

Erweiterungen für das Editierprogramm NEdit*

H. Hollatz

16. Januar 2006

Zusammenfassung

Es werden Arbeitsmenüs vorgestellt, die man in das Editierprogramm NEdit einzubeziehen hat. Zunächst geht es um die Nutzung von NEdit als \LaTeX -Editor. Insbesondere wird dabei ein hauptdokumenten-freies Erarbeiten größerer \LaTeX -Dokumente dargestellt. Das 2. Menü betrifft die C/C++-Programmierung; das dritte dient der Arbeit mit normalen Textdateien. Den Abschluß bildet das Konzept einer losen Projekt-Bearbeitung mit einem entsprechenden Arbeitsmenü.

1. Einführung

Die hier vorgestellten Arbeitsmenüs entstanden einerseits aus dem Bedürfnis, für verschiedene Aufgaben den gleichen Editor zu nutzen und andererseits aus dem Bedürfnis, sich bei der Arbeit mit dem Rechner nicht ständig damit beschäftigen zu müssen, wie man eine Aufgabe löst, sondern seine Anstrengung darauf richtet, was man tun möchte. Nur wenige Editoren bieten eine geeignete Schnittstelle an. So besitzt der Editor NEdit auf UNIX eine eigene Sprache zum Beschreiben und Aufruf neuer Funktionen. Unter WINDOWS hat der Editor TextPad eine Schnittstelle, mit deren Hilfe Parameter, deren Werte sich während der Arbeit mit dem Editor ändern, an andere Anwendungsprogramme, insbesondere an bat-Dateien übergebbar sind. Die hier vorgelegte Erweiterung darf als Muster für weitere Arbeitsmenüs gesehen werden.

Es sind 6 Arbeitsmenüs realisiert:

LaTeX, Language, File, Project, Netservice, Program,

von denen 4 hier beschrieben werden. Mit den ersten drei Menüs ist eine Menü-Dateitypiste verbunden:

```
lst_LaTeX='.tex .sty .cls'  
lst_Language='.c .cpp .h .f'
```

Eine Ordnerliste ist mit dem System assoziiert (System-Ordnerliste); jedem Ordner ist eine Ordner-Dateitypiste zugeordnet (`dirs.lst`). Die Menü-Dateitypisten definieren gemeinsam mit der System-Ordnerliste je eine Menü-Ordnerliste für das Menü LaTeX und für das Menü

*Programm-Download: <http://www.hhollatz.de>

Language. Ein Ordner aus der System-Ordnerliste wird genau dann in die Menü-Ordnerliste aufgenommen, wenn die Ordner-Dateitypliste mit der Menü-Dateitypliste mindestens eine Maske gemeinsam hat. Die Menü-Ordnerliste für das Menü `File` ist die System-Ordnerliste. Bei der Ordnerauswahl innerhalb eines Menüs werden nur die Ordner aus der Menü-Ordnerliste zur Auswahl gestellt. Bei der Dateiauswahl aus einem Ordner werden jene Dateien zur Auswahl gestellt, die zwei Filter durchlaufen haben: Zunächst wird mit der Menü-Dateitypliste gefiltert; danach mit der Ordner-Dateitypliste.

Die Arbeitsmenüs sind über Funktionen in der Datei `x` realisiert und über eine Import-Datei (`nedit.rc`) in `NEdit` eingebunden.

2. Menü LaTeX

An `LaTeX`-Texten werden normale Briefe, normale Texte, Artikel, Berichte, Folien sowie Bücher unterstützt. Die einzelnen Dokumentvorlagen stehen als Makros im `NEdit`-Menü bereit. Jeder Text beginnt mit einem Eröffnungsmakro und endet mit dem Makro `\Ende`:

normaler Brief `\Brief ...`

normaler Text: `\Text ...`

Artikel: `\Artikel ...`

Bericht: `\Bericht ...`

Buch: `\Buch ...`

Folien: `\Folien ...`

Die Dokumentklassen `Artikel`, `Bericht` und `Buch` gestatten die unabhängige Übersetzung einzelner Teile des gesamten Dokumentes, ohne die Teile für das Übersetzen des Gesamtdokumentes ändern zu müssen. Dadurch entfällt die Definition eines Hauptdokumentes.

Beispiel: Es seien bereits 2 Kapitel `k1.tex`, `k2.tex` eines Buches geschrieben; jedes der beiden Dokumente hat die Struktur `\Buch ... \Ende`. Durch die Datei

```
\Buch
\input k1
\input k2
\Ende
```

wird das aktuelle Gesamtdokument beschrieben.

Hinter dieser Technik steht die Datei `_latex`, die entweder jedem `LaTeX`-Dokument in der Form `\input _latex` voranzustellen oder in die Datei `latex.ltx` des `TeX`-Systems zu kopieren ist, um dadurch beim Start von `LaTeX` in der Datei `latex.fmt` bereits vorhanden zu sein. Zu jeder Dokument-Umgebung gehört eine `sty`-Datei; so z. B. zur Umgebung `\Text` die Datei `Text.sty`; sie enthält die Informationen für den Dokumententyp und sollte in den Arbeitsordner kopiert werden, um sie dort den spezifischen Bedürfnissen anzupassen. Es sei erwähnt, dass für die Dokumente `Artikel`, `Bericht` und `Buch` eine 14pt-Schrift verfügbar ist.

Das `LaTeX`-Menü enthält folgende Menüpunkte:

New LaTeX file

Nachdem der richtige Ordner aus der Menü-Ordnerliste ausgewählt und ein Datei-Name eingegeben worden ist, wird man aufgefordert, aus den obigen \LaTeX -Dokument-Vorlagen auszuwählen. Diese Aktion ist auch in `NEdit` eingebunden (`Macro->LaTeX-Dok`).

Open LaTeX file

Aus einem Ordner der Menü-Ordnerliste werden die Dateien entsprechend der Menü-Dateityp- und der Ordner-Dateityp-Liste zur Auswahl gestellt, um die ausgewählten Dateien gleichzeitig zu öffnen.

Compose

Es wird der \LaTeX -Compiler mit der aktuellen Datei aufgerufen. Falls ein anderer Compiler (z. B. der \TeX -Compiler) bzw. andere Kommandos aufgerufen werden sollen, ist dies im Dokument darzustellen:

```
\iffalse%%  
tex dokument.tex  
\fi%%
```

Hierin sind die beiden Prozent-Zeichen wesentlich, um von anderen Umgebungen gleicher Art zu unterscheiden.

pdf-LaTeX

Es werden der `pdfLaTeX`-Compiler mit der aktuellen Datei aufgerufen und eine pdf-Datei erzeugt.

View

Die zuletzt erzeugte `dvi`-, `pdf`- oder `ps`-Datei wird angezeigt.

Error filter

Aus der `log`-Datei wird eine neue Datei erzeugt und geöffnet, die die Fehlermeldungen enthält (einschließlich `overfull` und `underfull`).

View error

Der den Fehler enthaltene Quelltext wird geöffnet und zur Fehlerstelle gesprungen.

View logfile

Anzeigen der `log`-Datei.

Index register

Erzeugen eines Indexregisters (bei einem Buch).

Postscript

Erzeugen einer `ps`-Datei aus einer `dvi`-Datei für das aktuelle Dokument.

Printing letters

Mit diesem Menüpunkt lassen sich Briefe (Serienbriefe) drucken. Das Menü arbeitet auf der Grundlage einer Adressen-Datei (Dateityp `adr`) und eines \LaTeX -Dokumentes vom Typ `Brief` (beides im aktuellen Ordner). Eine Adressen-Datei beinhaltet Blöcke entsprechend folgender Struktur:

```
<SEL>   Toni
NAME    Tschechow
VORNAME Anton
TITEL   Herrn Geheimrat
STR     Puschkinstr. 17
ORT     51379 Rachmanin
ANREDE  Hallo Toni,
</SEL>
```

In einem Block dürfen noch weitere Informationen stehen, wie z. B.:

```
TEL      039245 3805
EMAIL    Anton.Tschechow@web.de
WWW      http://www.tschechow.ru
SONSTIGES geb. 11.11.11
```

Dabei ist die Reihenfolge beliebig. In jedem Datensatz müssen alle Identifikatoren stehen (gegebenenfalls mit Leerwort als Wert). Man schreibe nun ein \LaTeX -Dokument vom Typ Brief, in dem u. a. steht

```
\input {brief.vle}
\Adresse{\TITEL\ \VORNAME\ \NAME\ \STR\ \ORT}
\Anrede{\ANREDE}
```

Beim Aufruf des Menüs wird man aufgefordert, über die SEL-Namen Personen auszuwählen, die den aktuellen Brief erhalten sollen. Aus dem einer Person zugeordneten Datenblock wird die Datei `brief.vle` erzeugt:

```
\def\NAME{Tschechow}
\def\VORNAME{Anton}
\def\TITEL{Herrn Geheimrat}
\def\STR{Puschkinstr. 17}
\def\ORT{51379 Rachmanin}
\def\ANREDE{Hallo Toni,}
```

und beim Übersetzen verwendet.

`ps to ps`

Aus der zum aktuellen Dokument gehörenden `ps`-Datei wird eine neue erzeugt, die für geänderten Ausdruck vorbereitet ist. Dabei gibt es folgende Möglichkeiten: Ein- oder zweiseitiger Druck, 2, 4 oder 8 Dokumentseiten auf einer Druckseite.

`Edit documentstyle`

Die zum aktuellen Dokument gehörende `sty`-Datei wird in den aktuellen Ordner kopiert und zum Editieren geöffnet.

Graphics to LaTeX

Es darf eine Graphik-Datei ausgewählt werden, um daraus einen \LaTeX -Font zu erzeugen. Die dafür benötigten Daten werden interaktiv erfragt. Als Graphiken sind `bmp`-, `tif`-, `gif`- und `pcx`-Dateien zugelassen.

Die hier genannten Menü-Punkte sind nur Beispiele. Ein Blick in die Datei `x` sollte beim Erstellen weiterer Menü-Punkte hilfreich sein.

3. Menü Language

Die modernen C/C++-Compiler sind mit leistungsfähigen Editoren ausgestattet, so dass es hier nicht darum gehen kann, diese zu ersetzen. Vielmehr geht es darum, zusätzliche Entwurf-funktionen zu kreieren. Für kleine, einfache Programme ist das hier realisierte Menü völlig ausreichend. Das Menü enthält die folgenden Menü-Punkte:

New source text

Nachdem der geeignete Ordner ausgewählt wurde, ist der vollständige Dateiname anzugeben; danach wird eine source-Datei kreiert, die bereits jene Anweisungen enthält, die für die weitere Arbeit zweckmäßig sind.

Open source text

Aus der Menü-Ordnerliste ist ein Ordner auszuwählen, aus dem dann mittels der Menü-Dateitypliste geeignete Dateien zur Auswahl gestellt werden.

Compile

Es wird der C/C++-Compiler aufgerufen; eine im Sourcetext befindliche Klammer der Form

```
/*:UNIX  
... Kommandos ...  
*/
```

hat beim Aufruf Vorrang.

Run

Das aus dem aktuellen Source-Code erzeugte Programm wird nach Abfrage der Aufruf-Parameter aufgerufen. Falls dieser Fall nicht eintritt, wird zur Auswahl eines ausführbaren Programms/Kommandos aufgefordert; die dabei angezeigte Auswahlliste wird allein auf Grundlage des Kriteriums ausführbar erzeugt.

4. Menü File

Das Menü enthält insbesondere die folgenden Menü-Punkte:

Open file

Aus der Menü-Ordnerliste ist ein Ordner auszuwählen; aus diesem werden mittels der Ordner-Dateitypliste Dateien ausgewählt, die dann gleichzeitig geöffnet werden.

Open high

Im aktuellen Text ist ein Dateiname markiert; die entsprechende Datei wird gesucht und geöffnet.

New file

In dem ausgewählten Ordner wird eine neue Datei angelegt, deren Name interaktiv erfragt wird.

Save as

In einem aus der Menü-Ordnerliste auszuwählenden Ordner wird eine Kopie der aktuellen Datei mit dem eingegebenen Namen abgelegt.

Copy file

Über diesen Menü-Punkt kann man Dateien in einen anderen Ordner kopieren oder bewegen.

Create copy

Im aktuellen Ordner wird eine Kopie der aktuellen Datei angelegt.

Include file

Aus einem ausgewählten Ordner ist eine Datei auszuwählen, die in die aktuelle Datei kopiert wird (ab der Zeile, in der der Cursor steht).

Script to LaTeX

Die aktuelle Datei wird in ein \LaTeX -Dokument transformiert, das sowohl als selbständiges Dokument behandelbar ist als auch als Input-Datei für ein anderes \LaTeX -Dokument dienen kann.

Script to postscript

Aus der aktuellen Datei wird eine ps-Datei erzeugt.

Sort

Die Zeilen der aktuellen Datei werden aufsteigend sortiert. Im Falle einer adr-Datei werden die Datenblöcke sortiert.

Column selection

Aus der aktuellen adr-Datei werden interaktiv Datensätze ausgewählt, um daraus eine neue Datei zu bilden.

Clear folder

Löschen von Dateien in einem Ordner aus der Menü-Ordnerliste. Es werden alle nicht über die Ordner-Dateitypliste auswählbaren Datei-Namen angezeigt.

Connect folder

Ordner zur System-Ordnerliste hinzufügen. Gleichzeitig werden auch die Dateityplisten erstellt.

Disconnect folder

Ordner aus der System-Ordnerliste entfernen.

5. Menü Projekt

Ein Projekt ist eine Sammlung von Dateien, die gemeinsam eine wohlbestimmte Einheit bilden. So kann ein C++-Programm mit seinen zahlreichen Input-Dateien ein Projekt bilden; auch ein größeres L^AT_EX-Dokument, wie etwa ein Buch, wird meist aus mehreren Dateien bestehen, die jede für sich eine gewisse Selbständigkeit haben, aber innerhalb des Gesamtdokumentes unerlässlich sind. Jedes Projekt hat einen Namen (Projektname); die Namen der zum Projekt gehörenden Dateien sind in einer Projekt-Datei abgelegt. Alle Projektdateien befinden sich in einem gemeinsamen Ordner, dem Projekt-Ordner. Die Projekt-Datei hat den Namen `projektname.proj` und besitzt den folgenden Aufbau:

```
<DIR> Pfad (relativ zur Wurzel)
Dateiname Dateiname ...
</DIR>
...
<BIN> Pfad (relativ zur Wurzel)
Dateiname Dateiname ...
</BIN>
...
```

In einem DIR-Komplex stehen die Namen aller jener editierbaren Dateien, die beim Projekt-Erarbeiten einem ständigen Wandel unterworfen sind; sie müssen im angegebenen Ordner stehen; der Wurzelordner wird mit `.` angegeben; alle weiteren relativ zu diesem, aber ohne den Punkt. Die BIN-Komplexe sind analog zum DIR-Komplex aufgebaut; sie enthalten neben nicht-editierbare Dateien auch solche, die nicht editiert werden sollen. Das Menü enthält die folgenden Menü-Punkte:

Open project

Es wird ein Projekt ausgewählt und die betreffende Projektdatei geöffnet.

Open high

In der aktuellen Projektdatei ist der Name eines Projektmitgliedes markiert; die betreffende Datei wird geöffnet.

New project

Ein neues Projekt wird kreiert und geöffnet.

Open member

Die ausgewählten Projektmitglieder des aktuellen Projektes werden geöffnet.

Extend

Das aktuelle Projekt wird um Dateien erweitert. Bei der Dateiauswahl werden gegebenenfalls nur jene Datei-Namen angezeigt, die sich noch nicht im Projekt befinden.

Gather

Aus der Namensliste aller editierbaren Projektmitglieder des aktuellen Projektes werden Dateien ausgewählt, diese zu einer einzigen zusammengelegt, um sie danach zu öffnen; eine solche Datei erhält den Name `projektname_DATUM.all`, wobei das Datum die

Form `TagMonatJahr` hat. Damit ist es möglich, mehrere Projektmitglieder einheitlich zu editieren, was das Ändern mehrerer Dateien eines Projektes wesentlich sicherer und effizienter gestaltet.

`Expand`

Ein zusammengelegtes Projekt wird expandiert.

`Pack`

Ausgewählte Projektmitglieder des aktuellen Projektes werden zu einer `gz`-Datei gepackt, wobei wie oben das Datum in den Datei-Namen eingearbeitet wird.

`Pack home`

Aus ausgewählten, assoziierten Ordnern werden ausgewählte Dateien zu einer `gz`-Datei gepackt, wobei wie oben das Datum in den Datei-Namen eingearbeitet wird.

`Unpack`

Entpacken von Projektmitgliedern aus einer korrespondierenden `gz`-Datei .

`Unpack home`

Entpacken aus einer über den Menüpunkt `Pack home` gepackten `gz`-Datei .

6. Installation

Die Distribution enthält bereits den Editor `NEdit`. Um die hier vorgestellten Menüs innerhalb von `NEdit` nutzen zu können, ist es nötig, einige Umgebungsvariable als Pfade in der `login`-Datei (z. B. `.dtprofile`) zu setzen:

1. `X_LIB`: System-Ordner (auch `inPATH` aufnehmen)(`$HOME/X_NEdit`).
2. `X_TERM`: Zu verwendendes Terminal (`dtterm`).
3. `EDITOR`: Standard-Editor
(`$X_LIB/nedit -server -import $X_LIB/nedit.rc`).
4. `LANG`: `de_DE.iso88591`.
5. `PROJECT`: Projekte-Ordner (`$HOME/Projekte`).
6. `browser`: Internet-Browser (`/opt/Netscape/netscape`).
7. `gv`: Pfad für `ghostview` (`/opt/gv/bin/gv`).
8. `lst_Language`: Datei-Typen für Programmiersprachen (`' .c .cpp .h .f '`).
9. `lst_LaTeX`: Datei-Typen für `LATEX` (`' .tex .sty .cls '`).
10. `lpr`: Drucker-Aufruf (z. B. `'lp -dletter'`)
11. `TEX_PRIVATE`: Wurzelordner für `LATEX`-Inputs; nach der Wurzel werden die Ordner `inputs`, `mf`, `pk`, `tfm` und `tif` erwartet.

Weiterhin werden u. U. folgende Programme aufgerufen, die nicht zur Distribution gehören, aber über die Umgebungsvariable PATH erreichbar sein müssen:

```
abiword, tar, gzip, gunzip, unpack, uncompress, xmaple,  
dtpad, xfig, ssh, scp, ftp, telnet, elm, latex, pdflatex,  
xdvi, acroread, dticon, imageview, dvips, ps2pdf, bibtex,  
dvi2pdf, CC, f90, dde, fgrep
```

Nach dem Auspacken ist alles in den Systemordner zu verschieben. In den C-Source-Texten findet man jeweils eine Kurzdokumentation.