

Befehle

Alle Befehle, über die **emacs** verfügt, können auf dreierlei Weise aufgerufen werden: erstens über die schon vorgestellten Tastenkürzel, sofern der Befehl an eine Tastenkombination gebunden wurde. Zum anderen über die Menüleiste, sofern der Befehl dort aufgeführt ist. Und zum dritten über den Funktionsnamen: Beispielsweise ruft die Tastenkombination **C-y** die Funktion **yank** auf. Diese kann auch mit ihrem Namen aufgerufen werden: **M-x** nimmt den Namen im Minibuffer entgegen — ebenfalls mit Vervollständigung durch die **Tab**-Taste — und führt ihn dann aus.

Tour de Features

Böse Zungen behaupten, **emacs** könne alles, außer Kaffee kochen. Im Zeitalter des Internet der Dinge geht das bestimmt auch.

Das Killer-Feature von **emacs** ist meines Erachtens die Fähigkeit, durch sprachabhängige Modi und Erweiterungen das Verhalten von **emacs** auf den gerade vorliegenden Anwendungsfall zu spezialisieren. Eine subjektive Auswahl von Tricks, die **emacs** beherrscht²:

Unicode: Unterstützung für annähernd alle Sprachen der Welt. **C-h h** um die Datei HELLO anzusehen. Man kann mehrsprachige Dokumente verfassen, z.B. auch für das **T_EX/L^AT_EX**-Satzsystem.

Programmiersprachen: Für eine erstaunliche Anzahl von Programmiersprachen gibt es entsprechende Modi, die Syntax-Highlighting, Automatische Einrückung, Vervollständigung, Anbindung der entsprechenden Compiler und Debugger etc. anbieten. Kleine Auswahl: Assembler, C/C++, Erlang, Forth, Java, JavaScript, Python, perl, shell . . . aber auch für HTML, XML, **T_EX/L^AT_EX** und anderen strukturierte Sprachen. Siehe auch **ecb** (Emacs Code Browser), **cedet** und **auctex**

Remote Zugriff: mit **emacs** kann man auch Dateien editieren, die auf einem anderen Rechner liegen. Die Erweiterung heißt **tramp** und benutzt **ssh/scp** als Transportmedium.

Rechner: es gibt einen sehr mächtigen (Taschen-)Rechner namens **calc**, der **stack** basiert ist (Freunde der HP-Taschenrechner lieben das) und eine unglaubliche Anzahl von eingebauten Funktionen beherrscht.

ediff, emerge: Zwei Versionen einer Datei vergleichen? In bunt? Aus diesen eine dritte Version erstellen?

Versionskontrolle: Anbindung an die Versionsverwaltung mit **subversion**, **git**, etc. per Tastenkürzel.

Dateimanager: **dired** ist eingebaut

Newsreader: **gnus** ist eingebaut

Email: Es gibt **vm** (viewmail), welcher noch heute unterstützt wird, und einen eingebauten email Modus.

²Andere Editoren beherrschen solche Sachen auch. Keine Religionskriege, bitte!

Personal Information Manager: Es gibt mit **org-mode** ein sehr mächtiges Werkzeug, um aus **emacs** heraus Notizen, Termine, Aufgaben, Kontakte zu verwalten, zu verknüpfen, ein Journal zu führen, Zeiterfassung und -management bis hin zur Projektplanung und zur Erstellung von per Webserver zugänglichen Exporten dieser Information. Das beste: Alle Dateien sind *plain ascii* und können zur Not ohne emacs gelesen und editiert werden. Eine Anbindung an Mobile-Geräte existiert ebenfalls (**MobileOrg**).

Spreadsheet: Mit **org-mode** kommt auch der **orgtbl-mode**, mit dem es ein Kinderspiel ist, Information in Tabellenformat zu halten. Formeln sind möglich, auch hier ist alles in *plain ascii* abgelegt. Dieser Modus ist als *minor mode* auch unabhängig von **org** verwendbar.

Manpages: es gibt mehrere Modi, um die vorhandenen manual pages zu lesen, z.B. **woman**.

Spiele: gibt's selbstverständlich auch, z.B. **tetris**.

Shell: nötig? **M-x shell** hilft.

Webbrowsing: geht mit einer Erweiterung namens **w3m**.

Internet Relay Chat: **M-x irc**

Eigentlich gibt es nicht viele Gründe, emacs zu verlassen.

Hilfe

Das eingebaute Hilfesystem (Info) ist über das Tastaturkürzel **C-h i** oder über die Menüleiste erreichbar. Es ist sehr umfangreich, ein paar Stunden Stöbern sind gut angelegte Zeit. Wichtigster Anlaufpunkt ist der *Concept Index*, den man bequem durchsuchen kann (**C-s**).

Für Anfänger unbedingt empfehlenswert ist das Tutorial (**C-h t**, welches über die Menüleiste auch in anderen Sprachen verfügbar ist (**HELP | EMACS TUTORIAL (CHOOSE LANGUAGE)...**) Empfehlenswert ist außerdem die *Emacs Tour* [2].

Außerdem ist das Emacswiki [3] die wahrscheinlich größte Sammlung von Informationen außerhalb der Infoseiten. Für Org-Mode gibt es eine eigene Webseite [4] sowie eine Community Seite [5].

Links

1. <http://www.gnu.org/software/emacs/>
2. <http://www.gnu.org/software/emacs/tour/>
3. <http://www.emacswiki.org/>
4. <http://orgmode.org>
5. <http://orgmode.org/worg/>

L^AT_EX-Dokument zum Chemnitzer Linuxtag 2013-03-16, leicht geändert zum Linuxtag Berlin 2013-05-22, Erich Wälde

Werkzeug: emacs

www.gnu.org/software/emacs

www.emacswiki.org

Werkzeug

Programmierer verbringen viel Zeit am Rechner. Gutes und gut erlerntes Werkzeug kann helfen, bequemer und in kürzerer Zeit zum Ziel zu gelangen. In Unix/Linux-Land gehören meiner Meinung nach zu diesen Werkzeugen mindestens a) eine hervorragende Tastatur, b) die Shell, c) ein Editor und d) **make** oder ein anderes Build-System.

Der Editor ist ein Programm, mit dessen Hilfe jegliche Dateihalte angezeigt, geändert und gespeichert werden können.

Editor: emacs

emacs ist ein ehrwürdiger Editor, dessen Anfänge auf eine Makrosammlung für den Editor **TECO** zurückgehen (1976). Ab 1984 wurde **emacs** für das frisch entstandene GNU Projekt neu entwickelt. Die Entwicklung wurde hauptsächlich von Richard Stallman vorangetrieben. **emacs** ist in C und emacs lisp geschrieben. Die aktuelle Version ist 24.3.

emacs ist hochgradig konfigurierbar, erweiterbar (in emacs lisp), selbst-dokumentierend (die Dokumentation von Funktionen ist im Quelltext hinterlegt, es gibt ein mächtiges Hilfesystem). Der Kern von **emacs** ist ein Lisp Interpreter, der sich meistens wie ein Texteditor benimmt.

emacs läuft auf sehr vielen Systemen: GNU/Linux, diverse Unixe, VAX/VMS, *BSD, Windows, MacOS . . .

emacs unterliegt der General Public License (GPL v3).

Installation

Um **emacs** zu installieren konsultiert man am besten den Paketmanager seiner bevorzugten Distribution. Andernfalls bekommt man die Quellen von der Projektseite [1]. Es gibt mehrere Varianten von **emacs**, z.B. **xemacs**. Dieser Text bezieht sich nur auf den **GNU/emacs**.

Aus lizenztechnischen Gründen ist in Debian-basierten Distributionen die Dokumentation in dem separaten Paket **emacs24-common-non-dfsg** hinterlegt, welches sich in der Kategorie **non-free** befindet.

Wichtige Konzepte

```
File Edit Options Buffers Tools Help

U:--- test.txt All L1 (Text)
```

Menü- und Werkzeugleisten

In der ersten Zeile findet sich eine Menüleiste, wie man es bei sehr vielen Programmen findet. Läuft der `emacs` in X, dann lässt sich diese Leiste auch mit der Mouse bedienen. Das ist zumindest zum Herumprobieren und Suchen sehr praktisch. Die Tastenkürzel sind netterweise in den Menü-Einträgen mit angegeben.

Die Werkzeugleiste ist so aufgebaut, wie man sie bei vielen Programmen findet. Wem der Platz zu schade ist oder wer, wie ich, mit den Symbolen wenig anfangen kann, der kann diese Leiste ohne Verlust deaktivieren (`OPTIONS | SHOW/HIDE | TOOLBAR | NONE`).

Puffer

Der Puffer (buffer) ist die wichtigste Einheit, um Inhalte anzuzeigen. Wenn man eine Datei editiert, so ist dieser ein Puffer zugeordnet, in dem alle Aktionen (einfügen, löschen, verändern von Inhalt) stattfinden. Es gibt aber auch Puffer, denen keine Datei zugeordnet ist, zum Beispiel für Meldungen oder für die Liste aller vorhandenen Puffer.

Es kann gleichzeitig mehrere Buffer geben, von denen einer oder mehrere gleichzeitig angezeigt werden.

Rahmen

Ein Rahmen (frame) ist das, was wir normalerweise Fenster nennen. Er wird vom Fenstermanager verwaltet. In diesem können ein oder mehrere Puffer angezeigt werden. Ein `emacs`-Prozess kann mehrere Rahmen öffnen, es gibt keinen Grund, dafür mehrere Instanzen von `emacs` zu starten.

Statuszeile

Die zweitletzte Zeile beinhaltet einige (konfigurierbare) Informationen über den zugehörigen Buffer: Ein Marker, ob die Datei geändert wurde, den Namen des Puffers (der meistens mit den Dateinamen übereinstimmt), in welcher Zeile sich der Cursor befindet und welcher Modus aktiviert ist.

Minipuffer

Die unterste Zeile im `emacs`-Fenster ist der Minipuffer. Hier werden Meldungen angezeigt und Eingaben abgefragt.

Erster Versuch

Wer `emacs` zum ersten Mal startet, der bekommt einen Puffer namens `*GNU Emacs*`, in dem unter dem rot-blauen GNU Emacs Logo die allerwichtigsten Dinge angezeigt werden. Wer schon einmal *irgendwie* in `vi` gestrandet ist, und nicht wusste wie man da wieder raus kommt, der möchte doch sofort wissen, wie er/sie da wieder rauskommt, ohne Flurschaden anzurichten. Die Antwort heißt

`C-x C-c`

Diese kryptische Schreibweise heisst übersetzt: Drücke die `Ctrl`-Taste und halte sie gedrückt. Dann drücke ein `x`, lass es los, anschließend ein `c` und lass es los. Erst jetzt lass die `Ctrl`-Taste wieder los. Uffz — der `emacs` verschwindet, wie ihm geheißen. Ausprobieren!

Wer bis hier gekommen ist, hat gelernt, den `emacs` zu starten und zu verlassen, und viel wichtiger: was es mit der Notation der Tastatur-Kürzel auf sich hat. Ok, Mouse-geübte Naturen wählen *intuitiv*¹ das linke Menü `FILE` und finden als letzten Eintrag `QUIT`. Aber wir wissen jetzt, was es mit den geheimnisvollen Zeichen `C-x C-c` auf sich hat, die hinter `QUIT` stehen.

Notaus

Wer sich irgendwie in den Tastenkürzeln hoffnungslos verhasht hat, der kann sich daraus mit ein oder mehreren

`C-g`

befreien

Zweiter Versuch

Die vornehmste Aufgabe eines Editors ist das Einfügen/Löschen von Zeichen in eine Datei. Also — nein wir wählen jetzt nicht `FILE | VISIT NEW FILE ...`, sondern wir tippen mutig

`C-x C-f`

Dann öffnet sich kein Fenster, in dem wir irgendwie eine Datei aussuchen, sondern `emacs` fragt uns im Minibuffer (die letzte Zeile)

Find file: ~/

nach dem Dateinamen. Wir können da jetzt einen Namen ergänzen, z.B. `test.txt`. Ein abschließendes `Return` erzeugt einen Puffer mit den Namen `test.txt`, dem eine Datei gleichen

Namens zugeordnet ist. Der Cursor wird am Anfang des Puffers plaziert.

Wir können während der Eingabe im Minipuffer aber auch mit `Tab`-Vervollständigung ein Verzeichnis oder eine vorhandene Datei angeben lassen. Drückt man erneut `Tab`, dann halbiert sich die Fläche, in der der `*GNU Emacs*` Puffer angezeigt wurde, und die möglichen Vervollständigungen werden angezeigt.

In unserem neuen Puffer können wir jetzt munter irgendeinen Text eintragen. Die meisten Tasten der Tastatur machen genau das: sie tragen ihr Zeichen im Puffer ein.

Wer jetzt in der Shell oder im Dateimanager nachsieht, wird feststellen, daß es gar keine Datei `test.txt` gibt. Der Inhalt des Puffers wandert erst in die Datei, wenn wir den Puffer sichern. Selbstverständlich können wir dieses eine Mal im Menü `FILE` nachsehen, welches Tastenkürzel zum Abspeichern vorgesehen ist:

`C-x C-s`

`emacs` bestätigt uns den Speichervorgang im Minipuffer. Alles ist gut.

Mit diesen beiden Tastenkürzeln kann man schon überleben. Aber bequem ist möglicherweise anders. Wichtig: `emacs` startet man ein mal, etwa beim Anmelden, und lässt ihn laufen. Es gibt keinen Grund für jede Datei, die man editieren möchte, einen neuen `emacs` zu starten. Er kann auch mehrere Fenster gleichzeitig anzeigen, ohne einen neuen Prozess zu starten. Bei mir läuft der `emacs`, bis ich den Rechner herunterfahre, manchmal wochenlang ohne Unterbrechung.

Dritter Versuch

Editieren endet sehr schnell in der Aufgabe, ein Stück Text von einer Stelle an eine andere zu verschieben oder zu kopieren. Dazu verwendet man ein Konzept namens *region* mit den Begrenzern *mark* und *point*. Der *point* ist an der Stelle, an der sich der Cursor befindet. Genau genommen direkt vor dem Cursor. Diese Position kann mit `C-space` (Control-Leertaste) in den *mark* kopiert werden. Bewegt man den Cursor an eine andere Stelle, so existiert zwischen *point* und *mark* die Region. Viele Befehle reagieren auf die Region, wenn diese definiert ist. Normalerweise wird die Region auch etwas anders gefärbt.

Hat man eine Region markiert, dann kann man sie per

`M-w` in die Ablage kopieren

`C-w` in die Ablage kopieren und löschen

`C-y` bei *point* einfügen

`M-w` bedeutet, daß anstelle der `Ctrl`-Taste die `Meta`-Taste gedrückt wird. Normalerweise ist die mit `ALT` beschriftet. Diese zugegebenermaßen sehr dürre Auswahl wird auf der Referenzkarte ausführlicher dargestellt.

¹Für eine hinreichend intuitive Definition von *intuitiv* :-)