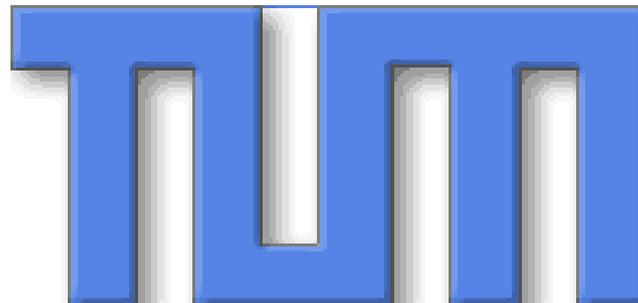


Der vi Editor

Vortrag im Rahmen des Proseminars Unix Tools
an der Technischen Universität München
im Sommersemester 2005

Vladislav Lazarov

`lazarov@in.tum.de`



3.05.2005

- vi?! Was ist dass denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

- vi?! Was ist das denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

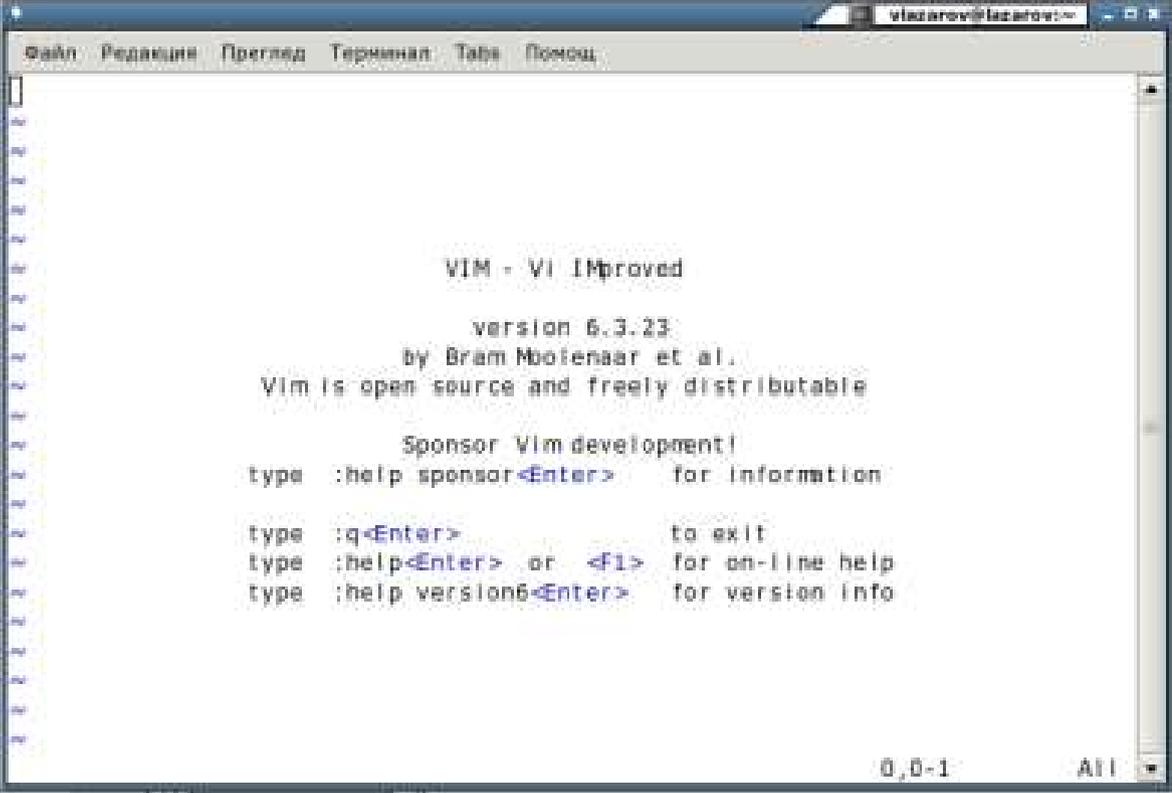
- vi?! Was ist dass denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

- vi?! Was ist das denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

- vi?! Was ist das denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

- vi?! Was ist das denn?
- Vor- und Nachteile
- Die ersten Schritte
- Die Befehle:
 - Cursor Bewegen
 - Text einfügen/löschen
 - Search and Replace
- Ex: Was steckt hinter *vi*?
- Emacs vs. *vi*

- vi (*visual* editor)
- ein (nicht so) einfacher Texteditor



The screenshot shows a terminal window titled 'vlazarov@lazarov:~'. The menu bar contains 'Файл', 'Редакция', 'Просмотр', 'Терминал', 'Табы', and 'Помощь'. The main content is the Vim startup screen for version 6.3.23, which includes the text: 'VIM - Vi Improved', 'version 6.3.23', 'by Bram Moolenaar et al.', 'Vim is open source and freely distributable', and 'Sponsor Vim development!'. It also lists commands: ':help sponsor<Enter>' for information, ':q<Enter>' to exit, ':help<Enter>' or '<F1>' for on-line help, and ':help version6<Enter>' for version info. The status bar at the bottom right shows '0,0-1' and 'All'.

```
VIM - Vi Improved
      version 6.3.23
    by Bram Moolenaar et al.
Vim is open source and freely distributable

Sponsor Vim development!
type  :help sponsor<Enter>  for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version6<Enter> for version info

0,0-1 All
```

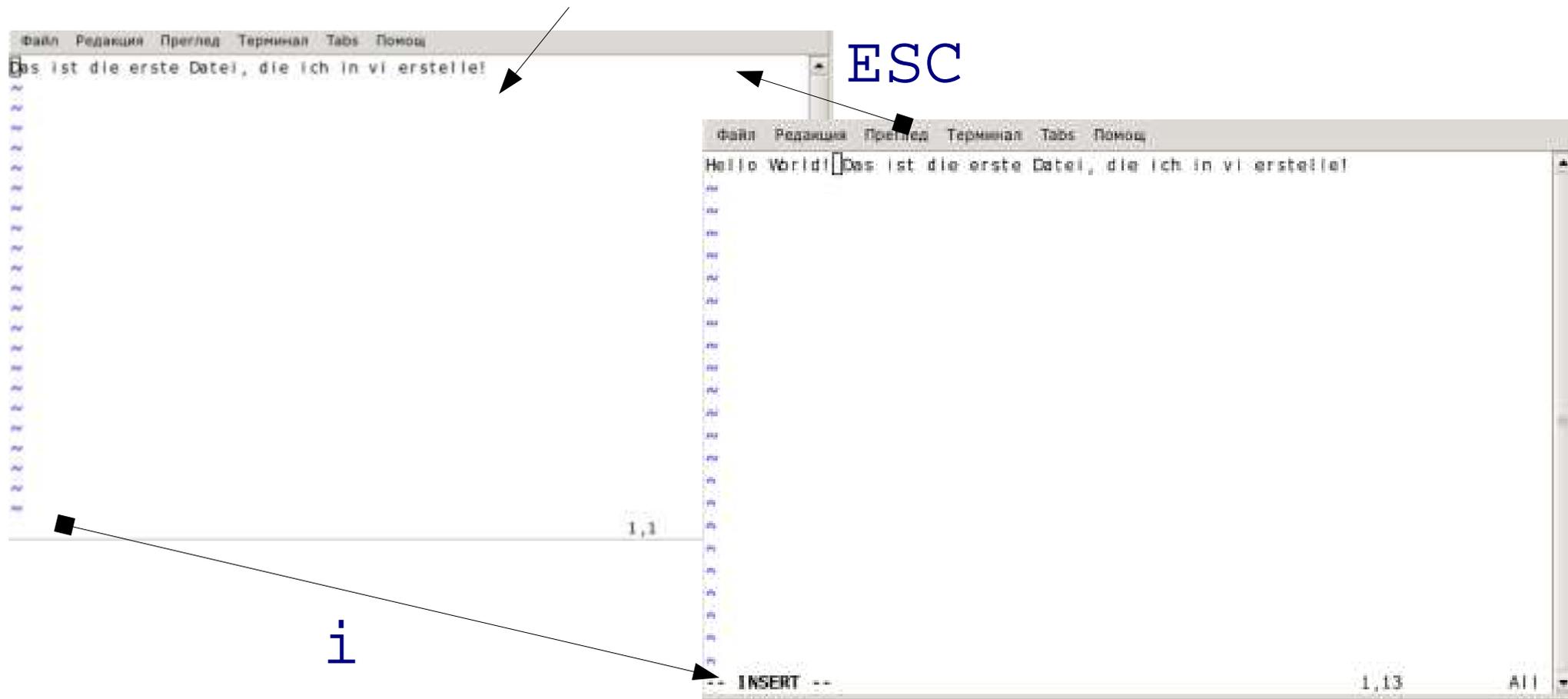
Vorteile:

- identisch auf UNIX Systemen
- vorinstalliert mit fast allen UNIX Systemen
- Finger bleiben immer auf der Tastatur
- kurze Ladezeiten
- benötigt kein X11
- braucht weniger Ressourcen als Emacs ;-)

Nachteile:

- kein WYSISWYG Editor
- nicht sehr intuitiv für Anfänger
- zu viele Befehle (die aber recht einfach sind)

Aufrufen: `vi dateiname`



- vi hat zwei verschiedene Modi:
 - command mode
 - insert/input mode

- Die meisten vi Befehle:
 - sind case-sensitive
 - werden nicht auf den Bildschirm angezeigt
 - brauchen die Enter-Taste nicht um ausgeführt zu werden
- Sie bestehen aus eine oder zwei Buchstaben (Beispiel: `i` oder `cw`)
- Ausnahme: Befehle die mit `/` , `?` oder `:` beginnen.
- **Allgemeines Befehlsformat:**
[Zahl] Befehl [Zahl] [Bewegung]

i	insert mode
ESC	command mode
ZZ	speichern und vi verlassen
:wq	das gleiche
:w	nur speichern; :w! überschreiben; :w <i>neuedatei</i>
:q	vi verlassen
:q!	verlassen von vi ohne Speicherung der Änderungen
:e <i>datei</i>	öffnen einer Datei

<code>h(←)</code>	<code>j(↓)</code>	<code>k(↑)</code>	<code>l(→)</code>	links, rechts, oben, unten (Beispiel: 4h)
<code>w</code>	<code>W</code>			nächstes Wort
<code>b</code>	<code>B</code>			Beginn des Worts
<code>e</code>	<code>E</code>			Ende des Worts
<code>)</code>	<code>(</code>			Nächster/vorheriger Satz
<code>}</code>	<code>{</code>			Nächster/vorheriger Absatz
<code>(line)G</code>				Sprung zur Zeile line (1G / G – Ende)
<code>^</code>				Beginn der Zeile
<code>\$</code>				Ende der Zeile

- Man kann Markierungen setzen, um später schnell bestimmte Stellen wiederzufinden
- Bestimmte Stelle markieren: `mx` , wobei „x“ eine beliebige Buchstabe
- Springen zur Markierung: `'x`

a, A	Text nach aktuelle Position einfügen
c, C, r	Text austauschen (Beispiel: c\$text)
d, dd, D, x	Text löschen (Beispiel: dw)
y, yy, p	Text kopieren/einfügen
J	Zeilen verbinden
.	Wiederholung des letzten Befehls
u	Letzten Befehl rückgängig machen

<code>/string</code>	Einen bestimmten String finden (vorwärts)
<code>?string</code>	Dasselbe in die andere Richtung
<code>n, N</code>	Wiederholung des Suchvorgangs
<code>:s</code>	Ein bestimmtes Muster austauschen (Beispiel: <code>:s/muster/string</code>)

Man kann *reguläre Ausdrücke* beim Suchen benutzen

vi hat einen Puffer, der alle Änderungen speichert. Aufrufen kann man es mit

```
vi -r
```

```
vlazarov@r040144:~> vi -r
```

```
Swap files found:
```

```
  In current directory:
```

```
1.  .datei.swp
```

```
    owned by: vlazarov   dated: Sun May  1 15:03:56 2005
```

```
    file name: ~/vlazarov/datei
```

```
    modified: YES
```

```
    user name: vlazarov  host name: r040144
```

```
    process ID: 8179
```

```
  In directory ~/tmp:
```

```
    -- none --
```

```
  In directory /var/tmp:
```

```
    -- none --
```

```
  In directory /tmp:
```

```
    -- none --
```

```
vlazarov@r040144:~> vi -r datei
```

```
Using swap file ".datei.swp"
```

```
"~/datei" [New File]
```

```
Recovery completed. You should check if everything is OK.  
(You might want to write out this file under another name  
and run diff with the original file to check for changes)  
Delete the .swp file afterwards.
```

```
Hit ENTER or type command to continue
```

vi ist oft als „Screen Editor“ bezeichnet.
Eigentlich steckt hinter *vi* der Zeileneditor:
Ex.

- Alle *Ex* Befehle beginnen mit „:“
- *Ex* Befehle sind vor allem nützlich um Zeileneditierungen oder globale Änderungen vorzunehmen.

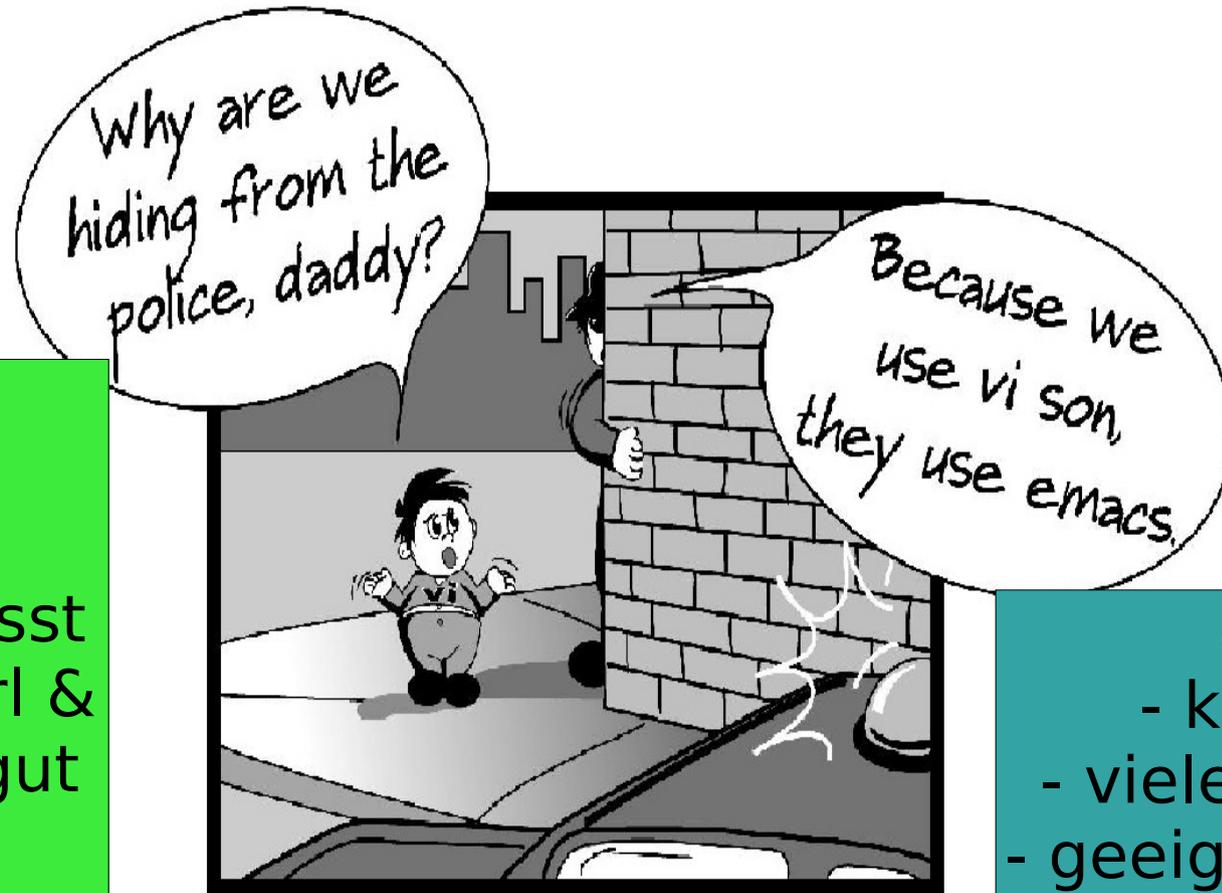
<code>:mline</code>	verschieben der Zeile
<code>:co, :t</code>	kopieren der Zeile
<code>:. </code>	löschen vom Cursor-Position bis Ende der Zeile
<code>:set option</code>	Einstellungen ändern
<code>:read datei</code>	andere Datei in die aktuelle einfügen
<code>:w neuedatei</code>	als neue Datei speichern
<code>:e datei</code>	Datei öffnen

– Befehle „global“ ausführen:

`:[address]g[!]/pattern/[commands]`

– Ausgabe eines Unix-Kommandos in den Text einzufügen. (Beispiel: `:r !date`)

- *vi* kein „open-source“ Projekt
- Versionen für andere Betriebssysteme
- am häufigsten verbreitete *vi Clones*:
 - *nvi*
 - *vim*
 - *elvis*
 - *vile* u.a.
- GUIs, Multiwindow Editing, andere (nützliche) *reguläre Ausdrücke* u.v.m.



vi

- Modi
- schnell
- man vergisst wozu die ctrl & alt Tasten gut sind ;-)

emacs

- keine Modi
- viele „Shortcuts“
- geeignet für TeXen
- langsam

„**vi is perfect:** VI in roman numerals is 6. The natural numbers less than 6 are 1, 2 and 3. $1+2+3=6$. So 6 is a perfect number. Therefore *vi* is perfect.“

- *Arthur Tateishi*

Danke für die
Aufmerksamkeit!



- Learning the vi Editor
Sixth Edition
Linda Lamb & Arnold Robbins
Publisher: O'Reilly & Associates
- Wikipedia
<http://en.wikipedia.org/wiki/Vi>
- The Vi Lover's Homepage
<http://thomer.com/vi/vi.html>
- <http://www.io.com/~dierdorf/emacsvi.html>