Unix I – Einführung in das Unix Betriebssystem

Stephan Thesing

VI. Vorlesung: vi

- **Editoren**
- **`vi`: spartanisch aber schnell**

Editoren dienen dazu, Textfiles zu bearbeiten.

Es gibt verschiedene Konzepte von Editoren:

- Zeilenorientiert. Nur eine einzelne Zeile kann jeweils editiert werden.
- Bildschirmorientiert. Im File kann beliebig herumgewandert werden, mehrere Zeilen werden angezeigt.

Unter Unix werden hauptsächlich zwei Editoren benutzt:

- `vi`: Alt, aus dem Zeileneditor `ed` hervorgegangen.
- `emacs`: Auch alt, aber bildschirmorientiert, enorm erweiterbar

Heute: `vi` besprechen

`vi` ist aus dem Zeileneditor `ex` hervorgegangen.

Bedienung gewöhnungsbedürftig:

- Es gibt einen *Kommandomodus*, in welchem Kommandos eingegeben werden
- es gibt den Einfügemodus, in welchem Text eingegeben wird

Man muss explizit von einem Modus in den anderen wechseln.

Vorteile des `vi`:

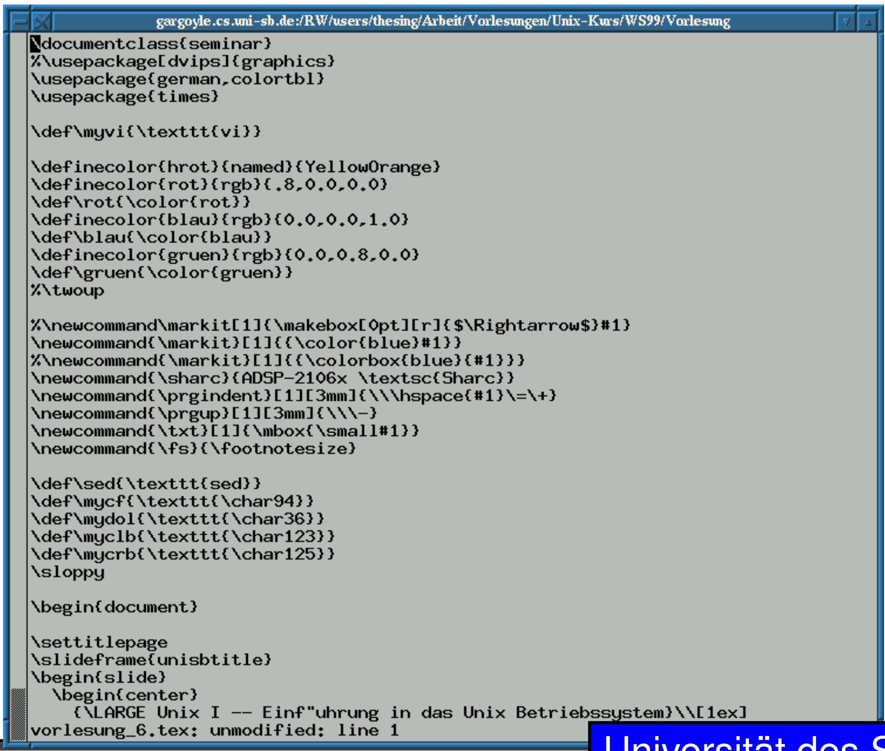
- Überall verfügbar
- Klein
- Schnell

Programm für heute:

- Aufruf von `vi`
- Cursor bewegen
- Text einfügen/ändern
- Text löschen
- Suchen
- Text verschieben
- Sonstiges: Variablen et.al.

Aufruf von `vi`

`vi` wird durch `vi` *`File(s)`* aufgerufen:



gargoyle.cs.uni-sb.de:/RW/users/thesing/Arbeit/Vorlesungen/Unix-Kurs/WS99/Vorlesung

```
\documentclass{seminar}
%\usepackage[dvips]{graphics}
\usepackage{german,colortbl}
\usepackage{times}

\def\myvi{\texttt{vi}}

\definecolor{hrot}{named}{YellowOrange}
\definecolor{rot}{rgb}{.8,0.0,0.0}
\def\rot{\color{rot}}
\definecolor{blau}{rgb}{0.0,0.0,1.0}
\def\blau{\color{blau}}
\definecolor{gruen}{rgb}{0.0,0.8,0.0}
\def\gruen{\color{gruen}}
%\twoup

%\newcommand\markit[1]{\makebox[0pt][r]{$\Rightarrow$}#1}
\newcommand{\markit}[1]{{\color{blue}#1}}
%\newcommand{\markit}[1]{{\colorbox{blue}{#1}}}
\newcommand{\sharc}{ADSP-2106x \textsc{Sharc}}
\newcommand{\prgindent}[1][3mm]{\\\hspace{#1}\=\+}
\newcommand{\prgup}[1][3mm]{\\\-}
\newcommand{\txt}[1]{\mbox{\small#1}}
\newcommand{\fs}{\footnotesize}

\def\sed{\texttt{sed}}
\def\mycf{\texttt{\char94}}
\def\mydol{\texttt{\char36}}
\def\myclb{\texttt{\char123}}
\def\mycrb{\texttt{\char125}}
\sloppy

\begin{document}

\settitlepage
\slideframe{unisbtitle}
\begin{slide}
  \begin{center}
    {\LARGE Unix I -- Einf"uhrung in das Unix Betriebssystem}\\[1ex]
vorlesung_6.tex: unmodified: line 1
```


Nach dem Aufruf ist man im Kommandomodus.

Beenden kann man `vi`, wenn man im Kommandomodus

- `ZZ` eingibt, oder
- `:q` eingibt (ohne Speichern), oder
- `:x` eingibt

Um vom Texteingabemodus in den Kommandomodus zu kommen gibt man `<ESC>` ein.

Bewegen des Cursors

Einfache Bewegungen:

- Links: `<BS>` oder `h` oder ←
- Runter: CTRL-`N` oder `j` oder ↓
- Hoch: CTRL-`P` oder `k` oder ↑
- Rechts: SPACE oder `l` oder →

Man kann beim `vi` (fast) jedem Kommando einen *Zähler* vorstellen.

So läuft etwa der Cursor nach Eingabe von `10`CTRL-`P` zehn Zeilen nach oben!

Allerdings sieht man die Eingabe des Zählers nicht auf dem Bildschirm:-(

Andere Bewegungen:

- `-` bewegt den Cursor zum Anfang der vorigen Zeile
- `+` bewegt ihn zum Anfang der nächsten Zeile
- `$` springt zum Ende der aktuellen Zeile
- `^` springt zum Anfang der aktuellen Zeile
- `|` springt zur Spalte, die durch den Zähler angegeben wird (1)
- `w` springt zum nächsten Wort
- `b` springt zum vorigen Wort
- `e` springt zum Ende des akt. Wortes
- `G` springt zu einer Zeile: Zähler gibt Zeilennummer. Ohne Zähler: letzte Zeile. (`1G` springt also zur ersten Zeile!)

Scrolling:

- CTRL-`D` scrollt *Zähler* Zeilen runter, default: 1/2 Screen
- CTRL-`U` scrollt entsprechend hoch
- CTRL-`F` scrollt zur nächsten Seite
- CTRL-`B` scrollt eine Seite zurück
- `%` Bewegt den Cursor zur nächsten passenden Klammer (i.e. `(){}`)
- `H` bewegt den Cursor zum Anfang der obersten Bildschirmzeile
- `L` entsprechend zum Anfang der untersten Zeile

Marken dienen im `vi` dazu, Positionen zu kennzeichnen.

Man kann mehrere Marken (26 insgesamt) definieren:

- `m`*a-z* markiert die akt. Position und speichert sie in der Marke. Bspe: `ma` oder `mh`
- `'`*Marke* springt zum Anfang der mit *Marke* markierten Zeile
- `` ` ``*Marke* springt zur exakten Position der Marke (z.B. `` `a ``)

Text einfügen/ändern

Um im `vi` Text einfügen/ändern zu können, muss man den Kommandomodus verlassen und in den Einfügemodus gehen.

Zum Schluss kommt man mit ESC wieder in den Kommandomodus.

Es gibt folgende Kommandos, um in den Einfügemodus zu kommen:

- `a` Fügt hinter dem Cursor den Text ein
- `A` Fügt Text am Ende der akt. Zeile ein
- `i` Fügt Text vor dem Cursor ein
- `I` Fügt Text am Anfang der Zeile ein
- `o` Fügt Text nach der Zeile ein, generiert neue Zeile
- `O` Fügt Text vor der Zeile ein, generiert neue Zeile

Um Text zu ändern gibt es diese Kommandos:

- `r`*Zeichen* Ersetzt das Zeichen unter dem Cursor durch *Zeichen*
- `R` Ersetzt den Text ab Cursorposition durch den neuen Text (overwrite mode).

Das Kommando `.` wiederholt das letzte Kommando, das Text änderte.

Beispiele: `aFOOBAR`ESC fügt den Text `FOOBAR` hinter dem Cursor ein.

`oNIXDA`ESC fügt eine neue Zeile des Inhaltes `NIXDA` nach der akt. Zeile ein.

`rx` ersetzt das Zeichen unter dem Cursor durch `x`

`RSo ein Unsinn`ESC überschreibt den Inhalt ab dem Cursor durch `So ein Unsinn`.

Text löschen

Im `vi` gibt es vier Kommandos, die Text löschen:

- `x` löscht das Zeichen ab dem Cursor (oder *Zähler* Zeichen)
- `X` löscht vor dem Cursor
- `D` löscht den Rest der Zeile ab Cursor
- Das `d` Kommando kann verschiedene Dinge löschen:
  - `dw` löscht das akt. Wort
  - `dd` die aktuelle Zeile
  - ...

Suchen

Im `vi` kann man auch im Text suchen.

Dabei kann man in der akt. Zeile suchen, oder im ganzen Text, je nach Kommando.

- `f`*Zeichen* sucht in der Zeile nach *Zeichen*: `fe` sucht etwa das nächste `e`
- `F` sucht entsprechend rückwärts
- `;` wiederholt das letzte `f` oder `F` Kommando
- `/`*String*`/`NL sucht im File nach dem String.
- `?`*String*`?`NL sucht rückwärts.
- `n` wiederholt die letzte Suche
- `N` wiederholt sie in der anderen Richtung

Sonstige Kommandos

- CTRL-`L` zeichnet den Bildschirm neu
- `u` macht die letzte Änderung rückgängig
- `.` wiederholt die letzte Textveränderung
- `J` hängt die nächste Zeile an die aktuelle Zeile an
- CTRL-`V` fügt das nächste Zeichen ”as-is” ein

`vi` ist aus dem `ed` Editor hervorgegangen.

Im `vi` kann man `ed` Kommandos geben, indem man ein **:** eingibt, gefolgt von den Kommandos.

Z.B. schreibt `:w foobar`RETURN den Text in das File `foobar`.

- `:w[!]`*File* Schreibt den Text in das *File*. `!` schreibt in bereits bestehendes File.
- `:q[!]` Beendet den `vi`.
- `:e[!]` [*File*] Neues (oder altes) File editieren.
- `:n` Nächstes Files editieren.
- `:r` *File* File einfügen.
- `:r !`*Command* Führt *Command* aus und fügt dessen Ausgabe ein.
- `:!`*Command* In einer Subshell *Command* ausführen.

Optionen, Variablen

`vi` liest beim Start das File `$HOME/.exrc`.

In diesem kann man Optionen für `vi` setzen (Variablen), etwa

```
set ignorecase
set autoindent
set shiftwidth=4
```

`vi` kennt eine ganze Reihe von Variablen (können auch über `:set` gesetzt werden)

Man setzt eine Variable durch `set` *Name*[=]*Wert*

Man löscht sie durch `set no`*Name*:

```
set notimeout
set recdir=/tmp
```


Tutorial

Unter `~thesing/Unix-Kurs/Uebungen/V` findet man einige Tutorials.

Aufruf durch

`~thesing/Unix-Kurs/Uebungen/V/vi.tut.csh beginner`

oder

`~thesing/Unix-Kurs/Uebungen/V/vi.tut.csh advanced`

Unter `~thesing/unix-Kurs/Uebungen/V/vi.ref.ps` steht ein anderes Tutorial als Postscript Datei (anzuzeigen mittels `gv vi.ref.ps`).