Eine kurze Einführung in Vim

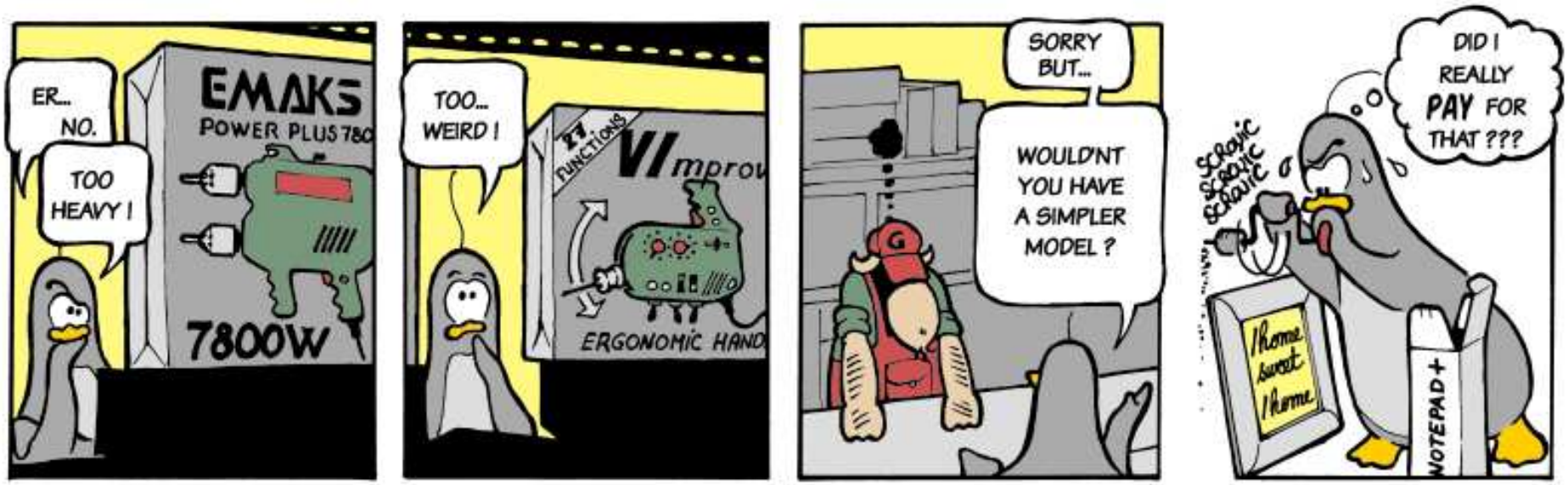
Johannes Englisch

Universität Leipzig
Institut für Linguistik

Seminar „Sprachwissenschaftliche Elementarkompetenz”

Plot

Vim und Emacs



(Quelle: [0xBABAF000L])

Warum Vim?

Vorteile

- viele Steuerbefehle sind mit nur einem Tastendruck erreichbar
- komplexere Befehle setzen sich logisch aus primitiveren Befehlen zusammen
- Arbeitserleichterungen wie das Erfassen von Klammerpaaren, Syntax highlighting und das Suchen mittels regulärer Ausdrücke sind fester Bestandteil von Vim
- Vim ist durch eine Masse an Plugins erweiterbar

Nachteile

- Vim braucht etwas Übung und Umgewöhnung
- die verschiedenen Modi können u. U. verwirrend sein – Stichwort ‘magical vim commands’

Der Vimtutor

der Vimtutor ist eine Textdatei, an der man die Steuerung von Vim lernen und üben kann

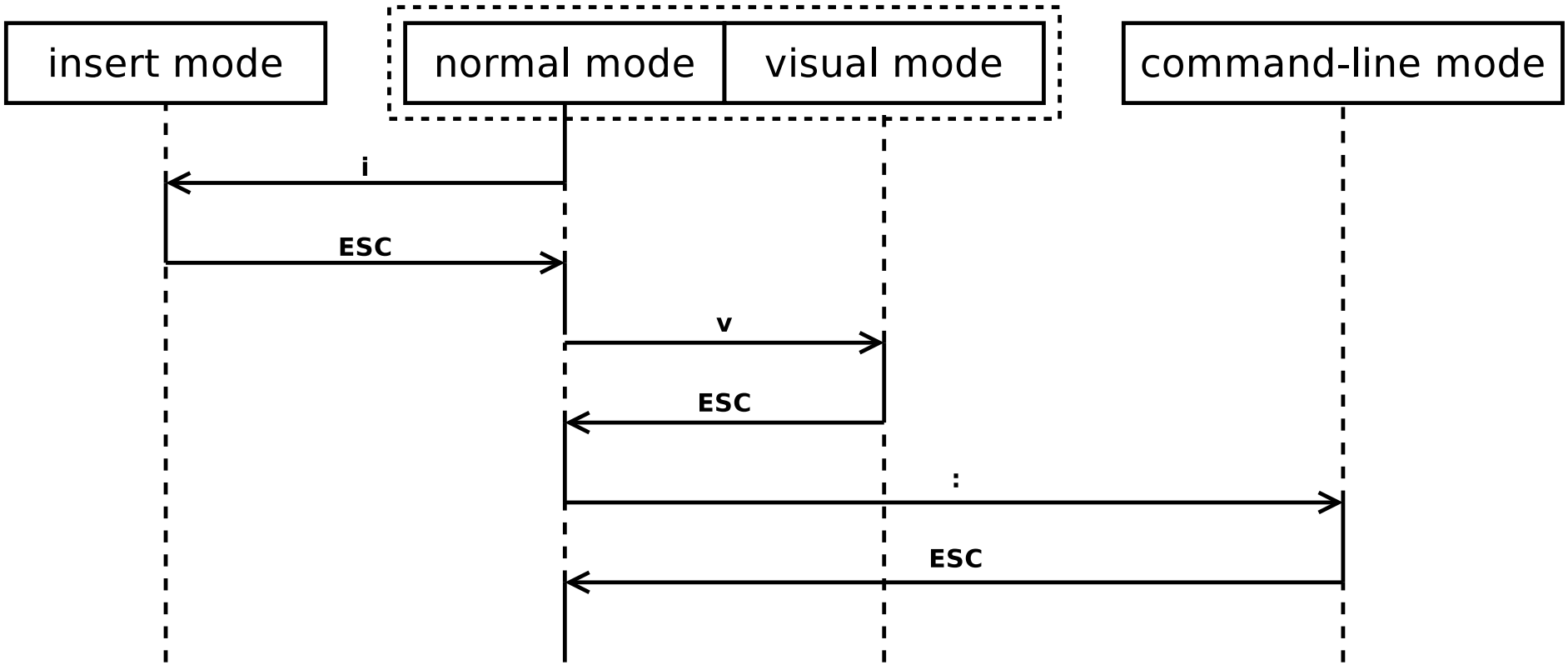
Starten im Windows

Startmenü $\longrightarrow$ Programme $\longrightarrow$ Vim 7.3 $\longrightarrow$ Vim tutor

Starten in Linux/Mac

- Terminal öffnen
- `vimtutor` eingeben und Enter drücken

Die Modi von Vim



(Graphik frei nach: [introduction])

Der normale Modus

- jeder Tastendruck ist ein Befehl
- hier wird der Cursor bewegt, kopiert, ausgeschnitten, eingefügt u. v. a. m.
- der Modus ist der Default und kann immer mit ESC erreicht werden

Der Einfügenmodus

- hier wird Text eingefügt
- funktioniert dann wie andere Editoren auch
- unten markiert mit `--INSERT--` oder `--EINFÜGEN--`

Der visuelle Modus

- Sonderfall des Kommandomodus
- Bewegungsbefehle markieren Text
- es gelten ansonsten dieselben Tastenbefehle
- unten markiert mit `--VISUAL--` oder `--VISUELL--`

Der Kommandozeilenmodus

- öffnet eine Kommandozeile, in der Befehle eingegeben werden können
- Befehle steuern meistens den Editor selbst (Dateien öffnen, speichern, schließen, Verwalten mehrerer Dateien usw.
- erkennbar am Doppelpunkt unten

Datei

Öffnen, Speichern und Schließen

:e [Datei]	**e**dit file	
:w [Datei]	**w**rite file	
:w! [Datei]	force **w**rite file	umgehe Schreibschutz
:q	**q**uit file	
:q!	force **q**uit file	schließe ohne Speichern
:qa	**q**uit **a**ll	
:qa!	force **q**uit **a**ll	
:wq	**w**rite & **q**uit	
:wq!	force **w**rite & **q**uit	
:wqa	**w**rite & **q**uit **a**ll	
:wqa!	force **w**rite & **q**uit **a**ll	

Steuerung I

Zeichen für Zeichen

k

h l

j

Steuerung I

Zeichen für Zeichen

k

h l

j

Hinweis

die Pfeiltasten gehen zwar standardmäßig auch, es lohnt sich aber, die HJKL-Steuerung zu lernen, da sie für eine leichtere Verwendung mit dem Zehnfingerschreibsystem konzipiert wurde

Steuerung II

weitere Bewegung

Einheit	zurück		vor	
Wort	b B	(*one word **b**ack*)	e E	(***e**nd of word*)
			w W	(*next **w**ord*)
Satz	(		)	
Zeile	^	(Zeilenanfang)	$	(Zeilenende)
	-	(Anfang letzte Zeile)	+	(Anfang nächster Zeile)
Klammer		% (zur passenden Klammer)		
Absatz	{	(Absatzanfang)	}	(Absatzende)
Text	gg	(ganz hoch)	G	(ganz runter)

Editieren I

Text einfügen

i **i**nsert text before character	I **i**nsert before line
a **a**ppend text after character	A **a**ppend after line
o **o**pen new line after current	O **o**pen new line before current

Editieren II

Text löschen und ersetzen

x lösche aktuelles Zeichen	
d **d**elete text^{m}	D **d**elete rest of line
c **c**hange text^{mi}	C **c**hange rest of line^{i}
r **r**eplace character	R **r**eplace text
s **s**ubstitute character^{i}	S **s**ubstitute whole line^{i}

(^{m}: der Befehl wird für den nächsten Bewegungsbefehl ausgeführt)

(^{i}: der Befehl wechselt in den Einfügenmodus)

Editieren III

Rückgängig und Wiederherstellen

u	**u**ndo
Strg+r	**r**edo
.	führe letzten Befehl nocheinmal aus
[Zahl]	führe den nächsten Befehl [Zahl]-mal aus

Kopieren und Einfügen

Kopieren und Einfügen

y **y**ank text^m	Y **y**ank whole line
p **p**ut text after curser	P **p**ut text before cursor

Hinweis I

alle Befehle, die Text löschen, schneiden den eigentlich aus

Hinweis II

Vim benutzt seinen eigenen Speicher zum Kopieren – um die Zwischenablage mit anderen Programmen zu nutzen muß man vor dem Lösch-/Yank-/Einfügenbefehl die Kombination "+ eingeben

Markieren von Text im visuellen Modus

Welchsel in den visuellen Modus

v	markiere zeichenweise
V	markiere zeilenweise
Strg+v	markiere blockweise

$\longrightarrow$ Bewegungsbefehle verändern die Markierung

$\longrightarrow$ Befehle zum Editieren werden auf die gesamte Markierung angewendet

Suchen und Ersetzen I

Suchen und Ersetzen im normalen Modus

/ suche vorwärts	? suche rückwärts
n **n**ext result	N letztes Ergebnis
* nächstes Vorkommen vom Wort unter dem Cursor	# letztes Vorkommen vom Wort unter dem Cursor

Suchen und Ersetzen II

Suchen und Ersetzen über die Kommandozeile

:s/ABC/XYZ	Ersetze das erste „ABC” in der aktuellen Zeile durch „XYZ“
:s/ABC/XYZ/g	Ersetze jedes „ABC“ in der aktuellen Zeile durch „XYZ“
:*x*,*y*s/ABC/XYZ/g	Ersetze jedes „ABC“ in den Zeilen x bis y durch „XYZ“
:%s/ABC/XYZ/g	Ersetze jedes „ABC“ im Dokument durch „XYZ“

Reguläre Ausdrücke I

→ die Suche und das Ersetzen von Text erlaubt die Verwendung von *regulären Ausdrücken*

Beispiel

```
:%s/\(\\\)\(sub\)\{-}\(section\)\({\)/\1\2\3*\4/g
```

→ setzt Sterne hinter alle sections, subsections usw.

Mehr Informationen zu RegEx in Vim unter

[vimregex]

Reguläre Ausdrücke II

Hinweis

Man kann das Verwenden regulärer Ausdrücke unterbinden, indem man am Anfang des Suchbegriffs ein „\V“ einfügt

$\longrightarrow$ Schrägstriche und Backslashes müssen trotzdem noch als \\ und \/ eingegeben werden

Konfiguration von Vim

→ Vim wird über eine Plaintextdatei konfiguriert

Windows

```
<Benutzerordner>\_vimrc
```

Linux/Mac

```
<Benutzerordner>/.vimrc
```

Die Hilfefunktion von Vim

Aufrufen der Hilfe

:help [Thema/Befehl/usw.]


Vim-Homepage [link]


http://vimregex.com/ [link]


Vim introduction and tutorial [link]


0xBABAF000L [link]