

Grundlegende Werkzeuge unter Linux

Auf diesem Blatt werden in Kürze die Werkzeuge vorgestellt, die für die Lösung der Aufgaben aus Übung und Praktikum unter Linux verwendet werden können, bzw. müssen. Natürlich steht es euch frei, die Aufgaben unter Windows zu lösen oder andere Linux-Programme, z.B. einen anderen Editor, zu verwenden. Es sei aber angemerkt, dass die hier vorgestellten Werkzeuge (unter Linux) grundlegend und weit verbreitet sind und es sich daher lohnt, sich mit ihnen zu beschäftigen.

1 Der Editor Emacs

Emacs ist eigentlich schon mehr als ein Editor, da er sehr viele Features hat. Man sagt auch, der eEmacs könne alles ausser Kaffee kochen. Für den Gebrauch in den Übungen und den Praktika muss man aber nur einen kleinen Bruchteil der Funktionen kennen.

1.1 Aufruf und Allgemeines

Programme und Texte kann man mit dem Editor Emacs schreiben. Hat man noch keine oder nur wenig Erfahrung mit dem Umgang mit irgendeinem Editor, sollte man sich zunächst die zugehörige Einführung (allerdings auf Englisch) ansehen. Diese findet ihr, wenn ihr nach dem Start auf *Help* und dann auf *Emacs Tutorial* klickt.

Der Emacs kann aus der Kommandozeile mit folgendem Aufruf gestartet werden:

```
emacs &
```

oder

```
emacs textfile &
```

wenn man die Datei *textfile* bearbeiten und - falls noch nicht vorhanden - erzeugen möchte. Ruft man den Emacs mit einer Datei mit der Endung *.c* (C-Datei) auf, startet er automatisch im C-Modus, in dem er z.B. Syntax-Highlighting unterstützt. Das gleiche gilt für Latex-Dateien (Endung *.tex*).

Für den Anfang ist es am einfachsten, die Menüleiste anstatt der Tastenkürzel zu verwenden. Im Menü findet man die Tastenkürzel auch in Klammern hinter den Menüpunkten angegeben, so dass man nach und nach die am häufigsten verwendeten Kommandos in der Kurzform lernen kann.

1.2 Tastenkürzel

Die meisten Kommandos können im Emacs nicht nur über die Menüleiste, sondern auch über die Tastaturkürzel eingegeben werden. Diese Bestehen in der Regel aus Kombinationen der Tasten *Ctrl*, bzw. *Strg* (je nach Tastatur), *Esc*, *Meta*, bzw. *Alt* (je nach Tastatur) und weiteren Tasten. Dabei wird die folgende Schreibweise verwendet:

- C-s* die *Ctrl*-(*Strg*-)Taste gedrückt halten und dazu die Taste *s* drücken
- C-s C-x* zuerst die *Ctrl*-Taste gedrückt halten und dazu die Taste *s* drücken
dann beide wieder loslassen
danach die *Ctrl*-Taste gedrückt halten und die Taste *x* drücken

Bei Zeichen, die nur über die *Shift*-Taste erreichbar sind, muss diese zusätzlich gedrückt werden. Entsprechendes gilt für die Taste *Meta*, bzw. *Alt*, z.B. die Tastenkombination *M-s*. Bei der *ESC*-Taste ist allerdings Vorsicht geboten, sie darf nicht gedrückt gehalten werden, sondern muss vor dem Pressen der zweiten Taste wieder losgelassen werden!

Wird zusätzlich zu einem Kommando noch eine Zusatzinformation wie z.B. der Dateiname benötigt, wird diese in den so genannten Minibuffer - die untersten Zeilen im Emacs-Fenster - eingegeben. Ähnlich wie in der Kommandozeile ist es oft möglich, die Tabulatortaste für automatische Erweiterungen zu verwenden, z.B. bei Dateinamen.

Die wichtigsten Befehle

- C-x C-f* Laden und ggf. Neuanlegen einer Datei
- C-x C-s* Speichern der aktuell bearbeiteten Datei
- C-x i* Einfügen einer Datei
- C-x C-w* Speichern einer Datei unter einem neuen Namen
- C-x C-c* Emacs beenden
- C-_* Undo: Rückgängigmachen der letzten Aktion(en), mehrfach verwendbar
- ESC d* Löschen bis zum Wortende
- C-k* Löschen bis zum Zeilenende
- C-SPACE* Setzen einer Marke (für Textmarkierungen zum Ausschneiden, Kopieren, etc.)
Bereich dann mit den Cursor-Tasten aufziehen
- C-w* markierten Bereich löschen
- M-w* markierten Bereich in den Zwischenspeicher zum Wiedereinfügen kopieren
- C-y* zuletzt gelöschten oder markierten Teil einfügen
- C-a* zum Zeilenanfang springen
- C-e* zum Zeilenende springen

Für Windows-Benutzer ist es wahrscheinlich ungewohnt, dass die Tasten *Pos1/Home*, bzw. *End* an Anfang und Ende des Textes und nicht der Zeile springen. Bei neueren Linux-Versionen verhalten sich diese Tasten allerdings wie unter Windows.

Befehle abbrechen

Bei allen Befehlen, die nicht sofort mit Eingabe der Tastenkombination abgeschlossen sind, kann das aktuelle Kommando folgendermaßen abgebrochen werden:

- C-g* Abbruch des aktuellen Kommandos

Damit geht der Emacs zurück in den normalen Textmodus.

Suchen und Ersetzen

- C-s Inkrementelles Suchen ab Cursorposition vorwärts
Wiederholtes Drücken sucht die weiteren Vorkommen
- C-r Inkrementelles Suchen ab Cursorpositions rückwärts
- M-% Suchen und Ersetzen, Erläuterung steht im Minibuffer

Dateiverwaltung

Mit dem Emacs können mehrere Dateien gleichzeitig bearbeitet und dargestellt werden. Mit jedem Öffnen einer neuen Datei wird ein zugehöriger neuer Buffer geöffnet, die zuvor bearbeiteten Dateien bleiben geöffnet und unverändert.

- C-x b *buffername* zum Buffer mit dem Namen *buffername* wechseln
- C-x C-b Bufferliste anzeigen, diese steht auch unter dem *Buffers* Menü zur Verfügung
- C-x k Buffer entfernen, d.h. die Bearbeitung der aktuellen Datei beenden
- C-x 1 nur eine Datei im aktuellen Fenster bearbeiten
- C-x 2 Fenster vertikal aufteilen, um zwei Dateien zu bearbeiten
- C-x 3 Fenster horizontal aufteilen
- C-x o aktuellen Fensterteil wechseln

1.3 Anpassen von Emacs an deutsches Latex

Arbeitet der Emacs im Latex-Mode, d.h. er wurde mit einer *.tex*-Datei aufgerufen, sollte man für die Benutzung mit deutschem Text die Benutzerkonfiguration ändern und zwar aus folgendem Grund. Wenn man die Taste " drückt, werden automatisch die richtigen öffnenden oder schließenden Anführungszeichen verwendet, " und", allerdings in der amerikanischen Form, d.h. beide Zeichen oben.

Um die deutsche Form der Anführungszeichen einzustellen, geht man wie folgendermaßen vor.

- klickt auf *Help, Customize, Browse Customization Groups*
- es erscheint ein Menü mit themenbezogenen Untermenüs
- klickt auf das + vor der Zeile *Wp*
- es erscheint ein Untermenü
- klickt auf das + vor der Zeile *Tex*
- es erscheint ein Latex-bezogenes Untermenü
- klickt auf das *Option* vor der Zeile mit *Tex Open Quote*
- nun kann man den Text für die öffnenden Anführungszeichen in den grauen Kasten eintragen, nachdem man auf ihn geklickt hat
- klickt auf *State* in derselben Zeile
- es erscheint ein Menü, klickt auf *Save for Future Sessions*
- macht dasselbe für *Tex Close Quote* für die schließenden Anführungszeichen

Anmerkung: ' ist die Backtick-Taste, auch als Grave-Akzent oder Abwärts-Akzent bezeichnet.
' ist die Apostroph-Taste und nicht die Aufwärts-Akzent Taste.

2 Der Compiler gcc

Nachdem man ein C-Programm z.B. mit dem Emacs geschrieben hat, möchte man es auch kompilieren und ausführen. Der Standard-C-Compiler unter Linux ist der GNU-Compiler gcc.

2.1 Übersetzen eines Programmes

Der Compiler gcc wird in der Kommandozeile ausgeführt. Zum Übersetzen wird folgendes Kommando verwendet:

```
gcc -o programm programm.c
```

mit *programm.c* als der zu übersetzenden C-Datei. Das ausführbare Programm heisst dann *programm*

Dies ist die einfachste Form des Übersetzens, die meistens auch ausreichend ist. Der Compiler gcc stellt zahlreiche Optionen zur Verfügung, mit denen man den Kompiliervorgang genauer steuern kann. Mehr dazu unter 2.3.

2.2 Ausführen eines Programms

Das kompilierte Programm kann auf einfache Art und Weise gestartet werden. Gebt einfach

```
programm
```

oder wenn das nicht funktioniert

```
./programm
```

ein.

2.3 Wichtige Optionen des gcc

Die vielfältigen Möglichkeiten des gcc sieht man sich am Besten im dazugehörigen Manual an. Dieses findet man, wenn man in der Kommandozeile

```
man gcc
```

eingibt.

Eine wichtige Option sollte allerdings an dieser Stelle erwähnt werden, nämlich *-Wall*. Die Angabe dieser Option beim Kompilieren bedeutet, dass nahezu alle Warnungen eingeschaltet werden, die erzeugt werden können. Diese beziehen sich auf C-Programme, die zwar korrekt übersetzt werden können, bei denen aber möglicherweise Fehler in der Verwendung von C aufgetreten sind. Diese Stellen kann man dann auf Korrektheit überprüfen. Besonders wertvoll ist diese Option in Zusammenhang mit unterschiedlichen Datentypen und Zeigern.

```
gcc -Wall -o programm programm.c
```

3 Das Textsatzsystem Latex

Da die Lösungen der Übungen in Latex abgegeben werden sollen, sei dieses System hier ebenfalls kurz erläutert. Zum Verfassen wissenschaftler Texte ist Latex sowieso DAS Mittel der Wahl, weshalb es sich zusätzlich lohnt, sich damit zu beschäftigen.

3.1 Allgemeines

Latex ermöglicht es, mit einfachen Mitteln sehr professionell aussehende Texte zu gestalten. Eine gute Einführung findet man in der *Latex-Kurzbeschreibung*, die man auf der Webseite dieser Veranstaltung finden kann (www.tzi.de/agbs/lehre/ws0203/gdi1/).

Der Aufruf von Latex erfolgt in der Kommandozeile mit folgendem Befehl:

```
latex latexdatei.tex
```

wobei *latexdatei.tex* die zu verarbeitende Latex-Datei ist. Sollten bei der Verarbeitung Fehler auftreten, so lässt sich die Übersetzung häufig mit der *Return*-Taste fortsetzen, das Ergebnis kann dann aber fehlerhaft sein. Mit *x-Return* wird die Verarbeitung abgebrochen.

Das Ergebnis der Verarbeitung ist eine *.dvi*-Datei (*Device Independent File*), die man sich mit unterschiedlichen Tools ansehen kann, z.B.

```
xdvi latexdatei.dvi
```

Zum Drucken sollte man die *.dvi*-Datei in das PostScript-Format umwandeln. Dies geschieht mit

```
dvips -o latexdatei.ps latexdatei.dvi
```

Das Ergebnis ist hierbei die PostScript-Datei *latexdatei.ps*, die man sich mit dem Programm Ghostview ansehen kann.

```
gv latexdatei.ps
```

Das Drucken funktioniert leider nicht direkt über das Programm Ghostview, da der Linux-Server im Rechnerraum nicht an einen Drucker angeschlossen ist. Zum Drucken muss daher die PostScript-Datei vom Linux-Rechner auf den Windows-Rechner übertragen werden. Dies geschieht mit Hilfe des Internet-Explorers, der auf dem Windows-Rechner installiert ist.

- klickt *Start, Programme, Internet Explorer*
- gibt als Internet-Adresse *ftp://benutzername@lablin/* ein, dabei ist euer Benutzername als *benutzername einzusetzen*
- es kommt eine Meldung, dass das Passwort fehlt und eingegeben werden muss
- gebt euer Linux-Passwort ein
- es erscheint ein Fenster mit euren Dateien und Verzeichnissen

- markiert die zu übertragende(n) Datei(en) mit einem einfachen Klick und klickt dann auf *Datei, Kopieren nach Ordner, Eigene Dateien, OK*
- klickt dann im Internet Explorer auf *Ordner, Eigene Dateien*
- wählt die übertragene PostScript-Datei mit einem Doppelklick
- es startet das Anzeige-Programm GSview
- gedruckt werden kann über das Drucker-Icon oben links

Wenn ihr die Dokumente platzsparend ausdrucken wollt, könnt ihr das Linux-Programm *psnup* verwenden:

```
psnup -2 eingabe.ps ausgabe.ps
```

Dadurch wird aus einer Eingabedatei *eingabe.ps* eine Ausgabedatei *ausgabe.ps* erzeugt, bei der zwei Seiten gedreht nebeneinander auf eine Seite kopiert sind.

DIE LÖSUNGEN ZU DEN ÜBUNGSAUFGABEN SOLLTEN ABER NICHT AUF DIESE WEISE BEHANDELT WERDEN, UM GENUG PLATZ FÜR KORREKTUREN ZU LASSEN!

3.2 Latex im Emacs

Wenn die Latex-Datei im Emacs bearbeitet wurde, kann man das Übersetzen der Datei auch im Emacs anstoßen, da dieser bei der Bearbeitung von *.tex*-Dateien automatisch in den Latex-Mode wechselt. Über das dann vorhandene Latex-Menü lassen sich viele hilfreiche Aktionen anstoßen. Am Besten probiert man diese mit einer Beispieldatei selbst aus. Das Übersetzen von Latex nach *dvi* geschieht mit folgendem Kommando:

```
C-c C-c
```

Die übersetzte Datei kann mit demselben Kommando auch angesehen werden.

3.3 Die Latex Mustervorlage für die Übungen

Auf der Homepage zu dieser Vorlesung findet ihr eine Datei namens *gdi1_muster.tex*, welche als Rahmen für die Lösung der Aufgaben verwendet werden soll. Zusätzlich wird die Datei *defs.tex* benötigt, die ebenfalls auf der Homepage bereitliegt. Beide Dateien müssen zusammen in ein Verzeichnis kopiert werden, damit sich die Musterdatei übersetzen lässt.

Für die Lösungen solltet ihr eine Kopie des Musters anlegen und anschließend modifizieren. Ausserdem kann man mit einer solchen Kopie prima die Möglichkeiten von Latex ausprobieren.