

## vi

vi ist ein kleiner und einfacher Editor, den man auf jedem Unix-System finden wird. Er ist jedoch in seiner Handhabung recht gewöhnungsbedürftig. vi hat zwei verschiedene Modi: den Kommandomodus und den Editiermodus. Nach dem Start von vi ist man im Kommandomodus.

### Starten des Editiermodus

<i>a</i>	append (schreiben nach aktuellem Zeichen/Cursor)
<i>A</i>	append am Zeilenende
<i>cw</i>	change word (Ändern des Wortes unter dem Cursor)
<i>i</i>	insert (schreiben vor aktuellem Zeichen/Cursor)
<i>I</i>	insert am Zeilenanfang, der Rest wird verschoben
<i>o/O</i>	overwrite (Zeile nach/vor aktueller erzeugen und schreiben)
<i>r/R</i>	replace (Überschreiben von Zeichen/Zeile unter Cursor)
<i>s</i>	substitute (Einfügen von Text anstatt Zeichen unter Cursor)

### Rückkehr in Kommandomodus

Aus dem Editiermodus kommt man durch ESC wieder zurück in den Kommandomodus.

### Grundlegende Befehle im Kommandomodus

Im Kommandomodus kann man vor allem Speichern und vi verlassen, aber auch den Cursor bewegen.

<i>:q</i>	quit (Verlassen vi)
<i>:q!</i>	forced quit (Verlassen ohne Änderungen zu speichern)
<i>:wq</i>	write quit (Speichern und Beenden - auch ,ZZ')
<i>:w</i>	write (Speichern der Datei)
<i>:w &lt;file&gt;</i>	(Speichern in Datei ,file')
<i>:w!</i>	forced write (Speichern auch auf schreibgeschützte Dateien)

### Bewegen des Cursors

Bewegung des Cursors ist nur im Kommandomodus möglich. Wenn die Pfeiltasten nicht unterstützt werden, kann man auf folgende Tasten ausweichen oder darüber hinaus benutzen:

<i>h</i>	links	<i>w</i>	springe zu nächstem Wort
<i>j</i>	runter	<i>b</i>	springe zu vorherigem Wort
<i>k</i>	hoch	<i>0</i>	gehe zu Zeilenanfang
<i>l</i>	rechts	<i>\$</i>	gehe zu Zeilenende
<i>number G, :number</i>			gehe zu Zeile Nummer number

### Clipboard Kommandos

vi besitzt ein Clipboard, mit dem man sich einige Dinge erleichtern kann:

<i>dw</i>	Kopiere ein Wort ins Clipboard, lösche es.
<i>dd</i>	Kopiere eine Zeile ins Clipboard, lösche sie.
<i>yy</i>	Kopiere eine Zeile ins Clipboard
<i>p</i>	Füge aus dem Clipboard in die nächste Zeile ein.
<i>u</i>	undo (letzte Änderung zurücknehmen)

### Diverses

<i>J</i>	lösche Zeilenumbruch der aktuellen Zeile
<i>/regexp</i>	suche nach Zeichenkette (auch regulärer Ausdruck)
<i>?regexp</i>	suche rückwärts nach Zeichenkette
<i>n</i>	wiederhole Suche

## (X)Emacs

Die Funktionen sind im (X)Emacs über verschiedene Wege erreichbar: Menüs, Tastenkombinationen und direkte Eingabe des Funktionsnamens. Funktionen werden über die Tastenkombination M-x (M wie Meta, s. u.) aufgerufen, nach dem Drücken dieser Kombination wechselt der Cursor in den Minibuffer (die Zeile ganz unten im Fenster) und man kann den Funktionsnamen eingeben. Dabei gibt es einen Vervollständigungsmechanismus, d.h. man braucht nur den Anfang des Befehls schreiben, ein Druck auf die Tabulatortaste vervollständigt den Befehl, wenn er eindeutig ist, oder öffnet ein Fenster mit den möglichen Alternativen. Wenn der Funktionsname vollständig ist, mit der Eingabetaste die Funktion starten, evtl. wird dann - wieder im Minibuffer - nach Parametern gefragt, wie z.B. Dateinamen.

### Tasten

C-x	Control gedrückt halten, x drücken
M-x	Alt gedrückt halten, x drücken

### Funktion Tastenkomb. Funktionsname

#### Dateimanagement

Datei laden/anlegen	C-x C-f	find-file
Datei speichern	C-x C-s	save-buffer
Geänderte Dateien speichern	C-x s	save-some-buffers
Datei umbenennen	C-x C-w	write-file
Umschalten auf andere Datei	C-x b	switch-to-buffer
(X)Emacs beenden	C-x C-c	save-buffers-kill-emacs

#### Editieren/Cursorbewegung

Das Undo ist unendlich, wenn man mehrere Schritte rückgängig machen will, ruft man mehrmals direkt(!) hintereinander die Undofunktion auf.

Undo	C-x u	advertised-undo
Zeilenanfang	C-a, Home	beginning-of-line
Zeilenanfang	C-e, End	end-of-line
Wort nach rechts	C-rechts	forward-word
Wort nach links	C-links	backward-word
Dateianfang	C-Home	beginning-of-buffer
Dateiende	C-End	end-of-buffer
Wort nach rechts löschen	M-d	kill-word
Zeile bis Zeilenende löschen	C-k	kill-line

#### Suchen/Ersetzen

Suchen vorwärts	C-s	isearch-forward
Suchen rückwärts	C-r	isearch-backward
Suchen vorwärts mit Regexp	S-M-s	isearch-forward-regexp
Suchen rückwärts mit Regexp	S-M-r	isearch-backward-regexp
Ersetzen	M-%	query-replace
Ersetzen mit Regexp	C-M-%	query-replace-regexp

### Blockoperationen

Blöcke werden im (X)Emacs erstellt, indem man den Cursor an den Anfangspunkt bewegt, set-mark-command aufruft, dann Cursor an den Endpunkt bewegt und dann den Block kopiert oder ausschneidet. Man kann natürlich auch die Maus zum Markieren nehmen und muß dann am Ende noch sagen, ob ausgeschnitten oder kopiert werden soll. Der (X)Emacs führt eine Liste der letzten kopierten/ausgeschnittenen Blöcke, man kann also mehrere auf einmal kopieren und später wieder einfügen.

Anfangsmarke setzen	C-Leertaste	set-mark-command
Markierten Block kopieren	M-w	kill-ring-save
Markierten Block ausschneiden	C-w	kill-region
Zuletzt kopierten/ausgeschnittenen Block einfügen	C-y	yank
Umschalten zu vorletztem Block in Liste (funktioniert nur, wenn direkt davor ein Block eingefügt wurde)	M-y	yank-pop

### Fenster

Weiteres vertikales Textfenster	C-x 2	split-window-vertically
Weiteres horizontales Textfenster	C-x 3	split-window-horizontally
Wechsel in anderes Fenster	C-x o,	
	C-Tab	command-other-window
aktives Fenster schliessen	C-x 0	delete-window
ein inaktives Fenster schliessen	C-x 1	delete-other-windows

### Diverses

Hilfe zu Funktion	C-h f	describe-function
Hilfe zu Tastenkombination	C-h k	describe-key
Tutorial starten	C-h t	help-with-tutorial
Abbrechen (merken!)	C-g	keyboard-quit

## Windows:

Windows ist ein weit verbreitetes Betriebssystem. Viele Dozenten haben eher Windows auf ihrem Rechner installiert als Unix. Die folgende Referenz ist nicht ganz ernst. Ernst gemeint ist aber der Hinweis, daß nur wenige gern Windows-Dokumententypen (ausser RTF/ANSI) als ein Austauschformat (z.B. via mail) annehmen. Schickt also lieber PDF oder PS, wenn ihr mal Dokumente verschicken müsst.

### Tastenkürzel

<i>Alt-F4</i>	Programm schliessen.
<i>Alt-Tab</i>	Zu anderem Programm wechseln.
<i>Ctrl-Alt-Del</i>	Taskmanager rufen (Programm umständlich schliessen)
<i>-, - mehrmals</i>	Windows abgestürzt, aufgeben

### Beliebte Programme

<i>telnet</i>	unsicher auf einen Rechner einloggen
<i>putty</i>	sicher (per SecureShell) auf einen Rechner einloggen ( <a href="http://www.chiark.greenend.org.uk/~sgtatham/putty/">http://www.chiark.greenend.org.uk/~sgtatham/putty/</a> )
<i>iexplore</i>	bunte Seiten besuchen, manchmal dabei abstürzen

## Unix-Befehle

Unix und Unix-Derivate sind an unserer Universität die verbreitetsten Betriebssysteme. Die folgende Auflistung ist nicht vollständig und kann teilweise für verschiedene Shells (Umgebungen) abweichen. Daher ist das Kommando „man“ besonderes ans Herz gelegt.

## Kommandosyntax

Kleinbuchstaben benutzen. Jedes Kommando mit "Return" beenden. Text durch die eigenen Angaben ersetzen.

### command options files

options, soweit vorhanden, gehen in der Regel files voraus. Es können ein oder mehrere files angegeben werden. Details in den man-pages.

### command < file

command benutzt als Standardeingabe file anstelle der Tastatur.

### command > file

command sendet seine Standardausgaben an file anstelle des Terminals. Wenn file existiert, wird es überschrieben.

### command >> file

command hängt seine Standardausgaben an das existierende file an. Falls file noch nicht existiert, wird es angelegt.

### command1 | command2 | ...

Die Standardausgabe von command1 wird zur Standardeingabe von command2, usw.

### nice -19 command

command wird mit einer niedrigen Priorität gestartet. Dies sollte benutzt werden, wenn command die Systemressourcen stark beansprucht.

## Grundlegende Kommandos

passwd	Ändert das Passwort auf allen Rechnern (gegen 16 Uhr). (Aus Sicherheitsgründen regelmäßig ändern.)
logout/exit	Beendet die Sitzung auf diesem Rechner.
finger user	Zeigt Informationen zu user
clear	Löscht den Bildschirm.
history	Zeigt eine Liste der zuletzt benutzten Kommandos an.
script file	Startet die Aufzeichnung der aktuellen Sitzung. exit beendet die Aufzeichnung wieder.

## Online Hilfe

Das gesamte Unix Reference Manual ist online verfügbar.

man topic	Zeigt das Manual zu topic (z.B. Kommandooptionen, Funktionsparameter); siehe man man
man -s 3 topic	Zeigt den Manualeintrag für topic aus Teil 3 des Online-Manuals (Library- Routinen).
info	Startet die interaktive Oberfläche für die GNU-Manuals.
command --help	Zeige die Hilfe eines bestimmten Programms an.

## Besondere Tasten

<ctrl>-c	Beendet das aktuelle Programm und kehrt zur (Shell) zurück.
<ctrl>-z	Unterbricht die Ausführung des aktuellen Programmes und kehrt zur Shell zurück. Siehe Jobkontrolle.
<ctrl>-d	Signalisiert Eingabeende an das wartende Programm. Funktioniert auch mit dem Login-Prompt.
<ctrl>-s/<ctrl>-q	Unterbricht die Terminalausgabe, bzw. setzt sie fort.

## Dateien und Verzeichnisse

Pfade sind mit "/" getrennte Sequenzen von Verzeichnissen, die mit einer Datei oder einem Verzeichnis enden. Standardpfade sind relativ zum aktuellen Verzeichnis. Absolute Pfade beginnen mit einem "/" und sind relativ zum Wurzelverzeichnis des Dateisystems.

ls -al directory	Zeigt alle Dateien (incl. Dateien, deren Namen mit einem Punkt beginnen) im directory im Langformat an.
more file	Zeigt den Inhalt von file seitenweise an. more kann mit q beendet werden.
less file	Wie more. Nur benutzerfreundlicher.
cat file	Komplettes Anzeigen des Inhalts von file.
split -b 500k file	Zerlegen einer Datei in Blöcke von 500kByte.
cat f1 f2 > f3	Schreibt den Inhalt von f1 und f2 hintereinander in f3. Geeignet um ge'split'ete Dateien wieder zu vereinen.
pwd	Zeigt absoluten Pfad des aktuellen Verzeichnisses an.
cd directory	Wechselt zu directory. Wenn kein directory angegeben ist, wird zum Home-Verzeichnis gewechselt.
mkdir directory	Erzeugt ein Verzeichnis namens directory
rmdir directory	Entfernt das Verzeichnis directory.
cp file newfile	Kopiert file nach newfile.
cp f1 f2 directory	Kopiert alle f1, f2 in das Verzeichnis directory. Die Originaldateien bleiben unverändert.
mv file newfile	Benennt file in newfile um.
mv f1 f2 directory	Verschiebt f1, f2 in das Verzeichnis directory.
rm -i f1 f2 f3	Löscht alle angegebenen Dateien. Ohne -i wird nicht nachgefragt und f1 f2 f3 ... verschwinden für immer ( <b>ein Undelete ist unter UNIX nicht möglich</b> ).
rm -ir f1 f2 d1 d2	Löscht alle Dateien und Verzeichnisse (incl. der in diesen Unterverzeichnissen).

rm -rf f1 f2 d1	Löscht ohne Nachfragen (-force) die angegebenen Dateien und Unterverzeichnisse (-recursive).
file f1	Zeigt Informationen über den Typ/Inhalt von f1 an.
mdir	Siehe mtools: Anzeigen des Inhalts einer Diskette.
mcopy file a:/	Kopieren einer Datei auf Diskette.
mread a:/...	Kopieren von Diskette.

## Nützliche Kürzel

Wildcards werden vor der Ausführung des Kommandos durch die Shell aufgelöst. Die folgenden Kürzel sind Beispiele.

ls ..	Zeigt den Inhalt des übergeordneten Verzeichnisses. Der Name des aktuellen Verzeichnisses ist mit in der Liste enthalten.
cp file ~	Kopiert file in das Home-Verzeichnis.
rm *.bak	Löscht alle Dateien mit Endung .bak (oft Sicherungsdateien).

## Umgebungs-Setup

### alias shortname command

Erzeugt das Kürzel shortname für command. Starte shortname um command auszuführen (siehe auch Datei ~/.alias)

### set/setenv wert

Je nach Shell kann man damit Umgebungsvariablen festlegen.

## Dateischutz

Dateien und Verzeichnisse gehören einem Benutzer und einer Gruppe und haben unabhängige read-, write- und execute-Rechte für user, group und others. Für ein Verzeichnis sind execute-Rechte notwendig, um es in einem Pfad angeben zu können. Das Erzeugen/Löschen eines Files erfordert write- und execute-Rechte am Verzeichnis. Is an einem Verzeichnis erfordert read- und execute-Rechte.

chmod u+x file	Macht file für den Eigentümer ausführbar.
chmod o+r file	Macht file für alle Benutzer lesbar. Eigentümer- und Gruppenrechte werden nicht beeinflusst.
chmod g+r file	Macht file für die Gruppenmitglieder lesbar.
chmod 644 file	Macht file für den Eigentümer les/schreibbar, für alle anderen lesbar

## Jobkontrolle

Jobs entsprechen von der Shell gestartete Kommandos. Jobs können beliebig viele Prozesse erstellen.

jobs	Listet alle Jobs (z.B. mit <ctrl>-z angehaltene Jobs) samt ihren Jobnummern auf.
fg jobnumber	Setzt den angehaltenen Job im Vordergrund fort.
bg jobnumber	Setzt den angehaltenen Job im Hintergrund fort. Bei versuchter Terminalausgabe hält der Job an.
kill jobnumber	Beendet den Job jobnumber und dessen verbundenen Prozesse.
kill -9 jobnumber	Hartes Beenden eines Jobs.

## Beliebte Software

Diese Software ist an allen Sun Workstations vorhanden.

cc -o outfile file.c	Übersetzt das C-Programm file.c. Das ausführbare Programm heißt outfile.
g++ -o out f.cpp	Übersetzt das C++-Programm f.cpp ...
emacs, vi, nedit	Editoren.
latex file.tex	Übersetzt file.tex mit dem Textsatzsystem LaTeX.
pine, elm, mutt	Mailprogramme.
Netscape	WorldWideWeb-Browser Netscape.
ftp/sftp	Dateiübertragungsprogramme. (sftp - SecureFTP)
ssh	SecureShell (sichere Shell, einzige Login-Möglichkeit von Fremdrechnern auf Rechner der RBG)
ssh -l wer wo	Versucht sich als wer auf dem Rechner wo einzuloggen.
gzip -9 file	Maximales Komprimieren einer Datei
tar	Zusammenkleben mehrerer Dateien
tar -cvzf file d1/*	Zusammenkleben aller Dateien im Verzeichnis d1 in die Datei file und Komprimieren mit gzip
tar -xvzf file	Dekomprimieren und Entpacken eines sogen. Tarballs.

## Wichtige Dateien/Verzeichnisse

~/forward	an dort eingetragene E-Mail-Adressen werden Mails weiterverschickt (mit Präfix \ weiterkopiert)
~/cshrc, ~/.bashrc, ~/.tcshrc	Konfigurationsdateien für die benutzte Shell.
~/public_html/	Verzeichnis für persönliche Homepage (IfI, HU Berlin)