Eingabepuffer und Zwischenpuffer können Daten
    hin und her transportiert werden.

    g    - Eingabepuffer wird durch Zwischenspeicher überschrieben

    h    - Zwischenspeicher durch Eingabepuffer überschreiben

      Beispiel:

           /bin/sed -n -f s25 d9
       s25:

             1h
             1d
             /^$/p
             /^$/d
             /#/g
             p
                                                                  b25

    H    - Inhalt des Eingabepuffers am Ende des Zwischspeichers
           anhängen, durch <NL> getrennt
```



```
    G    - Zwischenspeicher am Ende des Eingabepuffers
           anhängen, durch <NL> getrennt
      Beispiel:

           /bin/sed -n -f s26 d2

          s26:
             /^$/p
             /^$/d
             /o/H
             /o/d
             $G
             p
                                                                     b26

    x    - exchange (austauschen Zwischenspeicher und Eingabepuffer)
      Beispiel:

           /bin/sed -n -f s27 d2
         s27:
           1h
           1d
           /^$/p
           /^$/d
           /o/x
           p
           /#/x
                                                                     b27
```

```
Kommandos für die Ablaufsteürung
---------------------------------

    :ma    - label

    Definiert eine Sprungmarke für die Kommandos b und t.
    Eine Marke ist ein normaler Bezeichner.

      Beispiel:

        :marke
        :Ende

    b <label>      - branch

    Sprung zur Marke <label>. Zwischen dem b-Kommando und der
    Marke muss ein Leerzeichen stehen.

      Beispiel:

        /bin/sed -n -f s28 d2
       s28:
         /worf/b ENDE
         /o/d
         :ENDE
         p
                                                             b28
```



```
    t <label>      - test and jump

    Bedingter Sprung im sed-Script. Falls nach dem Einlesen in
    den Eingabepuffer bzw. nach dem letzten t-Kommando eine
    Substitution stattgefunden hat wird ein Sprung zur angegebenen
    Marke <label> ausgeführt.

      Beispiel:

         /bin/sed -n -f s29 d2
       s29:
         :Start
         /o/s/o/XXX/
         t Start
         p
                                                             b29

    !<Kommando>      - Verneinung der Adressangabe

    Wenn die Vorangestellte Adressangabe zutrifft, wird das
    nachfolgende Kommando <Kommando> nicht ausgeführt.

      Beispiel:

          /bin/sed -n -e '/worf/!s/o/XXX/' -e p  d2
                                                             b30
```

{ .. } - Klammerung von Kommandos

Hierdurch können Kommandos zu Gruppen zusammengefasst werden, die nur ausgeführt werden, wenn die vorangestellte Adressangabe passt.

Beispiel:

```
     /bin/sed -n -f s31 d2
   s31:
     /o/{
        s/r/R/g
        s/c/C/g
     }
     p
                                                      b31
```


Sonstige Kommandos
----

q   - quit

Beenden des Streameditors. Der aktuelle Eingabepuffer wird ausgegeben und der Streameditor wird beendet.

Beispiel:

```
     /bin/sed  -e '3q' d2
                                                      b32
```

=   - Ausgabe der Zeilennummer als eigene Zeile

```
     /bin/sed  -e '=' d2
                                                      b33
```

 #   - Kommentar

Zeilen, die mit # beginnen werden als Kommentar gewertet.