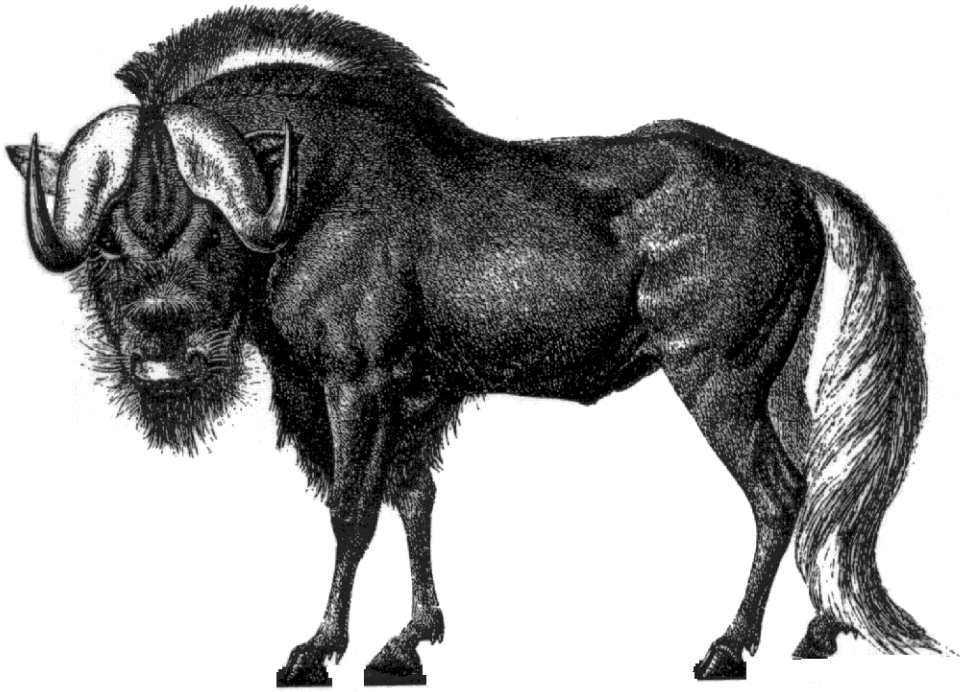
FernUniversität
Gesamthochschule in Hagen

Universitätsrechenzentrum



Gnu Emacs — eine Einführung und ein bißchen mehr ...

Manuela Jürgens
Abt. Wiss. Anwendung

27. März 1997

A/028/0194

Inhaltsverzeichnis

1 Einleitung

Der EMACS-Editor ist zur Zeit wohl einer der mächtigsten Editoren unter Unix überhaupt. Er ist nicht nur ein Editor, sondern stellt eigentlich eine komplette Arbeitsumgebung zur Verfügung, natürlich mit der Möglichkeit Text zu editieren, aber auch Dateien zu kopieren, umzubenennen und zu löschen, Programme zu kompilieren, TEX und LATEX-Texte zu erstellen und eine Syntaxprüfung vorzunehmen, Mails zu verschicken und zu empfangen, Tastaturbelegungen und Makros zu definieren; daneben bietet der Editor ein bißchen Unterhaltung (u.a. einen Zauberer und einen Psychotherapeuten), bei einem X-fähigen Endgerät eine Bedienung mit Fenstertechnik und vieles vieles mehr. Im Prinzip kann man morgens den EMACS aufrufen und muß ihn nie wieder verlassen. Diese Broschüre beschreibt hauptsächlich die wichtigsten Editorfunktionen, wird aber in den letzten Kapiteln noch einen Ausblick auf die weiteren umfangreichen Nutzungsmöglichkeiten geben.

2 Ein wenig Historie

Der Grundstein zum EMACS-Editor wurde bereits 1975 von Richard Stallman gelegt; es handelt sich EDV-mäßig gesehen also um ein „uraltes“ Produkt. Richard Stallman entwickelte am MIT eine Makro-Sammlung für den TECO-Editor und nannte diese „Editing Macros“, abgekürzt „EMACS“. Mit Gründung der „Free Software Foundation[1]“ und dem „GNU-Projekt[2]“ wurde diese Makro-Sammlung ständig erweitert und verbessert und als EMACS-Editor verbreitet. Zur Zeit der Broschürenerstellung ist die Version EMACS 19.21 vom 16. November 1993 im FernUni-Rechenzentrum installiert.

3 Ein paar Gundlagen und Begriffe am Anfang

Einige ungeduldige Leser möchten sicherlich sofort die ersten Kommandos kennenlernen und eingeben. Das dürfen Sie natürlich auch: lesen Sie dazu im Kapitel 4.2 auf Seite 10 weiter. Trotzdem versäumen Sie es bitte nicht, auch dieses Kapitel in einer ruhigen Minute durchzulesen, da hier einige grundlegende Begriffe erläutert werden, die in auf den folgenden Seiten benutzt werden.

Lassen Sie sich nicht abschrecken, wenn Sie anfangs den Eindruck haben, EMACS sei viel zu kompliziert und man könne sich die vielen Tastenkombinationen nicht merken. Wenn Sie sich Zeit nehmen und die einzelnen Befehle der Reihe nach üben, werden Sie sich auch an diesen Editor gewöhnen[3] und nach einer Weile seine komfortablen Vorteile zu schätzen wissen.

[1] Eine Gemeinschaft von Software-Entwicklern, die ihre Produkte frei, also für jedermann verteilbar und benutzbar, zur Verfügung stellt

[2] GNU=GNU's Not UNIX

[3] Wir haben uns ja auch alle an den VI gewöhnt

3.1 Kommandoeingaben

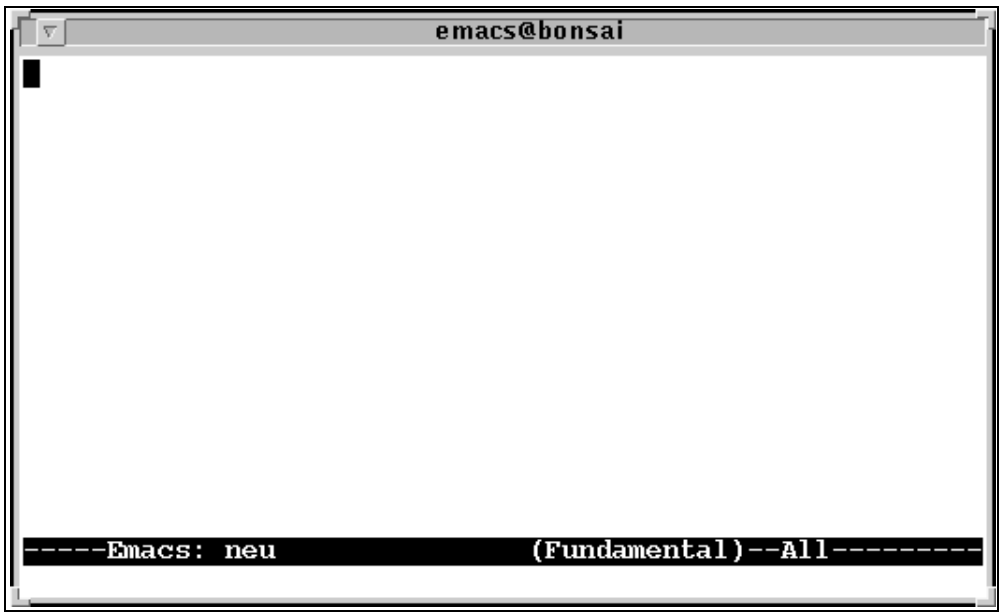
Zwei besondere Tasten werden zur Kommandoeingabe benötigt. Es sind dies die CTRL - oder auch STRG -Taste und die ESC[4]-Taste. Die CTRL -Taste wird dabei immer gleichzeitig mit der nachfolgenden Taste gedrückt, während die ESC -Taste zuerst losgelassen wird und anschließend die Befehlstaste gedrückt wird. Das heißt z.B.:

CTRL+h	CTRL Taste drücken und festhalten und dann den Buchstaben h drücken
ESC+x	ESC Taste drücken und loslassen; anschließend den Buchstaben x drücken.

3.2 Aufruf und Beenden des Emacs

EMACS wird aufgerufen, indem man `Emacs` gefolgt von dem Namen einer Datei eintippt. Eine bereits existierende Datei wird angezeigt, ansonsten erstellt EMACS eine neue leere Datei.

```
Emacs neu
```



Um EMACS wieder zu verlassen gibt es zwei Möglichkeiten:

1. durch Eingabe von CTRL+x CTRL+c (save-buffers-kill-emacs)
   Wurde das File verändert, so fragt EMACS automatisch nach, ob diese Änderungen abgespeichert werden sollen. Beantwortet man die Frage mit `y` (für „yes“), so wird der Editor verlassen und alle Änderungen werden gespeichert. Antwortet man mit `n` (für no), so muß man diesen Entschluß noch ein weiteres mal bestätigen. Lesen Sie aber unbedingt den Text der Frage genau durch; zur Bestätigung müssen Sie nun nämlich ein `yes` eingeben.

[4] Allgemein ist in der EMACS-Dokumentation und dem Online-Help von der Meta-Taste die Rede; bei uns ist das die ESC-Taste

2. soll der Editor nur temporär verlassen werden, so gelingt dies meistens durch Eingabe von CTRL+z. Man gelangt dann zurück in den Unix-Prompt und erreicht zu jedem Zeitpunkt durch Eingabe von **fg**[5] wieder die Stelle im EMACS an dem man ihn verlassen hat. Arbeiten Sie mit X-Windows, so müssen Sie den EMACS-Aufruf in den Hintergrund schicken[6]. Nur dann haben Sie die Möglichkeit, ihn temporär zu verlassen.

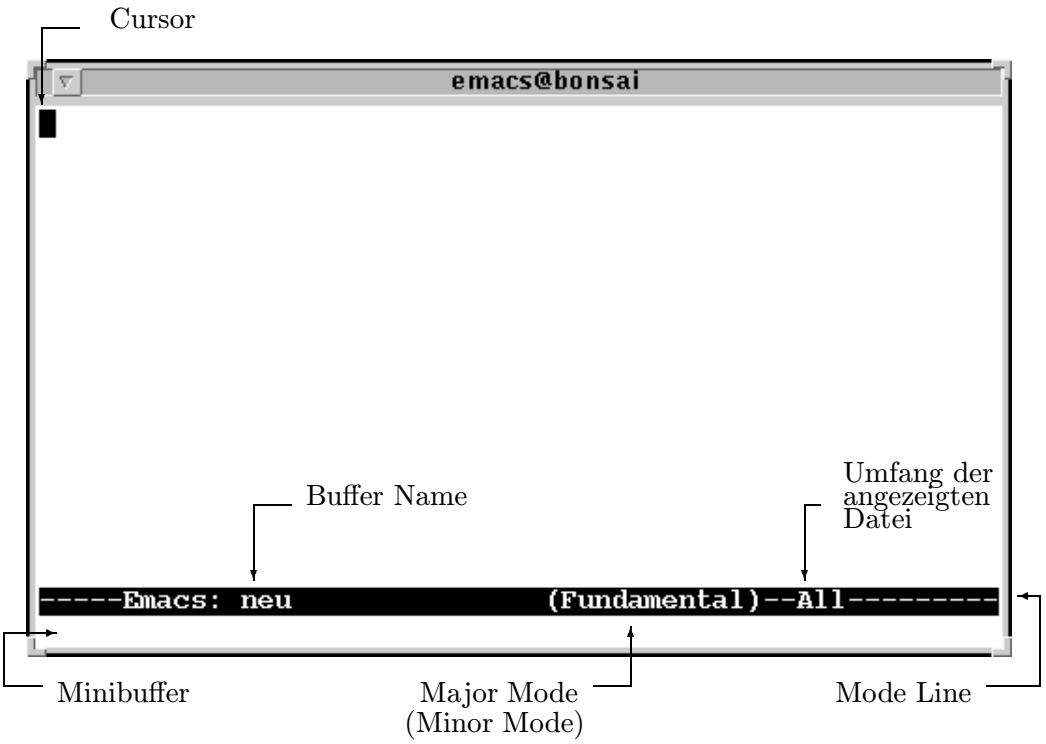
3.3 Sichern der Eingabe

Soll während der Dateibearbeitung die bereits erstellte Eingabe abgespeichert werden ohne daß der Editor verlassen wird, so geschieht das mit dem Befehl CTRL+x CTRL+s (save-buffer).

Soll die Eingabe auf eine Datei mit anderem Namen geschrieben werden, so benutzen Sie hierzu das Kommando CTRL+x CTRL+w (write-file). Sie werden anschließend aufgefordert, einen neuen Dateinamen anzugeben.

3.4 Der Emacs-Bildschirm

Nach Aufruf des Editors erscheint folgender Bildschirm:



[5] fg ist ein Unix-Befehl, der einen im Hintergrund laufenden Prozess wieder in den Vordergrund (foreground=fg) holt

[6] Also ein & an das Ende des Befehls setzen

Er unterteilt sich in einen großen *Arbeitsbereich*, indem sich der *Cursor* befindet und in den der Text eingegeben werden kann und einen Kommando-Bereich, *Minibuffer* genannt, in der untersten Bildschirmzeile. Mit Hilfe der Tasten CTRL und ESC kann in diesen Kommandobereich gewechselt werden. Oberhalb der Kommandozeile befinden sich in einer unterlegten Zeile, der sogenannten *Mode-Line*, der Name der editierten Datei, sowie weitere Informationen auf die an späterer Stelle noch eingegangen werden soll. Einzig die Angaben `All, Top, Bottom` und `nn%` seien hier schon vorgestellt: es handelt sich um einen Hinweis, an welcher Stelle der Datei man sich gerade befindet. `All` bedeutet, daß der komplette Dateiinhalt sichtbar ist, `Top` heißt, wir stehen am Dateianfang, `Bot`, wir sind am Dateiende, und eine prozentuale Angabe teilt uns die Position innerhalb der Gesamtdatei mit.

3.5 Ein Wort zu den Begriffen File und Buffer ...

In der EMACS Literatur, dem Online Help und den Kommandos ist immer wieder die Rede von Files bzw. Buffers. Deshalb hier ein wenig Hintergrundwissen dazu:

EMACS editiert, wie die meisten Editoren auch, niemals die tatsächliche Datei (*File*) auf der Speicherplatte, sondern legt sich automatisch eine Kopie des Files im Hauptspeicherbereich an. Diese Kopie wird als *Buffer* bezeichnet. Durch Eingabe einiger Kommandos (z.B. zum Verlassen des Buffers) wird diese Kopie als File tatsächlich abgespeichert.

Genau wie Files haben auch Buffer Namen und zwar in der Regel die gleichen wie das zugehörige File. Wird EMACS für das File NEU aufgerufen, so wird auch der Buffer den Namen NEU bekommen. Einige wenige Ausnahmen davon existieren, wie z.B. der Buffer mit Namen *SCRATCH* der erstellt wird, wenn EMACS ohne Angabe eines Dateinamens aufgerufen wird.

Im folgenden wird häufiger von Buffer die Rede sein: gemeint ist dann immer die Kopie der Datei, die gerade editiert wird.

3.6 Ein Wort zu dem Begriff Mode ...

EMACS ist, wie bereits angesprochen, etwas mehr als ein Editor. Zum Beispiel stellt er für unterschiedliche Anwendungen (Texterfassung, Programmierung, etc.) sogenannte *Modes* zur Verfügung, in denen spezielle zusätzliche Kommandos benutzt werden können. EMACS unterscheidet zwischen sogenannten *Major Modes* und *Minor Modes*.

Major Modes sind übergeordnete Modes, von denen immer nur genau einer aktiv sein kann. Einige zur Auswahl:

Fundamental Mode	Standard-Mode, der normalerweise nach dem EMACS-Aufruf aktiv ist; (siehe Abbildung auf Seite 4). keine zusätzlichen Kommandos verfügbar
Text Mode	für einfache Texteingabe
Picture Mode	für einfache Strichgrafiken
TEX/LATEX Mode	für Texte, die zur Bearbeitung mit TEX bzw. LATEX vorgesehen sind

C Mode	für C-Programme
FORTRAN Mode	für FORTRAN-Programme
LISP Mode	für LISP-Programme
usw.	

Einige dieser Modes werden in den nachfolgenden Kapiteln noch vorgestellt.

Minor Modes können zusätzlich und zwar mehrere gleichzeitig zu einem ausgewählten Major Mode benutzt werden. Dazu zählen z.B.:

Abbrev Mode	erlaubt das Abkürzen von Befehlen
Auto-Fill Mode	ermöglicht automatischen Zeilenumbruch bei der Texteingabe
Overwrite Mode	überschreibt Zeichen
Auto-save Mode	führt automatisch Sicherungen beim Editieren der Datei durch
usw.	

Welche Modes ausgewählt sind, können Sie der Mode-Line Ihrer Datei entnehmen.

3.7 Kommandos zunächst ganz allgemein

EMACS stellt dem Benutzer tausende[7] von Kommandos zur Verfügung[8]. Die Kommandos haben in der Regel sehr lange Namen, die bei der Eingabe aber abgekürzt werden können und in den allermeisten Fällen sogar durch eine Tastenkombination in Verbindung mit ESC und CTRL ausgeführt werden können[9]. Diese Zuordnung von Befehlen zu Tastenkombinationen wird *Binding* genannt.

Alle Kommandos, und insbesondere die, die nicht durch Tastenfunktion ausgeführt werden können, werden eingegeben durch

ESC+x kommando-name RETURN[10] .

Kennt man das Kommando nicht so ganz genau, so kann man sich der Unterstützung von EMACS gewiß sein: es gibt die Möglichkeit das Kommando soweit einzugeben, wie es bekannt ist und anschließend die TAB-Taste zu drücken.

Beispiel:
Geben Sie ein:

ESC+x ov TAB

so erscheint im unteren Minibuffer

[7] und das ist wörtlich gemeint

[8] die wir uns aber Gott sei Dank nicht alle merken müssen.

[9] Einige davon haben Sie ja auch schon kennengelernt

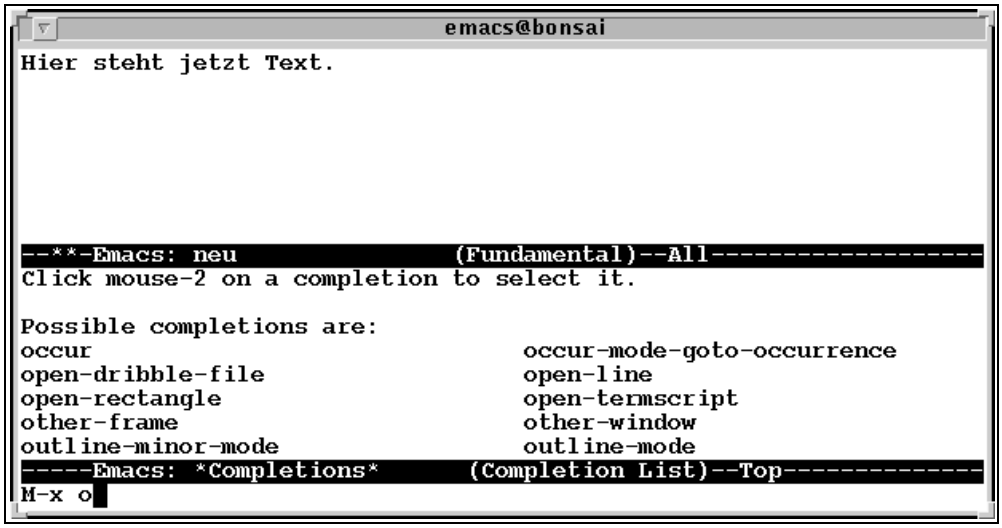
[10] bei den nachfolgenden Befehlen wird immer die Tastenkombination **und** der Kommando-Name angegeben. Sollten Sie die Tastenkombination einmal vergessen haben, so erinnern Sie sich vielleicht noch so ungefähr an den benötigten Kommando-Namen.

M-x overwrite-mode[11]

und Sie müssen nur noch RETURN drücken. Lassen die eingegebenen Buchstaben noch keinen Befehl eindeutig identifizieren, so erscheint ein weiterer Buffer mit allen Befehlen, die mit den eingegebenen Buchstaben beginnen.

Beispiel- Eingabe

ESC+x o TAB



Es erscheint eine sogenannte *Completion List* mit allen Befehlen, die mit `o` beginnen. Den gesuchten Befehl können Sie dann zum Beispiel einfach abtippen und mit RETURN ausführen.

Anmerkung: Momentan kennen Sie noch nicht die Befehle, um sich auch das Ende der Completion List ansehen zu können. Gedulden Sie sich dafür noch bis Kapitel 8.2.

Um alle mögliche Befehle eines Major-Modes zu sehen, geben Sie ein: ESC+x ?. Mit CTRL+g können Sie die angezeigt Liste abbrechen.

3.8 Abbrechen von Emacs-Kommandos

Mit Hilfe von CTRL+g (key-board-quit) kann jedes bereits durch CTRL bzw. ESC begonnene Kommando abgebrochen, sowie jedes Kommando was sich in Ausführung befindet, wieder beendet werden.

Wurde versehentlich zweimal die ESC-Taste gedrückt, so gelangt man in ein Menu, das man bei dem momentanen Kenntnisstand am einfachsten mit einem `n` wieder verläßt.

Wird durch Ausführen eines Kommandos automatisch ein neues Fenster geöffnet, wie z. B. bei den Help-Kommandos, so wird man dieses durch Eingabe von CTRL+x 1 wieder los. Näheres zur Fenstertechnik entnehmen Sie dem Kapitel 8.2 auf Seite 19.

[11] Das M steht für Meta-Key und wird automatisch durch das Drücken der ESC-Taste erzeugt.

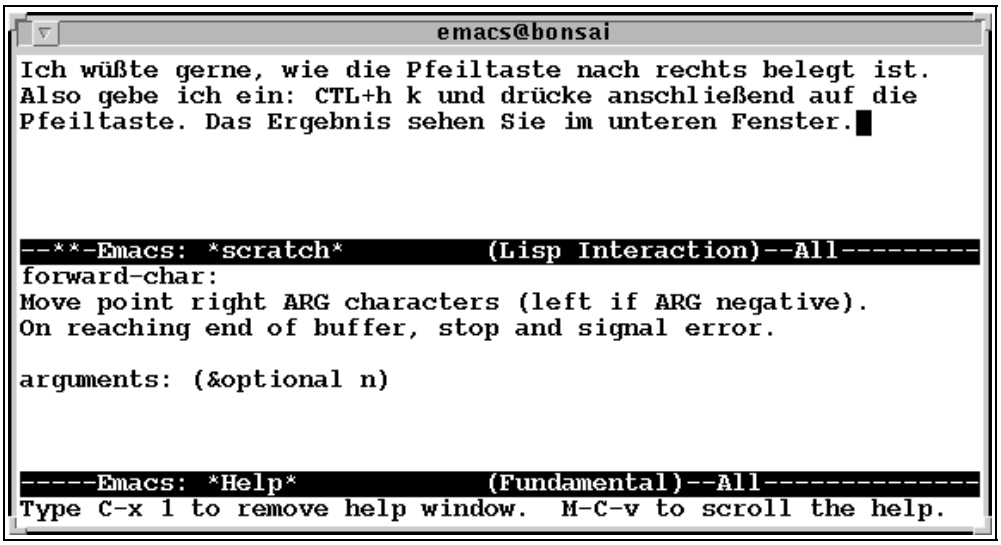
3.9 Hilfe — Emacs hilft bei Problemen

EMACS stellt mehrere Hilfefunktionen zur Verfügung:

1. Um das ausführliche Online-Help zu erhalten kann das Kommando CTRL+h[12] abgesetzt werden. Einige Beispiele:

CTRL+h k	(describe-key) fordert auf, eine beliebige Taste einzugeben. Die Funktion, die mit dieser Taste verbunden ist, wird anschließend erläutert
CTRL+h f	(describe-function) fordert auf, einen Befehl einzugeben,der anschließend erläutert wird.
CTRL+h m	(describe-mode) beschreibt den aktuellen Major-Mode und seine zusätzlichen Kommandos
CTRL+h w	(where-is) listet die zu einem eingegebenen Kommando gehörige Tastenfunktion
CTRL+h b	(describe-bindings) listet alle Kommandos mit ihren zughörigen Tastenkombinationen auf
CTRL+x 1	Verläßt das Hilfefenster

Beispiel: Sie möchten wissen, wie die Pfeiltaste nach rechts belegt ist. Geben Sie dazu ein: CTRL+h k. EMACS fragt Sie anschließend, zu welcher Taste Sie eine Erklärung wünschen; drücken Sie einfach die entsprechende Pfeiltaste und schon erscheint eine Erläuterung in einem eigenen Hilfe-Fenster:

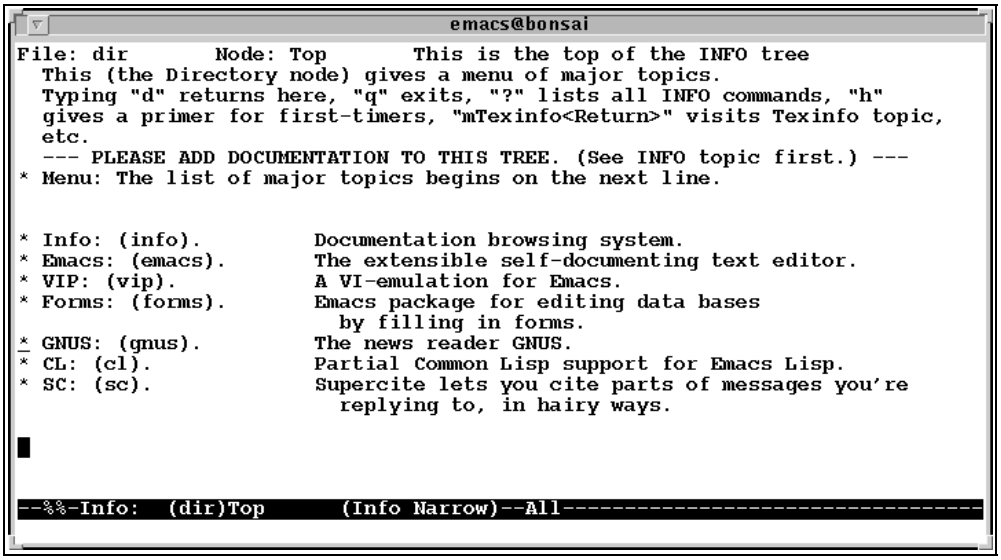


[12] Beachten Sie: CTRL+h werden gleichzeitig gedrückt; alle danach angegebenen Zeichen werden anschließend getippt.

Alle Hilfefenster werden durch die Eingabe CTRL+x 1 wieder verlassen.

2. EMACS stellt ein Tutorial zur Verfügung, mit dessen Hilfe ein EMACS-Anfänger den Editor kennenlernen und seine Funktionen am Bildschirm austesten kann[13]. Diese Funktion wird aufgerufen mit CTRL+h t. Das Tutorial können Sie bei Ihrem bisherigen Kenntnisstand noch nicht so ohne weiteres wieder verlassen. Deshalb hier ein Tip: geben Sie CTRL+x b RETURN ein und Sie befinden sich wieder in Ihrer ursprünglichen Datei. Für eine genauere Erklärung sehen Sie sich das Kapitel 8.2 auf Seite 19 an.

3. Schließlich gibt es auch noch ein Info-System, in dem unter anderem weitere EMACS-Dokumentationen zu finden sind und das über CTRL+h i angewählt werden kann. Es erscheint ein neuer Buffer, in dem durch Eingabe von `m` einer der angelisteten Menu-Punkte ausgewählt werden kann. Die Eingabe `q` verläßt den Buffer wieder.



Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+x CTRL+c	save-buffers-kill-emacs	EMACS verlassen
CTRL+z	suspend-emacs	EMACS unterbrechen
CTRL+x CTRL+s	save-buffer	Sichern der Eingabe
CTRL+x CTRL+w	write-file	Sichern der Eingabe auf eine andere Datei
CTRL+g	keyboard-quit	laufende Kommando abbrechen
CTRL+h	help-command	Help aufrufen
CTRL+x 1	delete-other-windows	Help-Fenster wieder verlassen

Genug geredet, jetzt geht's los ...

[13] Es ist zwar kein richtiges Lernprogramm — aber immerhin. Sie sollten es versuchen

4 Texteingabe

Nachdem EMACS aufgerufen wurde, befindet sich der Cursor im Arbeitsfenster und es kann direkt mit der Texteingabe begonnen werden. Für die ersten Übungszwecke nehmen Sie sich einfach ein spannendes Buch, tippen einige Seiten ab und probieren die im folgenden vorgestellten Kommandos gleich aus. So prägen sie sich am besten ein.

4.1 Automatischer Umbruch am Zeilenende

Normalerweise zählt EMACS alles zu einer Zeile, solange nicht die Return-Taste gedrückt wird. Beim Schreiben von Text-Dateien empfiehlt es sich, eine Einstellung vorzunehmen, in der EMACS einen automatischen Zeilenumbruch durchführt, sobald ein Wort nicht mehr ganz in eine Zeile paßt. Dies gelingt Ihnen mit der Eingabe

ESC+x auto-fill-mode RETURN.

In der Mode-Line am unteren Rand des Bildschirms erscheint dann der Begriff `Fill`. Um diesen Modus wieder auszuschalten geben Sie das gleiche Kommando einfach ein zweites Mal ein.

4.2 Bewegen im Emacs

Es gibt im Prinzip zwei Möglichkeiten, den Cursor innerhalb des EMACS zu bewegen:

1. Hat man Glück, dann sind die geläufigen Pfeiltasten so belegt, daß mit ihrer Hilfe der Cursor in alle Richtungen bewegt werden kann. Probieren Sie es einfach mal aus. Die Tasten BILD↑ und BILD↓ funktionieren dann möglicherweise ebenso wie die Tasten POS1[14], ENDE[15], EINFG und ENTF in der bereits bekannten Weise.

2. Hat man nicht ganz so viel Glück, dann kann man den Cursor aber auch mit Hilfe von Tastenkombinationen bewegen[16]. Ansonsten können Sie sich zu einem späteren Zeitpunkt Ihre Tasten auch selbst belegen (siehe dazu Kapitel 10 auf Seite 24).

Es gibt verschiedene Möglichkeiten, sich durch eine Datei zu bewegen: u.a. zeichenweise, zeilenweise oder auch seitenweise.

Cursorbewegungen innerhalb einer Bildschirmseite

Um den Cusor zeilen- bzw. zeichenweise zu bewegen benutzt man die Kommandos `previous-line` (eine Zeile zurück), `next-line` (eine Zeile vorwärts), `backward-line` (ein Zeichen zurück) und `forward-line` (ein Zeichen vorwärts). Diese Befehle können nun über entsprechende Tastenkombinationen ausgeführt werden. Dazu diene folgendes Bild:

[14] springt an den Anfang der Datei

[15] springt an das Ende der Datei

[16] Lesen Sie sich diese Vorgehensweise auch dann durch, wenn bei Ihnen die Cursor-Tasten belegt sind. Vielleicht kommen Sie ja mal in die Verlegenheit ...

$$\begin{array}{ccc} & previous-line, \boxed{\text{CTRL}}+p & \\ & \vdots & \\ backward-char, \boxed{\text{CTRL}}+b \ldots & Cursor-Position & \ldots forward-char, \boxed{\text{CTRL}}+f \\ & \vdots & \\ & next-line, \boxed{\text{CTRL}}+n & \end{array}$$

Die obigen Kommandos genügen im Prinzip schon für die ersten Cursorbewegungen. Um allerdings ein schnelles Positionieren zu ermöglichen, gibt es weitere Befehle mit denen man wort-, zeilen oder absatzweise mit dem Cursor springen kann. Hier sollen jedoch zunächst nur die gebräuchlichsten vorgestellt werden.

CTRL+a	springt an den Zeilen**a**nfang (beginning-of-line)
CTRL+e	springt an das Zeilen**e**nde (end-of-line)
ESC+<	springt an den Dateianfang (beginning-of-buffer)
ESC+>	springt an das Dateiende (end-of-buffer)

Seitenweise Blättern

Um sich seitenweise durch den EMACS zu bewegen, genügen die folgenden Befehle:

CTRL+v	eine Bildschirmseite vorwärts blättern (scroll-up)
ESC+v	eine Bildschirmseite rückwärts blättern (scroll-down)

4.3 Einfügen und Löschen von Zeichen, Zeilen und mehr

Einfügen und Übertippen von Zeichen Sollen mitten im Text Zeichen eingefügt werden, so muß zunächst der Cursor an die entsprechende Stelle bewegt werden. Alle Zeichen, die anschließend getippt werden, werden automatisch eingefügt. Zum Überschreiben von Zeichen schaltet man mit dem Befehl ESC+x overwrite-mode[17] in den Überschreibe-Modus. In den Einfüge-Modus gelangt man durch nochmalige Eingabe desselben Kommandos[18].

Löschen von Zeichen Es gibt mehrere Möglichkeiten, einzelne Zeichen zu löschen: ein gerade falsch eingetipptes Zeichen wird am einfachsten durch Betätigen der BACKSPACE-Taste[19] gelöscht.

Um ein Zeichen mitten aus einem Text zu entfernen, positioniert man den Cursor auf das zu löschende Zeichen und drückt die Tastenkombination CTRL+d (delete-character). Weitere Möglichkeiten wortweise vorwärts und rückwärts zu löschen existieren, werden aber hier nicht vorgestellt.

[17] Sie erinnern sich, daß man das auch abkürzen kann? Sonst blättern Sie noch einmal zurück auf Seite 6

[18] Versuchen Sie es auf jeden Fall auch einmal mit der EINFG-Taste. Vielleicht ist sie ja belegt; dann können Sie mit dieser Taste wie gewohnt zwischen Einfügen und Überschreiben wechseln

[19] Diese Taste ist hoffentlich mit der richtigen Funktion belegt, nämlich mit `backward-delete-char-untabify`

Einfügen von Zeilen Um eine Zeile einzufügen kann einfach die RETURN-Taste benutzt werden. Befindet sich der Cursor am Anfang einer Zeile, so wird die neue Leerzeile vor der aktuellen Zeile eingefügt, befindet sich der Cursor am Ende einer Zeile, so wird die Leerzeile danach eingefügt.

Teilen und Zusammenfügen von Zeilen Um eine Zeile zu teilen, wird der Cursor an die gewünschte Teilstelle positioniert. Nach Betätigen der RETURN-Taste wird hier automatisch eine zweite Zeile begonnen. Zum Zusammenfügen von Zeilen wird der Cursor am Ende einer Zeile positioniert. Durch die Tastenkombination CTRL+d wird die nächste Zeile mit der aktuellen Zeile zusammengefügt.

Löschen einzelner Zeilen Mit dem Kommando CTRL+k (kill-line) wird der Rest der Zeile ab Cursorposition gelöscht. Positioniert man den Cursor am Anfang einer Zeile und betätigt CTRL+k, so wird zunächst der Inhalt der Zeile gelöscht. Erst nach nochmaligem CTRL+k wird die Zeile gelöscht. Fazit: Um eine Zeile komplett zu löschen gibt man ein: CTRL+k CTRL+k.

Gelöschte Zeilen sind zunächst nicht ganz und gar gelöscht, sondern gelangen in einen sogenannten *Kill-Ring*. Hier wird alles gesammelt, was in *aufeinanderfolgenden Schritten* gelöscht wird. Soll die Löschung rückgängig gemacht werden, so genügt es, einmal CTRL+y (yank)[20] einzugeben und *alle* gelöschten Zeilen erscheinen wieder.

Anmerkung: Einzelne Zeichen, die mit CTRL+d gelöscht werden gelangen nicht in den Kill-Ring.

Markieren von Text Um Teile einzelner Zeilen oder aber auch einen ganzen Block von Zeilen zu löschen, umzubewegen oder zu kopieren, müssen diese Textpassagen zunächst markiert werden. Man positioniert den Cursor dazu auf den Beginn des Textbereiches und drückt die Taste CTRL+@[21] oder einfacher CTRL+SPACE (Leertaste)[22]. Im unteren Minibuffer erscheint der Hinweis: Mark set. Positionieren Sie anschließend den Cursor auf das Ende des Textblockes. Alles was sich nun zwischen der Marke und der Cursorposition befindet ist markiert. Leider wird der markierte Bereich am Bildschirm nicht besonders hervorgehoben. Um eine gewisse Kontrolle über die Markierungen zu haben, kann der Cursor durch Betätigen von CTRL+x CTRL+x (exchange-point-and-mark) zwischen den Markierungspositionen hin und her springen. Insbesondere vor einem Löschbefehl sollte so sicherheitshalber der markierte Bereich noch einmal überprüft werden.

Noch ein Hinweis: ein ganzer Absatz, also alles das, was zwischen 2 RETURN-Tasten steht, kann ganz einfach markiert werden: Der Cursor wird in den Absatz positioniert, Sie drücken ESC+h (mark-paragraph) und automatisch wird ans Absatzende eine Marke gesetzt und der Cursor auf den Absatzanfang positioniert. Die korrekte Markierung können Sie auch hier mit CTRL+x CTRL+x überprüfen. Das gesamte Dokument kann mit CTRL+x h (mark-whole-buffer) markiert werden.

[20] Vorsicht vi-Benutzer: Yank im EMACS bedeutet nicht genau das gleiche wie im vi.
[21] Dafür müssen drei Tasten gedrückt werden: CTL, Alt GR und @
[22] abhängig von der Terminalemulation funktioniert die Kombination CTRL+ SPACE leider nicht immer

Löschen von Zeilenblöcken Um Zeilenblöcke zu löschen, muß der entsprechende Bereich zuvor markiert werden. Dabei muß der Block nicht unbedingt komplette Zeilen umfassen, sondern kann auch mitten in einer Zeile beginnen oder enden. Der so markierte Bereich wird mit dem Befehl CTRL+w (kill-region) gelöscht. Die gelöschten Zeilen wandern zunächst wieder in den Kill-Ring und können mit CTRL+y wieder rekonstruiert werden.

Umstellen von Zeilenblöcken Zum Umstellen von Zeilen werden diese zuerst wie oben beschrieben in den Kill-Ring gelöscht. Danach bewegen Sie den Cursor an die gewünschte Stelle und drücken CTRL+y. Der Inhalt des Kill-Rings wird dann an der momentanen Cursorposition eingefügt. Mehrmaliges Einfügen an verschiedenen Stellen ist hiermit natürlich auch möglich.

Kopieren von Zeilen Um Zeilen zu kopieren werden sie zunächst markiert und anschließend mit ESC+w (copy-region-as-kill) in den Kill-Ring kopiert. Bewegen Sie den Cursor anschließend an die neue gewünschte Stelle im Text und drücken Sie CTRL+y um den Inhalt des Kill-Rings einzufügen. Die Vorgehensweise ist also dem Umstellen von Absätzen sehr ähnlich.

4.4 Abbrechen und Rückgängigmachen von Kommandos

Abbrechen von Kommandos Jedes bereits eingegebene oder in Ausführung befindliche Kommando kann abgebrochen werden mit CTRL+g (keyboard-quit).

Rückgängigmachen von Änderungen Fast alle Änderungen in einem Text können durch Eingabe von CTRL+x u (advertised-undo) wieder rückgängig gemacht werden. Mehrfache Eingabe dieses Kommandos macht der Reihe nach bereits ausgeführte Änderungen rückgängig.

4.5 Einige Tricks

Die oben vorgestellten Kommandos sind zunächst einmal die grundlegenden Befehle, die man im EMACS kennen sollte. Trotzdem kann es nicht schaden, die nachfolgenden Absätze zu lesen, die zusätzliche Kommandos zum schnellen Editieren vorstellen:

Austauschen von Zeichen Beim schnellen Eintippen von Text, ergeben sich häufig Fehler in der Reihenfolge der Buchstaben. Ein Austauschen der Buchstaben können Sie dann mit dem Befehl CTRL+t (transpose-chars) vornehmen.

Beispiel:

Eingegebener Text:	Nach Absetzen von CTRL+t:
Und der Schurke rief: Nien!	Und der Schurke rief: Nein!

Der Cursor wird unter den zweiten der auszutauschenden Buchstaben positioniert, danach drückt man CTRL+t und die Buchstaben sind ausgetauscht.

Austauschen von Wörtern Wird der Cursor auf das Blank zwischen zwei Wörtern positioniert und anschließend ESC+t (transpose-words) gedrückt, so werden die Wörter rechts und links vom Cursor vertauscht.

Austauschen von Zeilen Sollen zwei Zeilen vertauscht werden, so muß der Cursor irgendwo in der zweiten Zeile plaziert werden und danach CTRL+x CTRL+t (transpose-lines) gedrückt werden. Die beiden Zeilen werden dann vertauscht.

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+f	forward-char	Cursor ein Zeichen nach rechts
CTRL+b	backward-char	Cursor ein Zeichen nach links
CTRL+p	previous-line	Cursor eine Zeile nach oben
CTRL+n	next-line	Cursor eine Zeile nach unten
CTRL+a	beginning-of-line	Cursor an den Anfang der Zeilen
CTRL+e	end-of-line	Cursor an das Ende der Zeile
CTRL+v	scroll-up	eine Seite vorwärts blättern
ESC+v	scroll-down	eine Seite rückwärts blättern
ESC+<	beginning-of-buffer	Cursor auf den Dateianfang
ESC+>	end-of-buffer	Cursor auf das Dateiende
CTRL+d	delete-char	Zeichen unter dem Cursor löschen
CTRL+k	kill-line	Zeile vom Cursor bis zum Zeilenende löschen
CTRL+w	kill-region	markierten Bereich löschen
ESC+w	copy-region-as-kill	markierten Bereich kopieren
CTRL+y	yank	Inhalt des Kill-Rings zurückholen
CTRL+@ CTRL+SPACE	set-mark-command	Markierung anbringen
CTRL+x CTRL+x	exchange-point-and-mark	Cursor zwischen Anfang und Ende der Markierung bewegen
CTRL+g	keyboard-quit	laufendes Kommando abbrechen
CTRL+x u	advertised-undo	Rückgängigmachen von Kommandos
CTRL+t	transpose-chars	Zwei Zeichen vertauschen
ESC+t	transpose-words	Zwei Wörter vertauschen
CTRL+x CTRL+t	transpose-lines	Zwei Zeilen vertauschen

5 Suchen und Ersetzen von Zeichenketten

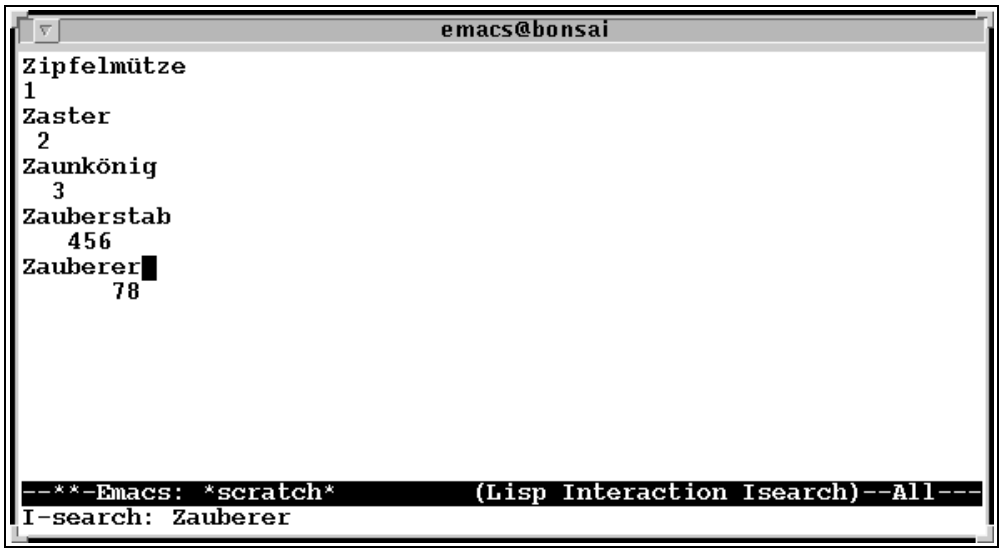
5.1 Suchen von Zeichenketten

EMACS kennt verschiedene Möglichkeiten, um nach Zeichenketten zu suchen. Vorgestellt werden soll hier jedoch nur die inkrementelle Suche.

Die Eigenheit der inkrementellen Suche ist, daß der Suchvorgang bereits mit Eingabe des ersten Buchstabens begonnen wird. Der Cursor positioniert automatisch auf den ersten gefundenen Buchstaben, der mit dem eingegebenen übereinstimmt. Beim Eintippen weiterer Buchstaben wird auch der Cursor fortlaufend auf übereinstimmende Zeichenketten plaziert.

Der inkrementelle Suchvorgang wird durch CTRL+s suchbegriff (isearch-forward) begonnen.

Beispiel: Sie suchen nach der Zeichenkette `Zauberer`. Bereits nach Eingabe von CTRL+s Z positioniert der Cursor auf dem ersten zu findenden `Z`. Geben Sie anschließend das `a` ein, so wird auf das nächste `Za` vorgerückt, nach Eingabe von `u` befindet sich der Cursor unter der Zeichenkette `Zau` usw. Haben Sie bei der Eingabe des Suchbegriffes ein falsches Zeichen getippt, so können Sie es jederzeit mit der BACKSPACE-Taste korrigieren. Nachdem die gesuchte Stelle gefunden wurde, kann die Suche mit ESC abgebrochen werden.



Beim Starten der inkrementellen Suche wird automatisch eine Marke an die Cursorposition gesetzt, an der die Suche beginnt. Um an diese Stelle zurück zu gelangen, benutzten Sie genau wie beim Markieren von Bereichen die Tastenkombination CTRL+x CTRL+x.

Durch wiederholte Eingabe von CTRL+s kann die Suche weitergeführt werden.

Eine Suche rückwärts im Text können Sie mit CTRL+r (isearch-backward) durchführen, das genau wie CTRL+s funktioniert.

Weitere Suchmöglichkeiten, wie z.B. einfache Suche (also nicht inkrementell), Wortsuche und die Suche mit regulären Ausdrücken existieren.

5.2 Ersetzen von Zeichenketten

Sehr häufig wird das Suchen und Ersetzen von Zeichenketten miteinander kombiniert. In der Regel werden zwei verschiedene Ersetzungsmethoden benutzt: generelles Ersetzen aller gefundenen Suchbegriffe, oder Ersetzen eines Suchbegriffes nach vorheriger Nachfrage beim Anwender.

Ersetzen aller Suchbegriffe Um in einem Text generell jeden gesuchten Begriff durch einen neuen zu ersetzen kann folgende Kommandofolge benutzt werden:

ESC+x replace-string RETURN[23]
Eingabe des Suchbegriffs RETURN
Eingabe des Ersatzbegriffs RETURN

Alle gefundenen Begriffe von der Cursorposition an abwärts werden auf diese Weise ersetzt.

Ersetzen von Suchbegriffen mit Rückfrage Um schrittweise einzelne Zeichenketten zu ersetzen oder auch nicht, kann die folgende Befehlsfolge benutzt werden:

ESC+%
Eingabe des Suchbegriffs RETURN
Eingabe des Ersatzbegriffs RETURN

Der Cursor wird auf der ersten Stelle plaziert, an der die Zeichenkette gefunden wird. Folgende Eingaben seitens des Anwenders sind u.a. möglich:

Taste	Resultat
y	der gefundene String wird ersetzt automatisches Positionieren auf den nächsten String
n	der gefundene String wird nicht ersetzt automatisches Positionieren auf den nächsten String
.	der gefundene String wird ersetzt der Suchvorgang wird beendet
!	die restlichen Strings werden ohne Nachfrage ersetzt
ESC	Abbrechen des Such- und Ersetzungsvorgangs

[23] Erinnern Sie sich, daß lange Kommandos abgekürzt werden können? Sonst sehen Sie auf Seite 6 noch einmal nach!

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+s	isearch-forward	inkrementelle Suche nach Zeichenketten (vorwärts)
CTRL+r	isearch-backward	inkrementelle Suche nach Zeichenketten (rückwärts)
	ESC+x replace-string	Suchen und Ersetzen von Zeichenketten
ESC+%	query-replace	Suchen und Ersetzen von Zeichenketten mit Nachfragen

6 Kopieren und Editieren anderer Dateien

Kopieren anderer Dateien Um den Inhalt einer anderen Datei in den aktuellen Buffer zu kopieren, plazieren Sie zunächst den Cursor an der gewünschten Stelle im Text und drücken CTRL+x i (insert-file). Sie werden danach aufgefordert einen Dateinamen einzugeben. Soll der Inhalt der Datei am Ende des Buffers eingefügt werden, so müssen Sie natürlich zunächst z.B. mit ESC+> ans Ende blättern.

Editieren anderer Dateien Um EMACS für eine andere Datei aufzurufen, müssen Sie nicht erst in die Shell-Ebene wechseln und EMACS anschließend erneut aufrufen. Geben Sie stattdessen den Befehl CTRL+x CTRL+f (find-file) ein. Sie werden dann aufgefordert, den Namen der neuen Datei anzugeben. Sie verlassen die ursprüngliche Datei allerdings nicht wirklich[24] und werden deshalb auch nicht aufgefordert, eventuelle Änderungen zu speichern. Spätestens wenn Sie die neue Datei mit CTRL+x CTRL+c verlassen, werden Sie zum Abspeichern der Änderungen in beiden Dateien aufgefordert.

Hinweis: Bei der Eingabe des Dateinamens bietet EMACS Ihnen wieder den Service, nur die ersten Buchstaben des gewünschten Namens einzugeben. Drücken Sie RETURN und die Anfangsbuchstaben waren eindeutig, so wird der Editor für diese Datei aufgerufen. Ansonsten bekommen Sie eine Auswahl über alle infrage kommenden Dateien.

Haben Sie sich bei der Eingabe des Dateinamens vertan, so können Sie mit CTRL+x CTRL+v weitere Alternativdateien aufrufen.

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+x i	insert-file	Einbinden anderer Dateiinhalte
CTRL+x CTRL+f	find-file	EMACS für eine andere Datei aufrufen
CTRL+x CTRL+v	find-alternate-file	Aufrufen einer Alternativ- Datei

[24] Genaueres dazu erfahren Sie im nächsten Kapitel

7 Das wars ... oder möchten Sie noch mehr?

In den vorigen Kapiteln haben Sie nun schon die wichtigsten Kommandos kennengelernt, um Ihre Dateien vernünftig editieren zu können. Aber es gibt im EMACS noch viel viel mehr schöne Dinge, die Ihnen nicht vorenthalten werden sollen. Beginnen wir also hier mit den Kapiteln, die Sie nicht unbedingt sofort lesen müssen.

8 Arbeiten mit Buffers und Fenstern

8.1 Arbeiten mit mehreren Buffers

Sie haben bereits gelesen, daß Sie mit dem Befehl CTRL+x CTRL+f (find file) EMACS für eine andere Datei aufrufen können. Dabei wird der aktuelle Buffer mit einem Buffer für die neue Datei überlagert. Sie haben dadurch die Möglichkeit, verschiedene Dateien in verschiedenen Buffern gleichzeitig zu halten und beliebig zwischen ihnen hin und her zu wechseln. Dabei liegen die Buffer wie in einem Stapel übereinander, so daß immer nur einer sichtbar ist.

Anlegen von Buffers Einen neuen Buffer für eine andere Datei erstellen Sie durch Eingabe von CTRL+x b (switch-to-buffer). Sie werden dann aufgefordert, einen Buffernamen anzugeben. Aber beachten Sie: ein Buffer ist nicht identisch mit einer Datei. Wenn Sie also in einen neuen Buffer gelangen, so ist er zunächst immer leer. Erst wenn Sie eine Datei mit CTRL+x CTRL+f laden, wird der Buffer belegt. Auf diese Art und Weise können Sie beliebig viele Buffer bereit halten. Um zu einem anderen Buffer zu gelangen, geben Sie wieder den Befehl CTRL+x b mit dem Namen des gewünschten Buffers ein.

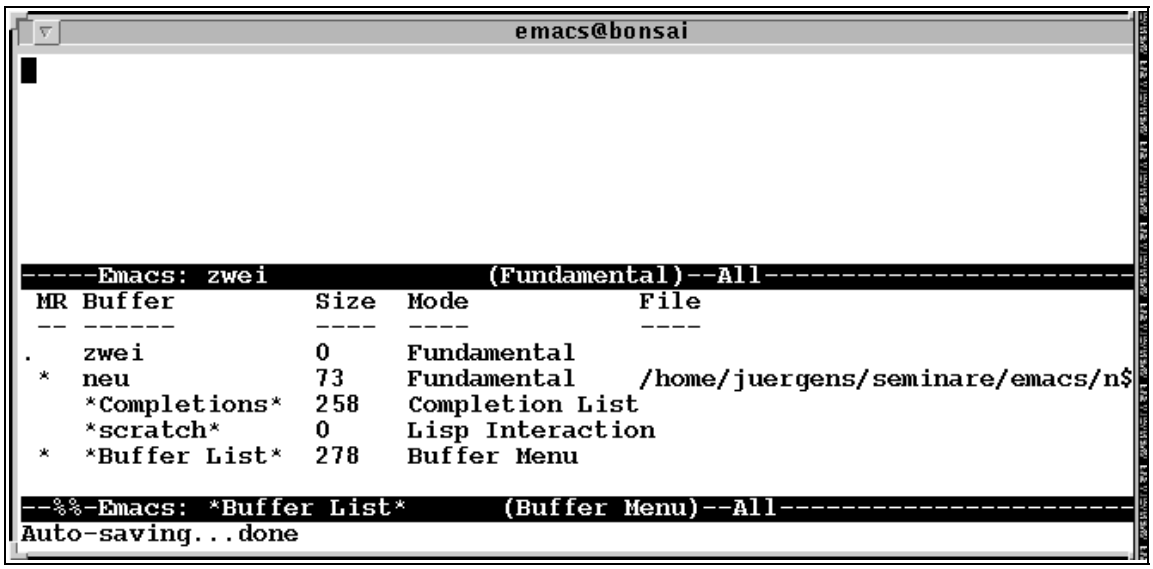
Abspeichern von Buffers Um die Änderungen in den unterschiedlichen Buffers abzuspeichern benutzen Sie den Befehl CTRL+x s (save-some-buffer). EMACS fordert Sie anschließend für jeden einzelnen Buffer auf, sich zum Abspeichern zu entscheiden oder auch nicht. Mit CTRL+x CTRL+s (save-buffer) wird nur der gerade aktuell angezeigte Buffer gesichert.

Löschen von Buffers Um einzelne Buffer wieder zu löschen gibt es zwei Möglichkeiten:

1. der aktuelle Buffer kann gelöscht werden mit dem Befehl CTRL+x k (kill-buffer). Sollte der Buffer Änderungen aufzeigen, so haben Sie noch die Möglichkeit, den Löschvorgang zu stoppen, Ihre Änderungen mit CTRL+x CTRL+s zu speichern und anschließend CTRL+x k zu drücken.

2. Um mehrere Buffer zu löschen benutzen Sie den Befehl ESC+x kill-some-buffers. Sie gelangen dann der Reihe nach in die einzelnen Buffer und werden aufgefordert, sich für oder gegen das Löschen zu entscheiden.

Buffer-Liste Da man recht schnell den Überblick über alle geöffneten Buffer verliert, können Sie sich mit CTRL+x CTRL+b (list-buffers) eine Liste aller Buffers in einem neuen Fenster (mehr zu Fenstern im nächsten Kapitel) erzeugen.
Beispiel:



Ein Sternchen vor dem Buffer deutet an, daß dieser Buffer geändert wurde.

Sie können sich nun entweder einen Buffer aussuchen und ihn mit den entsprechenden Befehlen auswählen oder Sie arbeiten direkt mit der Buffer-Liste. Dazu setzen Sie den Befehl

CTRL+x o

ab und gelangen mit dem Cursor direkt in das Fenster mit der Buffer-Liste. Dort können Sie nun den Cursor nach oben und unten bewegen und zwar mit den bereits bekannten Kommando CTRL+n und CTRL+p bzw. den Pfeiltasten. In dem Kommando-Bereich, in dem sich der Cursor befindet, können eine Reihe von Befehlen abgesetzt werden, von denen Sie einige der nachfolgenden Liste entnehmen können.

Befehl	Bedeutung	Ausführung
d	Markiert den Buffer zum Löschen	wenn x gedrückt wird
s	Markiert den Buffer zum Sichern	wenn x gedrückt wird
u	Markierung zurücknehmen	sofort
1	Wechseln in den Buffer	sofort

Um das Fenster mit der Buffer-Liste wieder loszuwerden, verlassen Sie es zunächst mit CTRL+x o und drücken dann Sie CTRL+x 1. Näheres dazu erfahren Sie im folgenden Kaptitel.

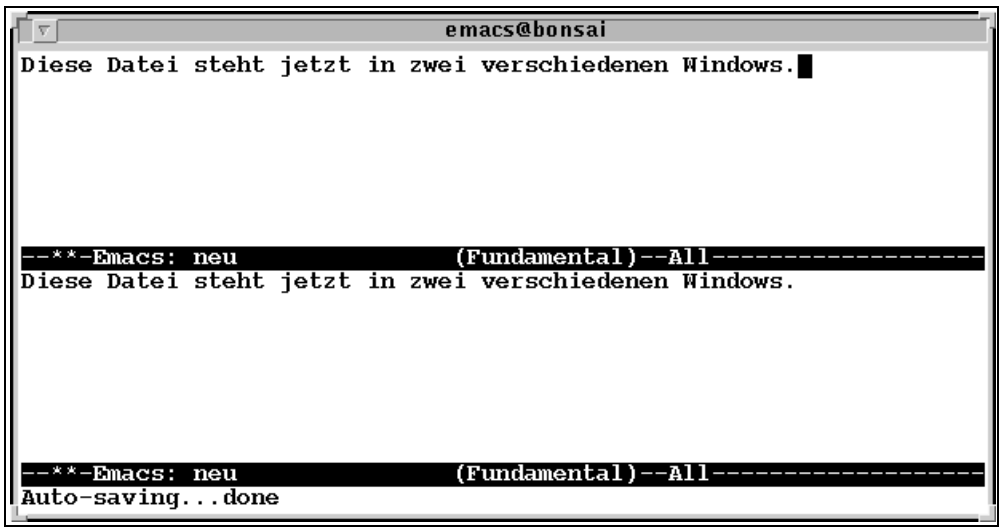
8.2 Arbeiten mit Windows

EMACS stellt ein eigenes Fenstersystem zur Verfügung, das man auch dann benutzen kann, wenn man nicht über die X-Windows-Umgebung verfügt (dazu siehe Kapitel 14 auf Seite 35).

Windows sind Bereiche auf dem Bildschirm, in denen EMACS Buffers anzeigen kann. Mehrere Fenster können gleichzeitig geöffnet sein und verschiedene oder auch den gleichen Buffer anzeigen. Je mehr Fenster geöffnet werden, um so kleiner werden sie, da ein Überlappen, wie es zum Beispiel unter X-Windows möglich ist, hier nicht zu realisieren ist.

Editiert werden kann zu einem Zeitpunkt natürlich immer nur in einem Window.

Erstellen von horizontalen Windows Möchten Sie zwei Fenster parallel übereinander angezeigt bekommen, so setzen Sie den Befehl CTRL+x 2 (split-window-vertically) ab. Der Buffer, der bisher nur in einem Window dargestellt wurde wird nun in zwei Fenstern gezeigt.



In jedem Fenster können alle bereits bekannten EMACS-Kommandos benutzt werden. Das heißt, Sie können sich zum Beispiel zwei verschiedene Stellen einer Datei gleichzeitig ansehen.

Soll in dem zweiten Fenster eine andere Datei angezeigt werden, so benutzen Sie zum Teilen des Fensters den Befehl

CTRL+x 4 f dateiname RETURN (find-file-other-window).

In dem neuen Fenster wird automatisch die angegebene Datei angezeigt.

Bewegen zwischen den Windows Um von einem Fenster in ein anderes zu gelangen, benutzen Sie den Befehl CTRL+x o (other-window). Der Cursor springt dann der Reihe nach in die einzelnen Windows, die Sie dann editieren können. Ein direktes Anspringen der Windows ist nicht möglich.

Kommandos in Windows Alle bisher vorgestellten EMACS-Kommandos können in jedem einzelnen Fenster abgesetzt werden. Besonders komfortabel kann so zum Beispiel auch ein Teilstück einer Datei in eine andere Datei kopiert werden: markieren Sie in einem Fenster den zu kopierenden Bereich, übertragen Sie ihn in den Kill-Ring, wechseln Sie dann in ein anderes Fenster und fügen den Inhalt des Kill-Rings dort wieder ein.

Löschen von Windows Um das gerade aktuelle Fenster zu löschen, benutzen Sie den Befehl CTRL+x 0. Um alle Fenster außer dem aktuellen zu löschen, benötigen Sie das Kommando CTRL+x 1.

Eine ganze Reihe weiterer Windows-Kommandos existieren, mit denen man zum Beispiel Kommandos in Fenstern ausführen kann, die gerade nicht aktiv sind, Fensterinhalte vergleichen kann, Fenster nebeneinander anordnen kann, vergrößern kann und vieles mehr.

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+x b	switch-to-buffer	Anzeigen eines Buffers
CTRL+x CTRL+b	list-buffers	Anzeigen der Buffer-Liste
CTRL+x k	kill-buffer	angegebenen Buffer löschen
	ESC+x kill-some-buffers	Alle Buffer löschen
CTRL+x s	save-some-buffers	Buffer sichern
CTRL+x 2	split-windows-vertically	Teilen des Bildschirms in zwei Teile
CTRL+x o	other-windows	Springen in ein anderes Fenster
CTRL+x 0	delete-window	Aktuelles Fenster löschen
CTRL+x 1	delete-other-windows	alle anderen Fenster löschen
CTRL+x 4 f	find-file-other-windows	ins zweite Fenster eine andere Datei laden

9 Schreiben von Makros

EMACS unterstützt das Schreiben von sogenannten Makros, wenn auch nicht unbedingt sehr komfortabel. Ein Makro ist dabei im Prinzip eine Aufzeichnung von nacheinander ausgeführten EMACS-Befehlen. Diese Befehle können unter einem Namen abgespeichert werden und dann durch Aufruf des Namens ausgeführt werden. Ein Makro zu schreiben lohnt sich immer dann, wenn eine Reihe von Befehlen immer wieder in der gleichen Reihenfolge benötigt wird.

Am Ende dieses Kapitels werden Sie sich ein Makro schreiben können, das die Umlautumsetzung in TEX/LATEX-Texten vornimmt.

Nachdem bestimmte EMACS-Einstellungen zur Umlautbenutzung (siehe Seite 31) vorgenommen wurden, können Sie diese in Ihren Texten benutzen[25]. Arbeiten Sie mit dem Textprogramm LATEX (näheres hierzu auf Seite 30) und dem sogenannten `german`-Style, so wird der Umlaut ä intern als "a[26] benötigt. Das bedeutet, nachdem Sie Ihren Text editiert haben und Sie den Editor verlassen wollen, müssen alle Umlaute durch die oben angedeutete Darstellung ersetzt werden. Dafür lohnt es sich, ein Makro zu schreiben. Umgekehrt benötigen Sie ein zweites Makro, das die Darstellung "a nach erneutem Aufruf des EMACS wieder durch ein ä ersetzt.

[25] Das setzt allerdings zusätzlich noch voraus, daß Sie über eine geeignete Terminalemulation zugreifen

[26] Ich weiß: es gibt auch noch die Möglichkeit z.B. mit `umlaute.sty` zu arbeiten; dann entfällt die angesprochene Umlautumwandlung.

Definition eines Makros

Wenn Sie ein Makro aufzeichnen wollen, so geben Sie zunächst

CTRL+x (

ein. Alle nachfolgenden Tastenkombinationen und Befehle werden aufgezeichnet, solange, bis Sie durch Eingabe von

CTRL+x)

das Makro beenden. Eigentlich ganz einfach. Das Problem besteht in der korrekten Eingabe der Tastenkombinationen, die das Makro bilden. Bei jeder fehlerhaften Eingabe wird die Aufzeichnung des Makros sofort beendet und Sie müssen von vorne beginnen. Soll das Makro explizit abgebrochen werden, so geben Sie einfach CTRL+g ein.

Beispieleingabe:

CTRL+x (	Makro starten
ESC+<	Cursor an den Anfang der Datei positionieren
ESC+x replace-string RETURN	Befehl zum Ersetzen von Zeichen aufrufen
ä	zu ersetzendes Zeichen eingeben
"a	Ersatzzeichen eingeben
CTRL+x)	Makro beenden

Obiges Makro setzt also alle ä in "a um. Auf die gleiche Art und Weise können in einem Makro natürlich alle anderen Umlaute ersetzt werden.

Bereits bei der Definition des Makros werden alle eingegebenen Befehle direkt ausgeführt, so daß Sie eine Kontrolle über die Korrektheit der ausgeführten Aktionen haben.

Um das Makro anschließend aufzurufen, benutzen Sie den Befehl

CTRL+x e (call-last-kbd-macro).

Alle Makro-Befehle werden dann automatisch ausgeführt. Ein Makro ist zunächst solange gültig, wie kein neues definiert und EMACS nicht verlassen wird.

Abspeichern von Makros

Um Makros aufzubewahren, müssen Sie einen Namen zugewiesen bekommen und abgespeichert werden. Dazu müssen die nachfolgenden Schritte durchgeführt werden:

1. Definieren Sie zunächst Ihr Makro

2. Geben Sie Ihrem Makro einen Namen mit dem Befehl

   ESC+x name-last-kbd-macro RETURN makroname RETURN

   Das Makro ist nun bis zum Verlassen des Editors verfügbar und kann jederzeit aufgerufen werden mit

   ESC+x makroname.

3. Um das Makro permanent zu speichern, muß es in einer Datei abgelegt werden. Rufen Sie dazu eine neue Datei auf mit

   CTRL+x CTRL+f dateiname RETURN.

   (Am besten benennen Sie Makro und Datei gleich).

4. Nun müssen Sie das Makro noch in die Datei holen mit dem Befehl

   ESC+x insert-kbd-macro RETURN

   EMACS speichert den LISP-Code, der Ihr Makro repräsentiert nun ab[27].

5. Sichern Sie Ihr Makro nun noch mit CTRL+x CTRL+s.

6. Um das Makro in verschiedenen Dateien verfügbar zu machen, muß es erst geladen werden. Das kann man durch den Befehl

   ESC+x load-file RETURN makrodatei RETURN

   erreichen. Zur Ausführung wird das Makro anschließend durch den Befehl

   ESC+x makroname

   gebracht.

Makros, die auf diese Art und Weise erstellt werden, sind natürlich nicht sehr flexibel. Es gibt keine Möglichkeiten der Abfrage oder Erstellung von Schleifenkonstrukten und ähnlich schöne Programmier-Elemente. Um solche Dinge zu benutzen, kommt man nicht umhin sich mit LISP zu befassen, der Sprache, in der EMACS programmiert ist und in die alle Makros automatisch umgewandelt werden. LISP soll jedoch nicht Thema dieser Broschüre sein.

[27] Den Inhalt des Makros in LISP müssen Sie nicht verstehen

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+x (	start-kbd-macro	Makrodefinition starten
CTRL+x)	end-kbd-macro	Makrodefinition beenden
CTRL+x e	call-last-kbd-macro	zuletzt definiertes Makro aufrufen
	ESC+x name-last-kbd-macro	einem Makro einen Namen zuteilen
	ESC+x insert-last-keyboard-macro	Abspeichern des Makros in einer Datei
	ESC+x load-file	Makrodatei laden
	ESC+x makroname	Makro aufrufen

10 Definieren von Tastenbelegungen

Viele Befehle, die hier vorgestellt wurden, sind bereits mit Tastenkombinationen verknüpft. Alle anderen können Sie aufrufen mit ESC+x kommandoname. Um auch solche Befehle oder sogar selbstgeschriebene Makros mit Tastenkombinationen ausführen zu können müssen sogenannte *Bindings* durchgeführt werden. Dazu zunächst einige grundlegende Begriffe:

Alle Tastaturbelegungen werden in sogenannten *keymaps* festgehalten. Die wichtigste keymap ist die *global keymap*. In ihr sind alle grundsätzlichen Tasten definiert und sie steht in allen Major-Modes zur Verfügung. *Prefix keymaps* beinhalten alle Tastenkombinationen, wie zum Beispiel CTRL+x (*ctl-x-map*), CTRL+h (*help-map*), ESC (*esc-map*) usw. Auch diese Tastenkombinationen sind immer verfügbar. Dahingegen existieren *local keymaps*, die nur in den einzelnen Major- und Minor-Modes (z.B. *latex-mode-map, c-mode-map* usw.) verfügbar sind und schließlich gibt es noch *minibuffer keymaps* für den lokalen Buffer-Gebrauch.

Um selbst Tastaturbelegungen zu erstellen, müssen also die entsprechenden keymaps verändert werden. Dazu benutzen Sie die folgenden Kommandos.

ESC+x global-set-key RETURN *taste kommando* RETURN
ESC+x local-set-key RETURN *taste kommando* RETURN

Um zum Beispiel die Taste F1 mit dem Aufruf Ihres eigenen geschriebenen Makros „Umlaute“ zu belegen, geben Sie ein:

ESC+x local-set-key RETURN F1 umlaute RETURN

anschließend brauchen Sie nur noch F1 zu drücken und Ihr Makro wird ausgeführt.

Nun können Sie auch die Pfeiltasten mit den entsprechenden Cursorbewegungen belegen, falls das nicht schon voreingestellt ist. Geben Sie dazu ein:

ESC+x global-set-key RETURN rechtePfeiltastedrücken forward-char RETURN
ESC+x global-set-key RETURN linkePfeiltastedrücken backward-char RETURN

usw.

Leider sind alle Tastenbelegungen nur so lange vorhanden, bis der EMACS wieder verlassen wird. Für permanente Belegungen müssen Sie sich Ihr eigenes Profil erstellen.

11 Erstellen eines Emacs-Profils

Normalerweise ist es nicht nötig, sich in LISP-Programmierung auszukennen. Einige wichtige Grundkenntnisse sind jedoch unumgänglich, will man sich ein eigenes EMACS-Profil erstellen.

Zunächst mal ist jeder EMACS-Befehl eine LISP-Funktion der Form (funktionsname argumente). Wenn Sie zum Beispiel den Befehl CTRL+f absetzen, den Cursor also um ein Zeichen nach rechts bewegen wollen, so verbirgt sich dahinter die LISP-Funktion (forward-char 1).

In Ihrem EMACS-Profil können Sie Tastaturbelegungen vornehmen, Makrobibliotheken laden, Modes verändern und vieles mehr. Das Profil muß sich in der Datei `.emacs` befinden. Diese wird bei jedem EMACS-Aufruf automatisch durchlaufen.

Da in vielen Fällen die .emacs-Datei benutzt wird, um Tastaturbelegungen zu definieren, werden die entsprechenden Befehle hier zunächst vorgestellt:

(define-key keymapname „tastenkombination“ 'commando-name) ganz allgemein kann hiermit eine Tastenkombination der keymap hinzugefügt werden.

(global-set-key „tastenkombination“ 'commando-name) der global-map wird eine Tastenkombination hinzugefügt. Dieser Befehl ist identisch mit (define-key global-map ...).

(local-set-key „tastenkombination“ 'commando-name) für den aktuellen Buffer wird eine Tastenkombination definiert. Dieser Befehl ist identisch mit (define-key local-map ...)

Einige Beispiele:

- Sie möchten die F1-Taste mit dem Aufruf des Hilfe-Tutorial belegen. Schreiben Sie dafür z.B. in Ihr .emacs-File die Anweisung:

  ```
  (global-set-key [kp-f1] 'help-with-tutorial)
  ```

- Sie möchten die Tastenkombination CTRL+z benutzen, um Shell-Kommandos im aktuellen Buffer eingeben zu können:

  ```
  (global-set-key "\C-z" 'shell)
  ```

  Die beiden obigen Beispiele zeigen Ihnen zwei verschiedene Möglichkeiten der Tastaturbelegungen: mit Hilfe der eckigen Klammern `[]`, können Sie zum Beispiel alle Funktionstasten (f1,. . .,f35), sowie die Pfeiltasten (left, up, right, down) belegen. Ein bißchen Ausprobieren müssen Sie möglicherweise die Belegung der Funktionstasten: in einigen Fällen müssen diese als kp-f1 ... kp-f35 angegeben werden. Die Namen aller Tasten und der Funktion mit der sie eventuell belegt sind, können Sie durch Eingabe von

CTRL+h c TastenDruck

ermitteln.

Die Gänsefüßen im zweiten Beispiel ermöglichen Ihnen die Festlegung von Tastenkombinationen z.B. in Zusammenhang mit der CTRL-Taste.

- Zur kurzen Erläuterung der weiteren umfangreichen Einstellungen Ihres .emacs-Files diene das nachfolgende Beispiel:

```
 1 (standard-display-european 1)
 2 (require 'iso-syntax)
 3 (load "~/umlaute")
 4 (global-set-key [right] 'forward-char)
 5 (global-set-key [left]  'backward-char)
 6 (global-set-key [f12] 'umlaute)
 7 (add-hook 'latex-mode-hook '(lambda ()
 8                               (setq tex-dvi-view-command "xdvi *")
 9                               (setq tex-dvi-print-comamnd "dvips *")
10                               (setq tex-open-quote "\"{}‘")
11                               (setq tex-close-quote "\"{}’")
12                               (auto-fill-mode 1)))
```

Erläuterungen:

Zeile 1 und 2: Einstellung der ISO-8859-Norm. Sie können anschließend die Umlaute benutzen[28].

Zeile 3: Laden der Makrobibliothek `umlaute`, die sich im $HOME-Verzeichnis befindet.

Zeile 4 und 5: Belegen der Pfeiltasten nach rechts und links mit der entsprechenden Cursorbewegung.

Zeile 6: Belegen der Funktionstaste F12 mit dem Aufruf des Umlaute-Makros.

Zeile 7 und folgende Hier wird eine Besonderheit des EMACS benutzt und zwar das Arbeiten mit einem sogenannten *Hook*. Hooks sind Variable, deren Inhalt aus LISP-Funktionen besteht, die ausgeführt werden, sobald ein Major-Mode aufgerufen wird. Wird der Inhalt einer Hook-Variablen mit dem Kommandos `add-hook` erweitert, so werden jeweils beim Aufruf des zugehörigen Major-Modes, die angegebenen Kommandos mit ausgeführt.

Im obigen Beispiel wird die Hook-Variable `latex-mode-hook` erweitert, die herangezogen wird, sobald der LaTeX-Mode benutzt wird.

Möchten Sie auch Hooks ändern, so halten Sie sich an die Syntax, die in der ersten Zeile des `add-hook`-Kommandos benutzt wurde. In allen weiteren Zeilen geben

[28] sofern Sie eine geeignete Terminalemulation benutzen, die 8Bit-Code-fähig ist. Können Sie bereits auf der Shell-Ebene außerhalb des Emacs keine Umlaute darstellen, so müssen Sie auch im Emacs auf die Umlaute verzichten.

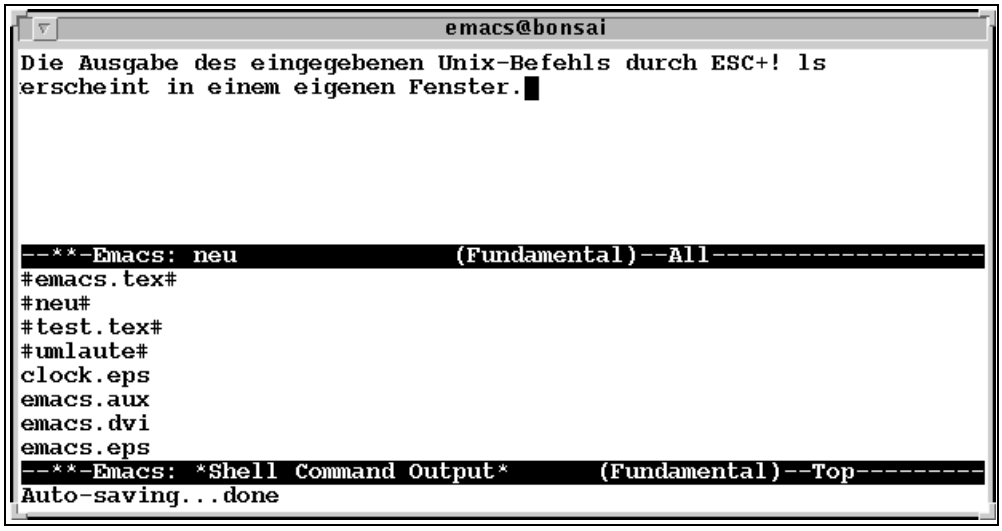
Sie einfach die von Ihnen gewünschten Kommandos ein. In unserem Beispiel werden die Variablen `tex-dvi-view-command` und `tex-dvi-print-command` mit dem tatsächlichen Aufruf des im FernUni-Rechenzentrum installierten Treibers initialisiert (siehe hierzu auch Kapitel 13 auf Seite 30.). Die Variablen `tex-open-quote` und `tex-close-quote` werden auf die gleiche Weise mit den deutschen Anführungszeichen oben und unten belegt. Schließlich wird in der letzten Anweisung noch der *auto-fill-mode* eingestellt, so daß EMACS einen automatischen Zeilenumbruch beim Eintippen Ihres Textes vornimmt.

Dies soll soweit als erster Einstieg in die LISP-Benutzung und damit das Erstellen eines EMACS-Profils genügen. Für Ihre ersten Tests beginnen Sie am besten mit ganz einfachen Einstellungen, um Frustrationen zu vermeiden. Wollen Sie sich mit diesem Thema näher beschäftigen, so kann ich nur auf das GNU EMACS und Lisp Manual verweisen.

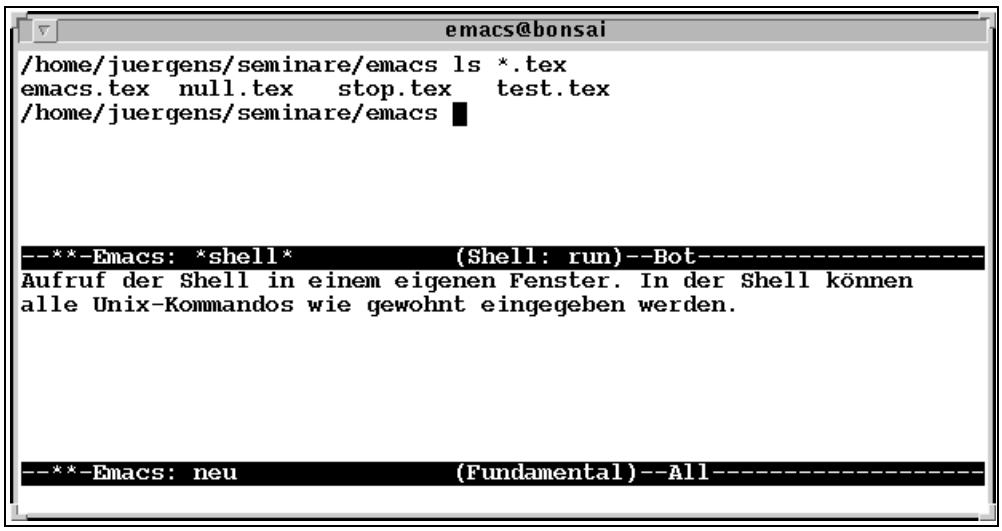
12 Emacs als Arbeitsumgebung

12.1 Ausführen von Unix-Kommandos

Es gibt zwei Möglichkeiten, um aus dem Editor heraus Unix-Kommandos abzusetzen: Zum einen können Sie den Befehl ESC+! zusammen mit einem Unix-Kommando benutzen, der Ihnen automatisch ein zweites Window eröffnet und Ihnen dort die Ausgabe des Unix-Befehls anzeigt.



Zum anderen können Sie einen eigenen Buffer oder auch ein extra Fenster benutzen, um dort direkt in die Shell-Ebene zu wechseln und Unix-Kommandos abzusetzen. Dafür benötigen Sie dann den Befehl ESC+x shell.



In dieser Shell-Ebene können Sie die meisten Unix-Kommandos benutzen, die Ihnen auch sonst zur Verfügung stehen. Sie können aber noch mehr: eigentlich befinden Sie sich ja in einem EMACS-Buffer und haben somit die Möglichkeit, die Kommandos (die ja eigentlich Texteingabe sind) beliebig zu editieren, zu markieren, zu löschen, zu kopieren und so weiter. Vielfältige Einsatzmöglichkeiten stehen Ihnen also zur Verfügung.

Um die Fenster zu aktivieren arbeiten Sie, wie schon gewohnt, mit CTRL+x o.

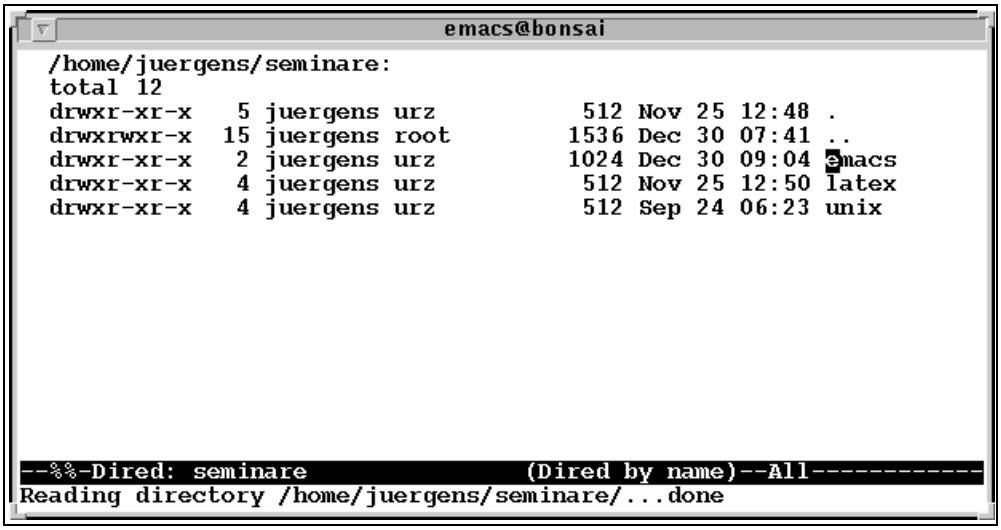
12.2 Arbeiten mit Unix-Verzeichnissen (Dired)

Ein mächtiges und komfortables Kommando im EMACS ist sicherlich der Directory Editor Dired. Er bietet höchsten Komfort zum Ansehen, Editieren, Löschen, Kopieren und Umbenennen von Dateien. Mit Dired können Sie ganz einfach arbeiten, indem Sie EMACS für einen Verzeichnisnamen aufrufen. Alle Dateien und untergeordneten Verzeichnisse werden Ihnen als Liste am Bildschirm angezeigt. In dieser Liste können anschließend Kommandos eingegeben werden, mit deren Hilfe die Dateien und Verzeichnisse manipuliert werden können.

Anhand des nachfolgenden Beispiels werden einige Funktionen des Dired näher erläutert.

Beispiel:

```
emacs /home/juergens/seminare
```



Der Inhalt von EMACS zeigt Ihnen nun im Prinzip die Ausgabe des Unix-Kommandos ls -l an, mit der Möglichkeit, die angegebenen Dateien für irgendeine Aktion auszuwählen. Aber Vorsicht: Sie arbeiten hier nicht mehr mit einem Buffer, der Ihre Dateien zunächst unberührt läßt. Wenn Sie in Dired eine Datei zum Beispiel löschen, ist sie anschließend tatsächlich verschwunden.

Nun aber zum Arbeiten mit der Verzeichnisliste: Der Cursor befindet sich standardmäßig vor oder auf den Dateinamen. Mit Hilfe der Pfeiltasten (bzw. falls nicht belegt CTRL+n und CTRL+p) können Sie sich nach oben und unten bewegen. Um sich beispielsweise den Inhalt einer Datei ansehen zu können, plazieren Sie den Cursor auf den Dateinamen und drücken die Taste e. EMACS wird automatisch aufgerufen. Haben Sie die Taste e bei dem Namen eines Unterverzeichnisses gedrückt, so gelangen Sie in die Dateiliste dieses Verzeichnisses. In die übergeordnete Verzeichnisstruktur gelangen Sie durch Eingabe von q.

Hier nun einige der Möglichkeiten, die Ihnen zur Verarbeitung Ihrer Dateien im Dired zur Verfügung stehen.

Taste	Aktion	Unix-Kommando
e	editieren der Datei	EMACS
C	kopieren der Datei	cp
R	umbenennen der Datei	mv
d	markieren zum Löschen der Datei	rm
u	rückgängigmachen obiger Markierung	
x	löschen aller markierten Dateien	rm und rmdir
v	ansehen von Dateien (view)	more

Obige Kommandos, die unter Unix immer die Eingabe eines zweiten Dateinamens erfordern, wie z.B. der cp-Befehl, können einfach durch Eingabe des entsprechenden Buchstabens angestoßen werden; Sie werden dann interaktiv nach dem erforderlichen Dateinamen gefragt.

Soweit hier einige der vielen Möglichkeiten des Dired.

13 Emacs und TEX/LATEX

Wenn Sie EMACS benutzen um zum Beispiel den Text für ein Dokument zu erstellen, unterstützt der Editor Sie bei der Verwendung der Textverarbeitungssysteme *troff/nroff/groff*, *Scribe* und TEX/LATEX. Eine kurze Vorstellung der erleichterten Texteingabe soll hier für die Erstellung von LATEX-Texten gegeben werden:

Um in den TEX/LATEX-Mode zu schalten geben Sie ein

ESC+x tex-mode RETURN bzw.

ESC+x latex-mode RETURN.

Rufen Sie EMACS für eine Datei mit dem Namen . . .`tex` auf, bzw. existiert bereits der Inhalt der Datei, so erkennt der Editor automatisch die Syntax und ruft den richtigen Major-Mode auf. Welche Funktionen stehen Ihnen nun zur Verfügung?

Überwachung paariger Klammerungen

Fast alle TEX/LATEX-Kommandos werden in der Form `\kommando{argument}` benutzt. Gerade bei der Klammerung treten häufig Flüchtigkeitsfehler auf, die in umfangreichen Texten im nachhinein nur schwer zu lokalisieren sind. Bereits bei der Texteingabe überprüft EMACS deshalb bei jeder schließenden Klammer, wo sich die zugehörige öffnende Klammer befindet. Der Cursor springt für einen kurzen Augenblick genau auf die linke Klammer, die er der rechten Klammer zuordnet. Sehen Sie also auf Ihren Bildschirm und kontrollieren Sie anhand des Cursorsprungs die korrekte Zuordnung. Während des Cursor-Rücksprungs wird die Schreibposition innerhalb des Textes nicht verändert und Sie brauchen auch nicht auf den Cursor zu warten, sondern können einfach weiterschreiben. EMACS gibt allerdings keinerlei Warnung über nicht zugeordnete Klammern, da er zwischen Klammern, die Kommandos betreffen und Klammern, die zu Ihrer Texteingabe gehören naturgemäß nicht unterscheiden kann.

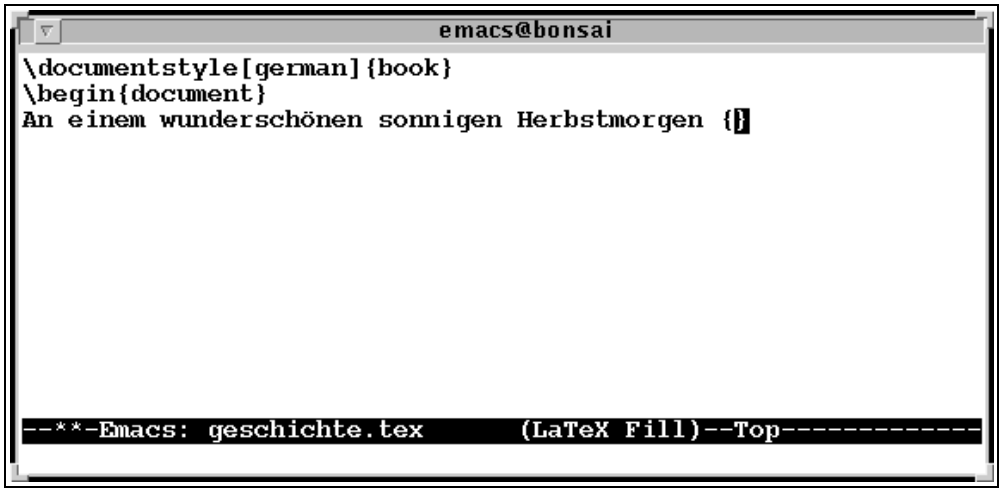
Die oben vorgestellte Überprüfung wird im übrigen auch automatisch für `$`-Zeichen vorgenommen, die ja ebenfalls nur paarig auftreten dürfen.

Der EMACS-Befehl

CTRL+c {[29] (tex-insert-braces)

[29] Für die, die mit auctex arbeiten: Viele der im nachfolgenden vorgestellten Befehle funktionieren bei Benutzung von auctex anders.

beugt dem Fehlen schließender Klammern vor: er erzeugt nämlich gleich ein komplettes Klammernpaar {} und positioniert den Cursor dazwischen, so daß der innere Text sofort geschrieben werden kann.



Eine letzte Möglichkeit der Überprüfung paariger Klammern kann durchgeführt werden mit

ESC+x validate-tex-buffer.

Nachdem der gesamte Text durchsucht wurde, öffnet sich ein zweites Fenster mit der Textstelle, an der EMACS eine fehlende oder überflüssige Klammer vermutet, zusammen mit der Zeilennummer. Mit

ESC+x goto-line zeilennummer

kann dann direkt auf die möglicherweise fehlerhafte Zeile gesprungen werden.

Eingabe von Umlauten

Deutsche Texte enthalten in der Regel auch deutsche Umlaute. Normalerweise sind diese im EMACS nicht verfügbar. Die Definition der Umlaute wird in einer LISP-Datei vorgenommen, die zuvor geladen werden muß. Setzen Sie dazu das Kommando

ESC+x standard

ab, oder erweitern Sie Ihr EMACS-Profil um die Anweisung (standard-display-european 1). Anschließend können Sie die Umlaute, wie auf der Tastatur vorhanden, eingeben[30]. Denken Sie aber daran, daß ein Umlaut für das TEX/LATEX-Programm nicht verständlich ist. Sie müssen vor dem Verlassen der Datei für eine ordnungsgemäße Umsetzung der Umlaute sorgen bzw. mit dem `umlaute`-Style arbeiten. Siehe hierzu auch Kapitel 26 auf Seite 22.

[30] Beachten Sie aber auch hier, daß die Eingabe der Umlaute voraussetzt, daß Sie über eine geeignete Terminalemulation zugreifen

Das Problem der Gänsefüßchen

Die Anführungszeichen ", die sich auf der Tastatur befinden, werden in der Regel nicht für TEX-Texte benötigt. Im Amerikanischen/Englischen werden stattdessen ‘‘ und ’’ benutzt und im Deutschen "‘ und "’. Der TEX-Major-Mode bewirkt, da von Amerikanern entwickelt, bei Eingabe von " eine Ausgabe von ‘‘ bzw. ’’[31]. Das ist einerseits natürlich praktisch, andererseits für deutschsprachige Texte ein Problem. Also was liegt näher, als die "-Taste entsprechend für deutsche Texte zu belegen? Als kleine Übungsaufgabe zum Beispiel. Sollte es Ihnen nicht gelingen hier ein Tip: Sie können das " zwingend erzeugen durch CTRL+q " [32].

Erzeugen von Absätzen

Absätze werden in TEX/LATEX durch Eingabe einer leeren Zeile erzeugt. Statt zweimal RETURN zu drücken, können Sie auch CTRL+j eingeben. Gleichzeitig wird dann nämlich auch der vorangehende Absatz auf fehlende Klammern überprüft.

Aufruf von LATEX

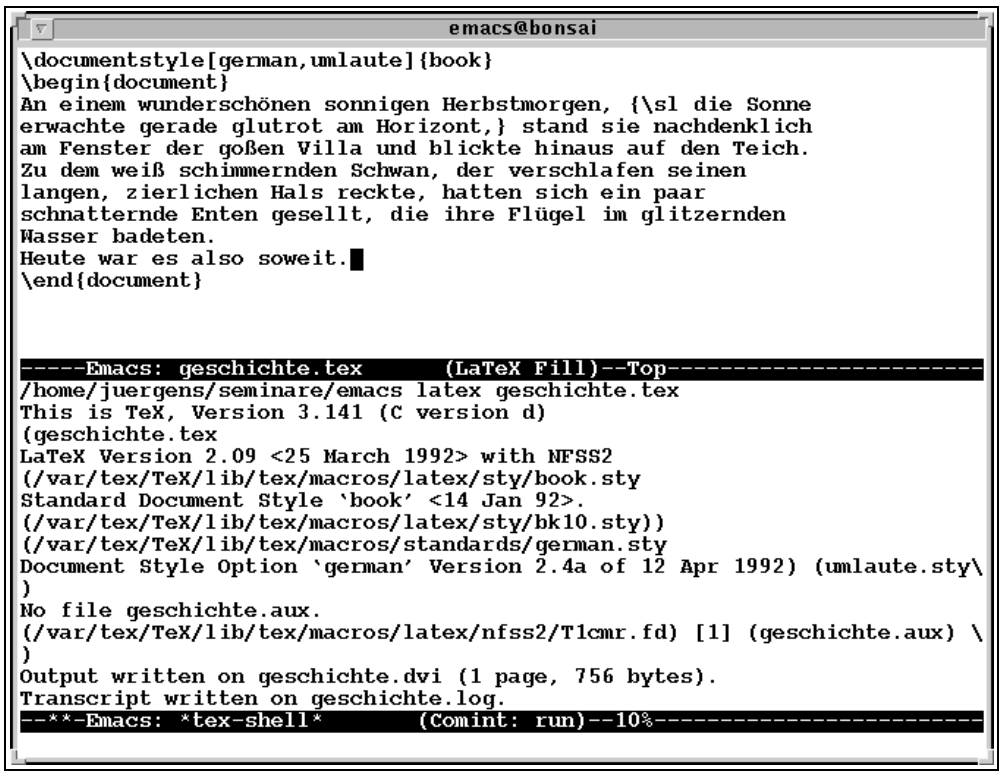
Ohne EMACS verlassen zu müssen, kann mit der Taste CTRL+c CTRL+b (tex-buffer)[33] oder CTRL+c CTRL+f (tex-file)[34] der TEX/LATEX-Lauf gestartet werden, um ein `dvi`-File zu erzeugen. Es wird ein zweites Fenster, die `TeX shell` eröffnet, in der der Ablauf beobachtet werden kann. Bei Fehlermeldungen, die ja normalerweise eine Zeilennummer beinhalten, können Sie in Ihrer Eingabe-Datei im ersten Fenster direkt die Korrekturen durchführen und anschließend den TEX/LATEX-Lauf erneut starten.

[31] EMACS entscheidet selbst, welche Anführungszeichen benötigt werden.

[32] Abhängig von der verwendeten Terminalemulation, wird statt des Gänsefüßchen eine Meldung `Mark set` erzeugt. Das Gänsefüßchen können Sie dann erzeugen durch CTRL+u 1 “.

[33] Erzeugt keine `aux, toc`, etc.-Dateien

[34] Die Datei muß hierfür zuvor abgespeichert werden.



Werden interaktive Eingaben benötigt, so kann das TEX-Shell-Fenster, wie bereits bekannt, mit CTRL+x o aktiviert werden.

Mit CTRL+c CTRL+k (Tex-kill-job) kann der TEX-Lauf abgebrochen werden.

Ansehen der LATEX-Ausgabe auf dem Bildschirm

Mit CTRL+c CTRL+v (tex-view) kann, sofern Sie in einer X Windows-Umgebung arbeiten, eine Bildschirmdarstellung der LATEX-Ausgabe vorgenommen werden. Dafür muß jedoch vorab die LISP-Variable tex-dvi-view-command mit dem Aufruf des entsprechenden Unix-Kommandos belegt werden.

Kleiner Zwischenbeitrag zum Arbeiten mit LISP-Variablen:

Mit CTRL+h v variable RETURN können Sie sich den Wert einer Variablen anzeigen lassen.

Mit ESC+x set-variable RETURN variablenname RETURN wert RETURN

ordnen Sie einer Variablen den gewünschten Wert zu. Im EMACS-Profil geben Sie stattdessen an:

```
(setq variablenname wert)
```

Ein Beispiel: Um die Bildschirmausgabe mit xdvi zu erreichen setzen Sie in Ihre .emacs-Datei den Befehl

```
(setq tex-dvi-view-command "xdvi *")
```

Drucken von LaTeX-Texten

Zum Ausdrucken Ihres Textes drücken Sie die Taste CTRL+c CTRL+p (tex-print).

Auch hier müssen Sie zuvor den richtigen Druckertreiber einstellen durch Setzen der Variable:

```
(setq tex-dvi-print-command "dvips *")
```

Ein kleiner Trick

Wollen Sie LaTeX nur für einen Teil Ihres Dokumentes laufen lassen, so gehen Sie folgendermaßen vor: Die sogenannte Präambel und alle eventuell zu Beginn des Dokumentes definierten Makros umgeben Sie mit den Kommandos

```
%**start of header
%**end of header
```

Markieren Sie anschließend den Text, für den der Lauf durchgeführt werden soll und drücken Sie danach CTRL+c CTRL+r (tex-region). Nur der markierte Text wird zusammen mit dem Header berücksichtigt. Praktisch, oder?

Umgebungen für LaTeX-Dokumente

Alle Umgebungen in LaTeX sehen im Prinzip folgendermaßen aus:

```
\begin{umgebung}

\end{umgebung}
```

Um auch hier das Schließen einer Umgebung nicht zu vergessen, geben Sie ein CTRL+c CTRL+o (tex-latex-block). Sie werden anschließend nach dem Namen der Umgebung (also z.B. itemize, tabular, figure usw) gefragt und die komplette Struktur wird in Ihr Skript geschrieben; der Cursor befindet sich dazwischen und Sie können sofort losschreiben.

Haben Sie eine Umgebung bereits selbst eröffnet, z.B. durch

```
\begin{lustig}
```

So können Sie sie jederzeit automatisch beenden durch CTRL+c CTRL+e (tex-close-latex-block). Emacs fügt automatisch das fehlende

```
\end{lustig}
```

ein.

Zusammenfassung der bisherigen Kommandos

Tasten	Kommando	Wirkung
CTRL+j	tex-terminate-paragraph	Absatzende einfügen
CTRL+c {	tex-insert-braces	Einfügen paariger Klammern
	validate-tex-buffer	Überprüfen paariger Klammern
CTRL+c CTRL+b	tex-buffer	Aufruf von TEX/LATEX
CTRL+c CTRL+r	tex-region	Aufruf von TEX/LATEX für einen markierten Bereich
CTRL+c CTRL+k	tex-kill-job	Abbrechen von TEX/LATEX
CTRL+c CTRL+v	tex-view	Ansehen der Ausgabe auf dem Bildschirm
CTRL+c CTRL+p	tex-print	Drucken der Ausgabe
CTRL+c CTRL+o	tex-latex-block	Einfügen einer LATEX-Umgebung
CTRL+c CTRL+e	tex-close-latex-block	Schließen einer LATEX-Umgebung

14 Emacs und X Windows

Haben Sie das Glück in einer X Windows-Umgebung arbeiten zu können, stehen Ihnen im EMACS weitere Möglichkeiten zur Verfügung. Unter X stellt sich EMACS in leicht veränderter Form dar.

Sie sehen u.a. zusätzlich eine Menuleiste oben, unter der sich einige EMACS-Befehle befinden, die Sie sonst durch Tastenkombinationen aufrufen mußten, sowie einen Scroll-Bar rechts, mit dessen Hilfe Sie sich in der Datei bewegen können. Die größte Erleichterung finden Sie aber sicherlich in der Verwendung der Maus.

Mausbenutzung

Folgende Hinweise sind zur Mausbenutzung zu beachten:

- ein Klick mit der linken Maustaste positioniert den Cursor an die entsprechende Stelle
- um einen Textbereich zu markieren, ziehen Sie die Maus bei gedrückter linker Maustaste über diesen Bereich. Er wird schattiert am Bildschirm markiert und kann genauso zum Kopieren, Löschen usw. benutzt werden, als wäre per Tastendruck markiert worden.
- drücken Sie die mittlere oder, falls das nicht funktioniert, die rechte und linke Maustaste gleichzeitig, so wird der Inhalt des Kill-Rings an der Cursorposition eingefügt und erspart Ihnen somit das Absetzen des yank-Befehl CTRL+y.
- die rechte Maustaste erstellt eine Kopie des Textes zwischen Cursor und Mauszeiger und schiebt sie in den Kill-Ring; wird die rechte Maustaste zweimal geklickt, so wird der Text zwischen Cursor und Mauszeiger in den Kill-Ring gelöscht.

Wichtige Editor-Funktionen können also relativ schnell über Maus-Klick durchgeführt werden.

Erstellen von X-Windows

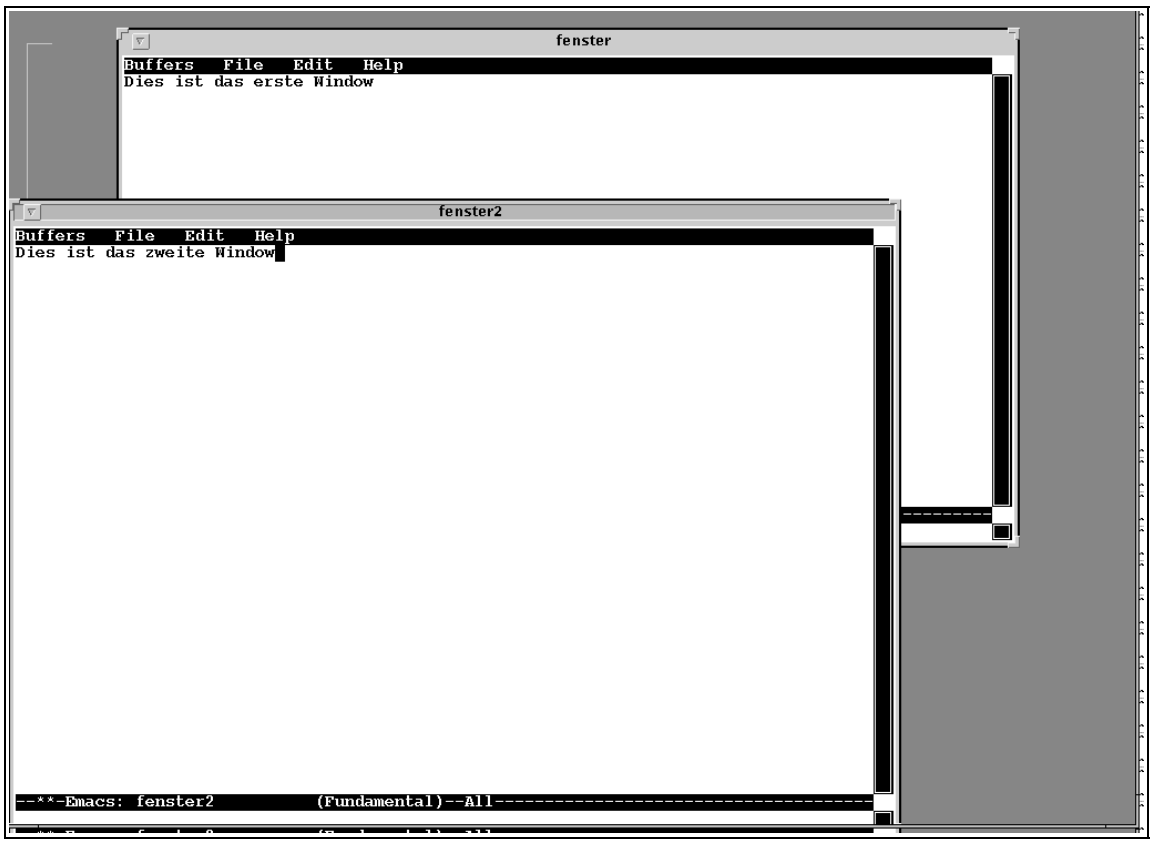
Ein weiterer großer Vorteil der X-Umgebung ist, daß Sie mehrere geöffnete Windows in mehreren „Rahmen“ erzeugen können. In Kapitel 8.2 auf Seite 19 haben Sie bereits gelernt, daß Sie ein zweites Window z.B. mit CTRL+x 2 oder CTRL+x 4 öffnen können. Das führt dazu, daß Ihr bisher einziges großes Fenster in zwei kleinere Fenster unterteilt wird. Unter X haben Sie andere Möglichkeiten: mit dem Befehl

CTRL+x 5 f

können Sie tatsächlich ein zweites neues Fenster erzeugen. Durch die Technik des Überlappens können Sie alle Windows in der gewünschten Größe halten.

Geben Sie ein:

CTRL+x 5 f fenster2



Durch Anklicken der obereren Bildleiste wird das entsprechende Fenster aktiviert und in den Vordergrund gebracht.

14.1 Benutzen des Scroll-Bars

Der Balken am rechten Rand des Windows, *Scroll-Bar* genannt, deutet an, an welche Stelle der Datei gerade im Bildschirmausschnitt zu sehen ist. Ist der Balken dunkel ausgefüllt, wie im obigen Beispiel, so ist Ihre Datei momentan noch nicht länger als eine Bildschirmseite. Ansonsten ist der Balken hell und am Anfang befindet sich ein dunkles Rechteck, der *Positionsanzeiger*, der mit Hilfe der Maus bewegt werden kann:

Klicken Sie die mittlere Maustaste (falls das nicht klappt: die beiden äußeren Maustasten gleichzeitig drücken) an einer gewünschten Stelle innerhalb des Balkens, springt der Positionsanzeiger direkt an diese Stelle.

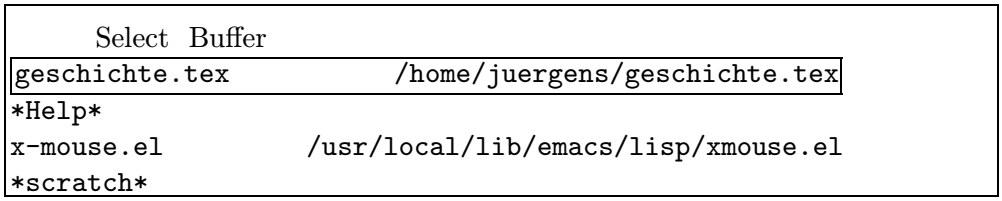
Klicken Sie die rechte Maustaste an einer beliebigen Stelle innerhalb des Scroll-Bars, so bewirkt das ein Zurückblättern, das Klicken der linken Maustaste ein Vorwärtsblättern in der Datei.

14.2 Die Menu-Leiste

Die Menu-Leiste ermöglicht Ihnen eine Ausführung von EMACS-Kommandos, indem Sie sie einfach mit der Maus anklicken.

Abhängig vom eingestellten Mode werden unterschiedliche Funktionen in der Menu-Leiste aufgeführt. Die Bedienung dieser Funktionen ist so einfach, daß sie hier nur anhand der ersten Funktion *Buffers* vorgestellt werden soll:

Klicken Sie mit der Maus auf *Buffers*, so erscheint in etwa folgendes Bild:



Hier kann der gewünschte Buffer einfach mit der Maus angeklickt werden und der Inhalt erscheint im EMACS-Fenster.

Diese Vorgehensweise ist in allen Menu-Funktionen gleich. Probieren Sie es einfach mal aus.

15 Und zum guten Schluß: ein bißchen Unterhaltung

Finden Sie diesen Lernstoff ein wenig trocken? Dann benötigen Sie sicherlich eine Pause.

EMACS bietet Ihnen kleine Möglichkeiten der Unterhaltung an. Lehnen Sie sich gemütlich zurück und versuchen Sie die folgenden Kommandos:

- ESC+x hanoi
  Wenn Sie selbst ausruhen wollen, können Sie hier das Türmchen-Bauen beobachten.
- ESC+x gomoku
  Spielen Sie Gomoku mit EMACS. (5 gewinnt)
- ESC+x blackbox
  4 Bälle müssen gefunden werden
- ESC+x mpuz
  Eine Multiplikationsaufgabe mit Buchstaben, deren Zahlenwert erraten werden muß
- ESC+x dunnet
  Ein typisches Abenteuerspiel
- ESC+x doctor
  Wenn's ohne Therapeut gar nicht mehr geht
- ESC+x yow
  Einfach ausprobieren

16 Das ist immer noch nicht alles ...

EMACS bietet noch mehr:

Zum Beispiel können Sie eine eigene Mail-Umgebung benutzen, mit deren Hilfe Sie relativ komfortabel Mails verschicken (CTRL+x m) und empfangen (ESC+x rmail) können, während Sie Ihre Dateien editieren.

Mit GNUS (ESC+x gnus) können Sie aus EMACS heraus mit dem NEWS-System arbeiten.

Es gibt einen Kalender (ESC+x calendar) in den Termine eingetragen werden können, der Feiertage, Sonnen- und Mondauf- und -untergänge und Mondphasen kennt und der eine Art Tagebuchführung unterstützt während Sie Ihre Dateien editieren.

Es gibt die Möglichkeit der Textformatierung. Sie können beispielsweise Zeileneinzüge, Textzentrierungen, Tabellenerstellung vornehmen und haben sogar die Möglichkeit einfache Strichzeichnungen zu erzeugen. Gleichzeitig kann Text zur besseren Übersicht ausgeblendet werden. Schließlich können Sie Ihre Eingabe auch noch in beliebiger Reihenfolge sortieren lassen.

EMACS unterstützt das Schreiben, Ausführen und Debuggen von Programmen in verschiedenen Sprachen: es existieren momentan Major-Modes für Lisp, Scheme, Awk, C, C++, Perl, Icon, Fortran und Muddle, sowie ein Major-Mode für die Erstellung von Makefiles.

Für diejenigen, die sich mit EMACS überhaupt nicht anfreunden können, wurde sogar eine VI-Emulation erstellt.

Es existieren zusätzlich noch eine ganze Reihe weiterer Lisp-Programme, die die Arbeit mit EMACS unterstützen. Dazu gehören u.a. `ispell`, mit dem der Text auf korrekte Rechtschreibung untersucht werden kann, `auctex`, mit dem LaTeX-Dokumente komfortabel erstellt werden können, `gopher`, mit dem das Informationssystem Gopher in einem eigenen Buffer interaktiv benutzt werden kann und verschiedenes mehr. Eine Aufstellung der im FernUniversitäts-Rechenzentrum verfügbaren zusätzlichen Lisp-Programme, sowie Hinweise zu deren Benutzung, wird in Kürze zur Verfügung stehen.

Die Beschreibung der oben angesprochenen Themen würde ganz sicher den Rahmen einer Einführung sprengen. Aber Sie haben nun hoffentlich einen guten Eindruck über die mächtigen Möglichkeiten des EMACS gewonnen und werden sich mit Spaß und Elan an das Kennenlernen dieses Editors begeben.