

### 3. Der Textstrom-Editor sed

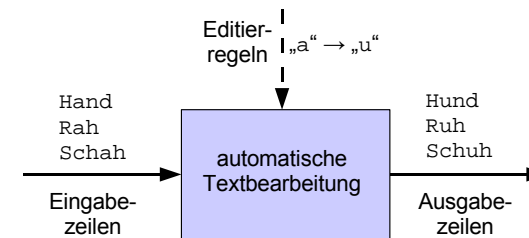
### Inhalte der Vorlesung

1. Einführung
2. Lexikalische Analyse
- 3. Der Textstrom-Editor sed
4. Der Scanner-Generator lex
5. Syntaxanalyse und der Parser-Generator yacc
6. Syntaxgesteuerte Übersetzung
7. Übersetzungssteuerung mit make

### 3. Der Textstrom-Editor sed

- 3.1 Grundprinzip eines Textstrom-Editors
- 3.2 Reguläre Ausdrücke in sed

### Grundprinzip eines Textstrom-Editors



## Einfache typische Anwendungen

- kleine Textmodifikation
  - Betreff in Vacation-Text einsetzen
- Formatierung einer Liste etwas ändern
  - festes Präfix in einer Dateiliste entfernen

## Betreff in Vacation-Text einsetzen

Liebe Kollegen, liebe Freunde,

mein Email-Programm hat die Nachricht betreffs "`<SUBJECT>`" empfangen. Ich werde sie lesen, sobald ich zurueck im Buero bin, d.h. am Montag, den 30. Oktober 2006, und werde sobald wie moeglich reagieren.

Mit freundlichen Gruessen,  
Jan Bredereke

## Betreff in Vacation-Text einsetzen: Ziel

Liebe Kollegen, liebe Freunde,

mein Email-Programm hat die Nachricht betreffs "`Re: Forschungsantrag`" empfangen. Ich werde sie lesen, sobald ich zurueck im Buero bin, d.h. am Montag, den 30. Oktober 2006, und werde sobald wie moeglich reagieren.

Mit freundlichen Gruessen,  
Jan Bredereke

Demo



## Betreff in Vacation-Text einsetzen: Lösung

vacation-msg.sh:

```
#!/bin/sh
# (In Wirklichkeit muss ${subject} von "procmail" gesetzt werden.)
subject='Re: Forschungsantrag'
sed -e "s/<SUBJECT>/${subject}/g" \  
    < vacation-msg-tmpl.txt > vacation-msg-filled.txt
```

## Festes Präfix in einer Dateiliste entfernen

```
sed/prefix-files.txt
sed/prefix-noprefix.txt
sed/prefix.sh
sed/vacation-msg-filled.txt
sed/vacation-msg-tmpl.txt
sed/vacation-msg.sh
sed/xterm.sh
```

## Festes Präfix in einer Dateiliste entfernen: Ziel

```
prefix-files.txt
prefix-noprefix.txt
prefix.sh
vacation-msg-filled.txt
vacation-msg-tmpl.txt
vacation-msg.sh
xterm.sh
```

Demo



## Festes Präfix in einer Dateiliste entfernen: Lösung

prefix.sh (bzw. direkt eintippen):

```
#!/bin/sh
sed -e 's!sed/!!' \
  < prefix-files.txt > prefix-noprefix.txt
```

## Die Grundkommandos von sed

- **Syntax:**  
`sed [-e Editierkommando] [-f Scriptdatei] [-n] [Datei ...]`
- **Editierkommando:**  
*Adresse Funktion*
- **Funktion:**  
`s /Ausdruck/Ersetzung/[g][p]` (substitute) → ersetzt auf Ausdruck passenden Text  
`d` (delete) → löscht aktuelle Zeile, nichts wird gedruckt  
`!` (not) → führt Funktion aus, falls Adresse nicht paßt
- **Adresse:**  
(leer)  
*/Ausdruck/*
- **Ausdruck:**  
ein regulärer Ausdruck wie bei grep
- **Modus:**  
`g` (global) → ersetzt alle passenden Texte, nicht nur den ersten  
`p` (print) → falls Ersetzung stattgefunden hat, drucke Zeile (trotz Option `-n`)

## Beispiele zum Üben

1. ersetze alle (kleinen) Umlaute
  - „ä“ → „ae“, „ö“ → „oe“, „ü“ → „ue“, „ß“ → „ss“
2. dasselbe, aber drucke nur die Zeilen mit Ersetzungen (zur Kontrolle)
3. in allen Zeilen in denen „input“ vorkommt, ersetze „.tex“ durch „“
4. dasselbe, mit Kontrolle

Demo

## 3. Der Textstrom-Editor sed

- 3.1 Grundprinzip eines Textstrom-Editors
- 3.2 Reguläre Ausdrücke in sed

## Typische Anwendungen von regulären Ausdrücken

- variable Textanteile erkennen
  - variables Präfix in einer Dateiliste entfernen
- interessante Textanteile extrahieren
  - Liste aller Benutzer aus /etc/passwd extrahieren
  - Liste aller „echten“ Benutzer aus /etc/passwd extrahieren

## Variables Präfix in einer Dateiliste entfernen

```
./article/implicit-concepts.tex
./article/fi-example-sli.tex
./article/circumvent.tex
./article/blocked.tex
./article/maint-tel-req.tex
./article/tina-concepts.tex
./dagstuhl-talk.tex
./dagstuhl.tex
```

## Variables Präfix in einer Dateiliste entfernen: Ziel

```
implicit-concepts.tex
fi-example-sli.tex
circumvent.tex
blocked.tex
maint-tel-req.tex
tina-concepts.tex
dagstuhl-talk.tex
dagstuhl.tex
```

Demo



## Variables Präfix in einer Dateiliste entfernen: Lösung

varprefix.sh:

```
#!/bin/sh
sed -e 's!^.*!/!!' \  
  < varprefix-files.txt > varprefix-noprefix.txt
```

## Liste aller Benutzer aus /etc/passwd extrahieren

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
wwwrun:x:30:65534:Daemon user for apache:/var/lib/wwwrun:/bin/bash
named:x:44:44:Nameserver Daemon:/var/named:/bin/bash
nobody:x:65534:65534:nobody:/var/lib/nobody:/bin/bash
brederek:x:500:100:Jan Brederke:/home/brederek:/bin/bash
mueller:x:501:100:B. Mueller:/home/mueller:/bin/bash
meyer:x:502:100:L. Meyer:/home/meyer:/bin/bash
schulze:x:503:100:G. Schulze:/home/schulze:/bin/bash
```

## Liste aller Benutzer aus /etc/passwd extrahieren: Ziel

```
root
bin
daemon
Daemon user for apache
Nameserver Daemon
nobody
Jan Brederke
B. Mueller
L. Meyer
G. Schulze
```

Demo



## Liste aller Benutzer aus /etc/passwd extrahieren: Lösung

passwd-names.sh:

```
#!/bin/sh
sed \
  -e 's/^\([^:]*:\)\{4\}\([^:]*\):.*$/2/' \
  < passwd > passwd-names.txt
```

(Schritt für Schritt entwickeln)

## Liste aller „echten“ Benutzer aus /etc/passwd extrahieren: Ziel

```
Jan Brederke
B. Mueller
L. Meyer
G. Schulze
```

Demo



## Liste aller „echten“ Benutzer aus /etc/passwd extrahieren: Lösung

passwd-realnames.sh:

```
#!/bin/sh
sed -e '/^\([^:]*:\)\{2\}5[0-9][0-9]:.*!d' \
  -e 's/^\([^:]*:\)\{4\}\([^:]*\):.*$/2/' \
  < passwd > passwd-realnames.txt
```

(Ersetzung bleibt gleich,  
nur weiteres Kommando zum Filtern vorweg)

## Reguläre Ausdrücke von sed/grep (1)

- **Ausdruck:**  
Folge von *Mustern*, optional mit *Zwischenraummarkierungen* dazwischen
- **Muster:**  
ein einzelnes *Zeichen* oder eine *Gruppierung*
- **Zeichen:**

<b>c</b>	(Buchstabe)	paßt auf sich selbst
<b>.</b>	(Punkt)	paßt auf ein beliebiges Zeichen
<b>\c</b>	(Backslash)	Sonderzeichen → normales Zeichen
<b>[Buchstaben]</b>	(Buchstabenmenge)	paßt auf eines der Zeichen
<b>[^Buchstaben]</b>	(Komplementmenge)	paßt auf ein nicht aufgeführtes Zeichen
- **Buchstaben:**  
können auch Bereiche *von–bis* sein,  
„[“, „]“ und „–“ müssen ganz vorne oder hinten stehen

## Reguläre Ausdrücke von sed/grep (2)

- **Gruppierung:**

<code>Muster*</code>	Folge von 0 oder mehr Mustern
<code>Muster?</code>	0 oder einmal das Muster
<code>Muster+</code>	Folge von 1 oder mehr Mustern
<code>Muster{n}</code>	Folge von $n$ Mustern
<code>Muster{n,m}</code>	Folge von $n$ bis $m$ Mustern
<code>\(Ausdruck\)</code>	faßt Ausdruck zusammen; markiert Text für $\backslash N$
<code>\N</code>	das $N$ -te mit Klammern eingeschlossene Muster
<code>Muster  Muster</code>	entweder das erste oder das zweite Muster

- **Zwischenraummarkierung:**

<code>^</code>	Zeilenanfang
<code>\$</code>	Zeilenende
<code>\&lt;</code>	Wortanfang
<code>\&gt;</code>	Wortende

## Beispiele zum Üben

1. Erkennen von Email-Adressen
2. Extraktion von include-Dateinamen aus einer LaTeX-Quelle
3. Extraktion aller derzeit aktiven Benutzer aus „w“-Ausgabe
4. Schützen von Sonderzeichen in der Vacation-Text-Ersetzung

## Erkennen von Email-Adressen

- aus dem Eingabestrom sollen *nur* meine Email-Adressen gedruckt werden, sonst nichts

```
brederek@tzi.de
brederek@informatik.uni-bremen.de
brederek@gemini.informatik.uni-bremen.de
brederek@saturn.informatik.uni-bremen.de
jan.bredereke@nwn.de
```

- jeder schreibt seine Lösung erstmal auf Papier

Demo

## Extraktion von include-Dateinamen aus einer LaTeX-Quelle

- LaTeX-Quelldatei enthält Zeilen der Art `\input{intro.tex}`
- Ausgabe:  
Liste der Dateinamen in diesen Kommandos
  - ignorieren: Mit „%“ auskommentierte LaTeX-Befehle

Demo

## Extraktion aller derzeit aktiven Benutzer aus „w“-Ausgabe

```
12:24pm up 56 day(s), 2:42, 10 users, load average: 0.00, 0.02, 0.03
User      tty      login@  idle  JCPU  PCPU  what
alone    smb/0    12:24pm          /usr/local/lib/samba/smbd -D
nobody   smb/1    12:24pm          /usr/local/lib/samba/smbd -D
roefeer  smb/2    12:14pm          /usr/local/lib/samba/smbd -D
root     pts/1    1Aug01 2days 20:31  2    /usr/local/bin/bash -login
roefeer  smb/4    12:18pm          /usr/local/lib/samba/smbd -D
mawe     pts/2    1Aug01 14    6:06  25    contool
root     pts/3    1Aug01 13    9:29  5     /usr/local/bin/bash -login
cx1      pts/4    Tue 5pm 1:23  35    xemacs
cx1      pts/5    Thu 9am 5days 10:22  -bash
brederek pts/6    11:19am          w
```

## Extraktion aller derzeit aktiven Benutzer aus „w“-Ausgabe (2)

- gewünscht:
  - Liste von Benutzernamen
  - keine Benutzer mit „idle“-Zeit
  - keine Samba-Daemons
    - (d.h. nur Benutzer an Pseudo-TTYs)
  - keine Kopfzeile des w-Kommandos

Demo

## Schützen von Sonderzeichen in der Vacation-Text-Ersetzung

- Problem: Betreff könnte „/“ enthalten
- Lösungsidee: füge vor alle „/“ im Betreff Backslashes ein
  - auch vor alle „\“

Demo

## Inhalte der Vorlesung

1. Einführung
2. Lexikalische Analyse
3. Der Textstrom-Editor sed
- 4. Der Scanner-Generator lex
5. Syntaxanalyse und der Parser-Generator yacc
6. Syntaxgesteuerte Übersetzung
7. Übersetzungssteuerung mit make