

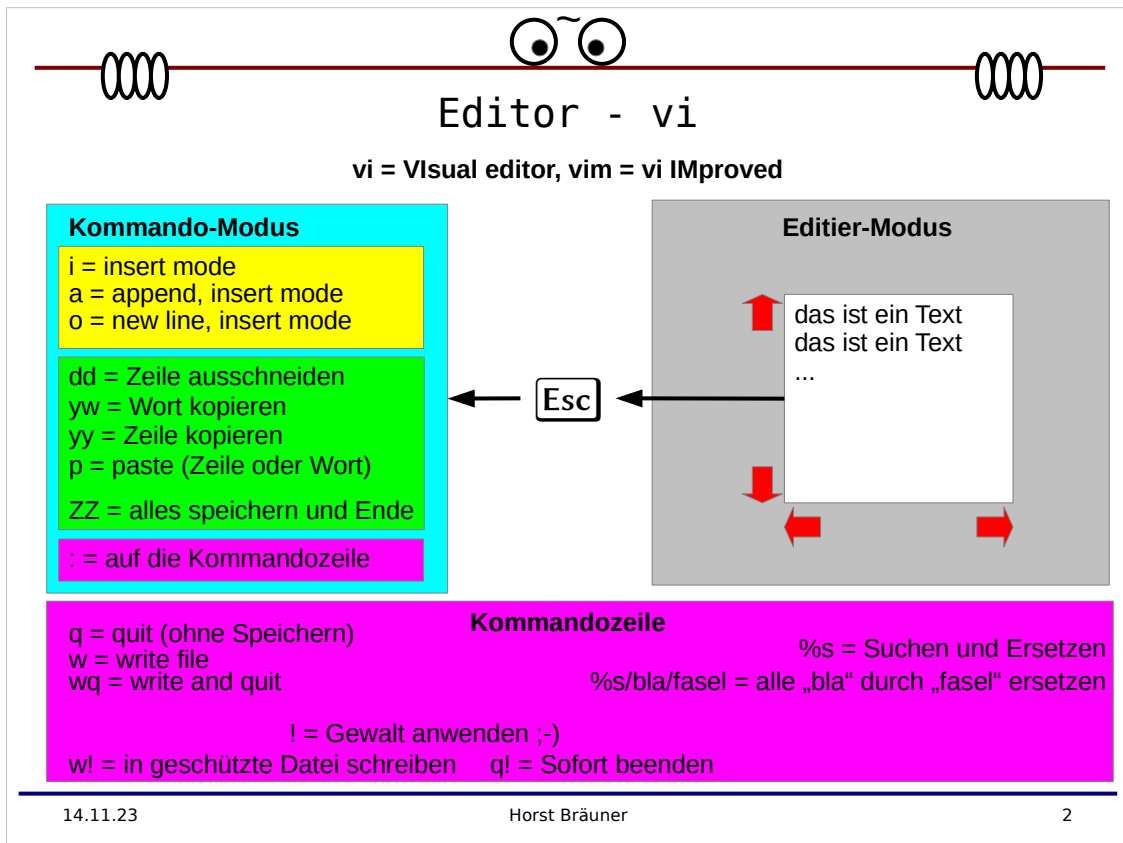
Praktische Datenverarbeitung

- Der Editor vi
 - apt-get install vim-nox

Der Editor vi

Aufwand 1 UE

vi ist ein Editor, der auf jedem UNIX/Linux-System vorhanden ist. Wenn Sie mit vi ein wenig umgehen können, können Sie nahezu jedes unixoide System rudimentär administrieren.




vi hat 3 (Grund-)Modi:

im **Kommando-Modus** (cyan) ... können Sie mit Tastaturbefehlen **Aktionen** auslösen (grün) oder vom Kommando- **in den Editiermodus wechseln** (gelb) oder **in die Kommandozeile wechseln** (magenta).

im **Editier-Modus** (grau) ... können Sie Text eingeben und sich wie in anderen, einfachen Editoren bewegen. Hinweis: Ältere Versionen von vi unterstützen nur teilweise die „Pfeil“- und „Bild“-Tasten. mit der Taste „ESC“ verlassen Sie den Editiermodus und gelangen in den Kommando-Modus.

in der **Kommandozeile** (magenta) ... können Sie den Editor verlassen, Ihre Eingaben speichern, suchen und ersetzen. Durch das anfügen eines „!“ an einen Befehl, verleihen Sie diesem Nachdruck :-)) und können, sofern Sie die Berechtigung besitzen, Dateiattribute wie „read only“ übergehen. Mit der Taste „ESC“ verlassen Sie den Kommandozeilen-Modus und gelangen in den Kommando-Modus.

siehe auch <https://de.wikipedia.org/wiki/Vi>



Häufig verwendete Kommandos

<code>vi / vim / vi [Datei]</code>	vi starten
<code>i / a / o</code>	in den Eingabemodus wechseln (<code>insert</code> / <code>append</code> / <code>neue Zeile</code>)
<code>Esc</code>	in den Kommandomodus zurück wechseln
<code>:</code>	in die Kommandozeile wechseln
<code>:q / :q!</code>	<code>beenden</code> / <code>beenden ohne Speichern</code>
<code>:w / :wq</code> oder <code>ZZ</code>	<code>Speichern</code> / <code>Speichern und beenden</code>
<code>:%s/alles das hier/durch das</code>	<code>Suchen</code> und <code>Muster1</code> durch <code>Muster2</code> ersetzen
<code>:e [Datei]</code>	eine Datei <code>öffnen</code>
<code>yb / yw / yy</code>	Wort <code>links</code> / <code>rechts</code> / <code>ganze Zeile</code> in Puffer schreiben
<code>db / dw / dd</code>	Wort <code>links</code> / <code>rechts</code> / <code>ganze Zeile</code> löschen
<code>P / p</code>	Puffer vor / hinter dem aktuellen Wort/Zeile einfügen


14.11.23

Horst Bräuner

3

Zum Üben empfehle ich Ihnen, kleine Dateien anzulegen und mit den Kommandos „zu spielen“. vi wird wahrscheinlich nicht Ihr Lieblingstexteditor werden, aber er ist auf jedem unixoiden System vorhanden. Daher sollten Sie zumindest die einfachen Kommandos einmal ausprobieren.

Normalerweise bearbeiten Sie mit vi Konfigurationsdateien, Skripte, kleine HTML/PHP-Dateien und einfache Texte. Mit vi könnten Sie dennoch grundsätzlich alle Arten von Dateien bearbeiten. Ob es jedoch sinnvoll ist, Musikdateien, Bilder oder Binärdaten damit zu bearbeiten, überlasse ich Ihnen.


vim - „bunt“

```
moritz@depp2:~$  
moritz@depp2:~$  
moritz@depp2:~$ pwd  
/home/moritz  
moritz@depp2:~$  
moritz@depp2:~$ echo "syntax on" >> .vimrc  
moritz@depp2:~$  
moritz@depp2:~$ cat .vimrc  
syntax on  
moritz@depp2:~$
```

depp2-Klon [wird ausgeführt] - Oracle VM VirtualBox

```
root@depp2:~#  
root@depp2:~# pwd  
/root  
root@depp2:~#  
root@depp2:~# echo "syntax on" >> .vimrc
```

14.11.23

Horst Bräuner

4

Eine kurze Übung:

vi wird über die Konfigurationsdatei `.vimrc` (Punkt am Anfang nicht vergessen ;-)), die in Ihrem `$HOME`-Verzeichnis liegt angepasst. Eine sinnvolle Anpassung ist „syntax on“ dort einzufügen. Damit wird vi bzw. vim „bunt“ und zeigt Ihnen zum Beispiel in HTML-, PHP-, XML-, C- oder Konfigurationsdateien Anfang und Ende von Klammerungen `() [] {} <>` oder stellt Codeblöcke, Variablen und Statements farblich dar.

Probieren Sie vi mit verschiedenen Dateitypen aus.


bash - „bunt“



```
moritz@depp2:~$  
moritz@depp2:~$ pwd  
/home/moritz  
moritz@depp2:~$ vi .bashrc
```



```
# uncomment for a colored prompt, if the terminal has the capability; turned  
# off by default to not distract the user: the focus in a terminal window  
# should be on the output of commands, not on the prompt  
force_color_prompt=yes  
  
if [ -n "$force_color_prompt" ]; then  
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then  
        # We have color support; assume it's compliant with Ecma-48  
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such  
        # a case would tend to support setf rather than setaf.)  
        color_prompt=yes  
    else  
        color_prompt=  
    fi  
fi  
  
if [ "$color_prompt" = yes ]; then  
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '  
else  
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '  
fi  
unset color_prompt force_color_prompt
```

14.11.23

Horst Bräuner

5

Zum Abschluss noch eine Möglichkeit „bunt“ zu werden. Stellen Sie sich mit vi in der .bashrc-Datei in Ihrem \$HOME-Verzeichnis einen farbigen „prompt“ ein, siehe auch Vorlesung „Linux, Erste Schritte“.