

Editieren unter LINUX

August Hörandl

Emacs und XEmacs

Dieser Editor ist ein mächtiges Werkzeug, die Bezeichnung Editor ist für diese Arbeitsumgebung fast eine Untertreibung. Es ist ein sehr umfangreiches Programm mit den wahrscheinlich meisten Features eines einzelnen UNIX-Programms.

Emacs basiert auf einer eigenen Programmiersprache (Lisp bzw. Emacs-Lisp) für Erweiterungen und Makros (die meisten Pakete des Emacs sind in dieser Sprache geschrieben).

Es können beliebig viele Dateien gleichzeitig bearbeitet werden. Neben den Funktionen zum Bearbeiten von Texten (Einfügen, Kopieren, etc.) gibt es eine Undo-Funktion (beliebig weit), inkrementelles Suchen (Suchergebnisse werden während der Eingabe des Suchbegriffs angezeigt), Autosave und Menüs.

Die enthaltenen Erweiterungen decken fast alle Bereiche ab: z.B. Programme schreiben, übersetzen und testen (wie eine integrierte Entwicklungsumgebung), alle Arten der Kommunikation (Mail, News, WWW) sind am Textschirm und unter X Windows möglich. Für sehr viele Arbeiten gibt es spezielle Unterstützung (sogenannte Modes) z.B. je ein Mode für die gängigen Programmiersprachen (einrückern, farbige Darstellung der Schlüsselwörter etc.), HTML Mode zum Gestalten von WWW Dokumenten usw.

Häufig verwendete Kommandos sind durch "shortcuts" (eine Kombination der Tasten **Meta**, **Ctrl** und einer weiteren Taste) abrufbar.

Alles in allem: es gibt fast nichts, was Emacs nicht kann.

Diese Funktionalität hat auch ihren Preis: Emacs braucht viel Speicher (bei der Installation und bei der Ausführung) und ist für den Neuling durch die vielen Möglichkeiten eher entmutigend.

Anmerkung

Es gibt zwei Versionen: Emacs und XEmacs. Beide sind für X Windows geeignet, sie unterscheiden sich etwas durch die Funktionalität - die Version XEmacs ist dabei etwas anwenderfreundlicher und umfangreicher.

Starten des Editors

```
$ emacs [optionen] [datei ... ]
$ xemacs [optionen] [datei ... ]
```

Bereits beim Start können die zu bearbeitenden Dateien angegeben werden.

Man kann eine Datei auch in einem neuen Buffer des bereits laufenden Editors laden (auch hier gibt es zwei Versionen):

```
$ emacsclient datei
$ gnucient datei
```

Das ist besonders dann sinnvoll, wenn Emacs als "externer" Editor für andere Programme genutzt wird. Die Konfiguration geschieht meist durch Setzen der Variable EDITOR:

```
$ export EDITOR=gnucient
```

Damit Emacs in dieser Form arbeitet, muss bei Start ein Server gestartet werden - am einfachsten in der Konfigurationsdatei .emacs:

```
(serverstart) ;; server for
emacsclient
(gnuserv-start) ;; server for gnucient
```

Bearbeiten

Der angezeigte Schirm ist in drei Teile geteilt: den Buffer zum Bearbeiten der Datei, eine Statuszeile und die unterste Zeile zur Eingabe von Kommandos und zur Anzeige von Meldungen.

Die Bewegung des Cursors geschieht am einfachsten durch die Cursorstasten.

Außerdem können alle Kommandos in der untersten Zeile direkt eingegeben werden: durch Eingabe von M-x (M steht dabei für **Meta**) - falls sie keine solche Taste auf ihrer Tastatur finden, kann dies entweder durch vorheriges Drücken der **ESC** oder gleichzeitiges Drücken der **Alt** Taste ersetzt werden d.h. **ESC X** oder **Alt X**) und Eingabe des Kommandonamens. Bei vielen Eingaben funktionieren die **TAB** Taste zur Ergänzung und die Cursorstasten zur Auswahl der vorigen Eingaben (Commandhistory).

Beispiel

Öffnen einer Datei: M-x find-file. Man kann aber auch C-x C-f (**Ctrl X**) gefolgt von **Ctrl F**) verwenden.

Die Taste C-g bricht jedes Kommando ab.

Die zweite Zeile von unten ist die sogenannte Modeline (Statuszeile). Dort findet man Informationen über die aktuelle Datei, den für die Bearbeitung der Datei gewählten Modus usw.

Beispiel

```
-*-Xemacs: main.C (C++ Font)-L2-Top-
```

**	die Datei wurde geändert (%% wenn die Datei nur zum Lesen geöffnet ist)
C++ Font	ausgewählte Modi
L2	aktuelle Zeile
TOP	Beginn der Datei

Text löschen und verschieben

Das Markieren eines Bereiches (Region) geschieht entweder mit der Maus oder mit der Tastenkombination C-@ (oder C-Space) und Bewegen des Cursors.

C-w löscht den Bereich, M-w kopiert den Bereich (ohne Löschen) - der Text kommt dabei in den Kill-Ring (Zwischenablage mit beliebig vielen Einträgen). Mit dem Befehl C-y (yank) kann dieser Text an anderer Stelle eingefügt werden; mit M-y können dann alle bisherigen Einträge des Kill-Rings durchgegangen werden.

Emacs als Entwicklungsumgebung

Der Editor Emacs bzw. Xemacs bietet vielfältige Unterstützung beim Programmieren:

- Automatische Erkennung der Programmiersprache bzw. des Dateityps und entsprechende Unterstützung durch einen speziellen Mode:
- abgestimmte Menüs
- zusätzliche Befehle zum Bewegen im Text (nächste Funktion etc.) und zur schnellen Eingabe von bestimmten Standardausdrücken
- Einrücken der einzelnen Zeilen - verschiedene Stile möglich
- Farbige Darstellung (Syntax-Highlighting: M-x font-lock-mode)
- Menü mit allen Funktionsnamen und Anzeige der aktuellen Funktion in der Statuszeile
- Tags
- Suchen bzw. Ersetzen in allen Dateien eines Projekts (M-x tags-search)
- Sprung zur Definition einer Funktion (auch in einer anderen Datei)
- Aufruf des Compilers
- Ablauf in einem eigenen Buffer (M-x compile)
- Schneller Sprung von der Fehlermeldung zur entsprechenden Zeile im Text
- Aufruf des Debuggers
- Ablauf in einem eigenen Fenster
- Anzeige der aktuellen Zeile durch Markierung im Text
- Hilfe anzeigen

- Man Pages
- Info Dateien
- Versionskontrolle

Weitere Möglichkeiten

Es gibt wahrscheinlich niemanden, der alle Emacs Befehle kennt und anwenden kann. Selbst nach einigen Jahren des intensiven Gebrauchs findet man ein neues Kommando; Nicht zuletzt deshalb, da alle Pakete weiterentwickelt werden und immer wieder Funktionalitäten neu dazu kommen.

Einige Beispiele

cvcs	in Emacs können fast alle CVS-Befehle direkt ausgeführt werden (siehe Kapitel , Seite)
file-locking	Emacs sperrt die bearbeiteten Dateien gegen Änderungen durch andere Benutzer.
diary/calender	Neben der Standard (Gregorianisches Datum) beherrscht Emacs folgende Kalender: Astronomisch, Hebräisch, Islam und einige seltener verwendete: Julianisch, Maya und den Kalender der Französischen Revolution (inklusive der Mondphasen und Sonnenauf- und untergang).
ange-ftp, EFS	zum Bearbeiten von Dateien auf anderen Rechnern. Die Dateien werden im Hintergrund mittels ftp geladen und beim Speichern wieder zurückgeschickt.
M-x smiley-buffer	zur besseren Darstellung der ;-)

Falls das alles zu viel ist: M-x doctor startet die eingebaute Psychoanalyse.

Ein C-Programm mit Emacs erstellen, übersetzen und testen.

Man kann in den leeren Buffer (*scratch*) der von Emacs beim Start angezeigt wird schreiben und später mit M-x write-file einen Dateinamen vergeben. Besser ist es aber, eine neue Datei mit der entsprechenden Bezeichnung zu öffnen. Damit wird der passende Modus (c-mode) ausgewählt:

M-x find-file (oder Tastenkombination C-x C-f)
first.c

Um die Übersichtlichkeit zu erhöhen kann mit M-x fontlock-mode die farbliche Darstellung aktiviert werden.

Bei der Eingabe ist zu beachten, dass der Editor etwas von der "richtigen" Einrückung weiß und beim Zeilenwechsel entsprechende Leerzeichen einfügt. Mit der TAB Taste kann die Einrückung einer einzelnen Zeile jederzeit wiederholt werden, auch ganze Bereiche können mit M-x indent-region bearbeitet werden. Die Anzahl der Zeichen und die Art der Ein-

rückung kann verändert bzw. mit M-x set-c-style aus verschiedenen Stilen ausgewählt werden.

Nach der Eingabe des Programms wird der Compiler aufgerufen. Das entsprechende Kommando lautet

M-x compile

Anstelle des Standardkommandos (Emacs schlägt make -k vor) benötigen wir
gcc -g -Wall first.c -o first

Bei der Eingabe von Kommandos und bei der Eingabe der Dateinamen kann mit der [TAB] Taste so weit eindeutig ergänzt bzw. eine Liste der Möglichkeiten angezeigt werden.

Der Editor speichert nach Abfrage alle ungesicherten Dateien. Der Compiler läuft als eigener Prozeß d.h. es kann während der Übersetzung weiter gearbeitet werden. In einem eigenen Fenster *compilation* werden die ausgeführten Kommandos und die aufgetretenen Fehler angezeigt.

Wird der Cursor in die Zeile mit einem Fehler positioniert, so kann mit Return die entsprechende Stelle in der Datei angesprungen werden.

Beim erneuten Aufruf des Compilers kann die Eingabe vereinfacht werden: M-x und dann Pfeil nach oben zur Auswahl der bereits eingegebenen Kommandos oder M-x recompile um den Compiler nochmals mit den gleichen Optionen aufzurufen.

Nach der Korrektur aller syntaktischen Fehlern kann für die Suche nach logischen Fehlern der Debugger mit M-x gdb gestartet werden. Nach der Angabe des Dateinamens erhält man ein Fenster zum Bedienen des Debuggers. Hier können alle gdb-Kommandos eingegeben werden.

Breakpoints können mit M-x gdb-break (C-x SPACE) direkt im Sourcecode gesetzt werden.

Gestartet wird das Programm mit run im gdb-Fenster. Beim Erreichen des Breakpoints wird die entsprechende Datei in einem zweiten Fenster angezeigt und die entsprechende Zeile mit = markiert.

NT & UNIX

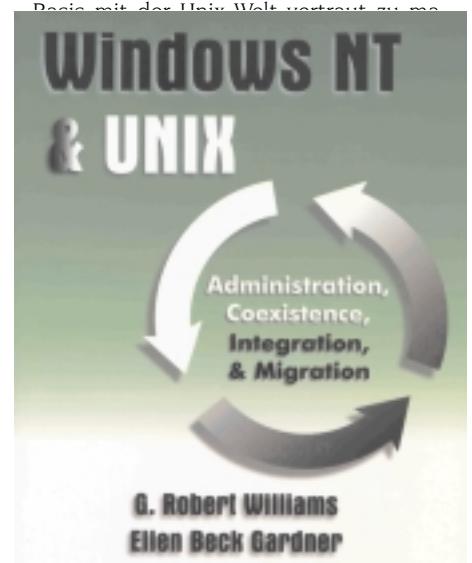
Christian Hofer

Mit einer ausführlichen Beschreibung der Funktionen und Hilfsprogramme beider Betriebssysteme und der Lösung administrativer Standardaufgaben wird der Leser auf die beiden Hauptkapitel hingeführt.

Darin erläutern die Autoren wie Unix und Windows NT nebeneinander am effizientesten genutzt und Migrationen in beide Richtungen durchgeführt werden können. Weiters wird die Integration von NT und Unix theoretisch und an Hand von zwei Beispielen (WEB- und E-Mail-Server) erklärt.

Im Anhang ist eine kommentierte, mit Beispielen versehene, Komplettübersicht an Unix-Befehlen mit den korrespondierenden Windows NT Programmen und umgekehrt enthalten.(!)

Das Buch eignet sich daher sowohl für erfahrene Windows NT Benutzer, sich auf NT Basis mit der Unix-Welt vertraut zu ma-



Williams, R. und Gardner E.: Windows NT & UNIX: Administration, Coexistence, Integration & Migration, 738 Seiten, Addison Wesley 1998, ISBN 0-201-18536-9, ATS 950.-