

gedit

Der GNOME - Texteditor

Gliederung

2

- Das GNOME-Projekt
- GTK+ / GObject
- gedit
- Problemstellung
- Lösung

GNOME - Ziele

3

- Free Software
- Usability
- Accessibility
- Internationalization + Localization
- Developer-friendliness
- Organization
- Support

GNOME - Organisation

4

- besteht aus vielen verschiedenen Projekten
 - GTK+, GVFS, GConf, Nautilus, gedit, ...
- ein Bugzilla-Bugtracker für alle Projekte
- SVN - Repository
- Mailinglisten
- Major-Releases im Abstand von 6 Monaten
 - koordiniert durch Release Team

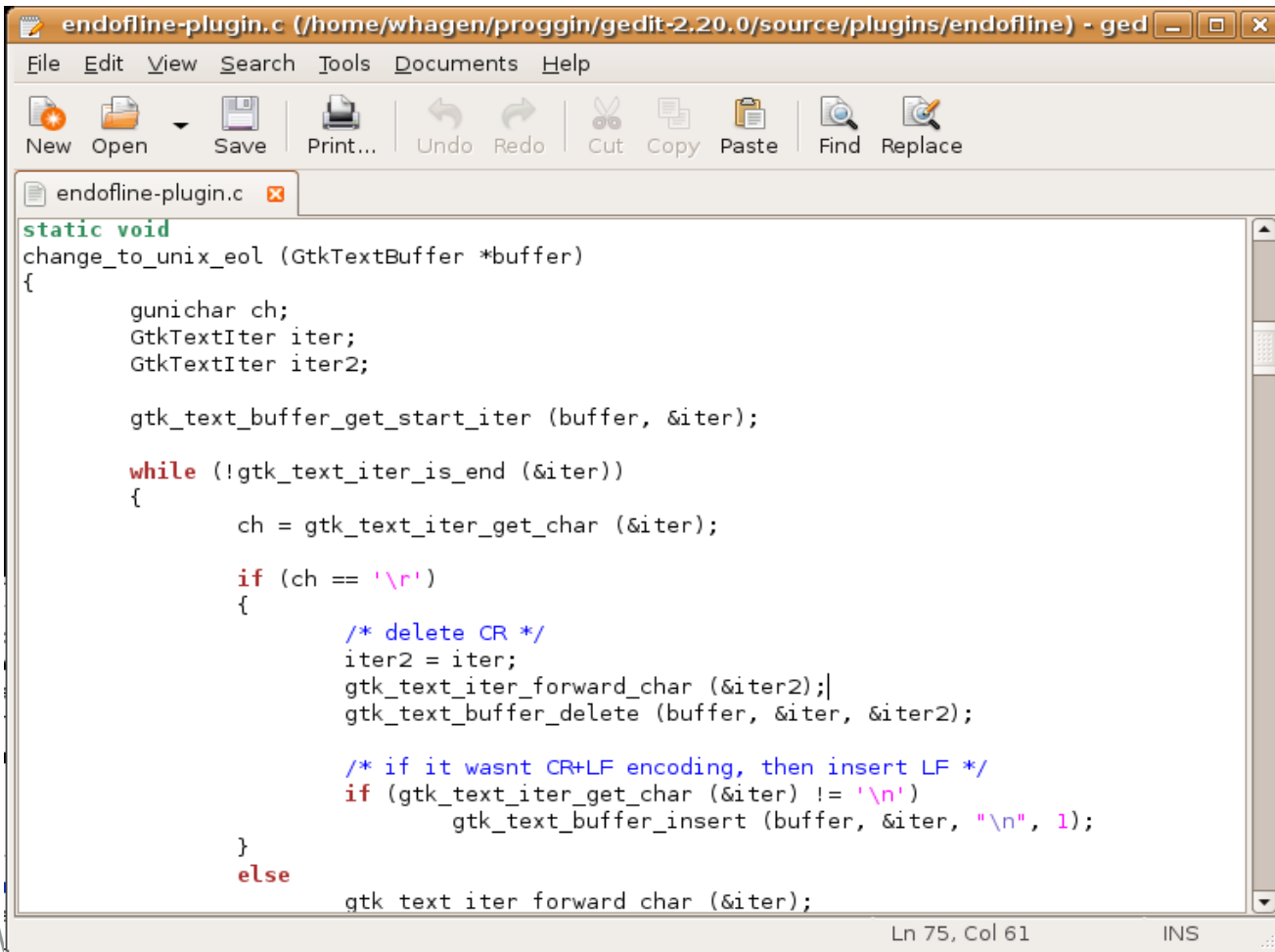
- GUI-Toolkit
 - bietet GUI-Elemente an: Menüs, Buttons, Fenster, ...
- 1997 ins Leben gerufen, ursprünglich für GIMP
- in C geschrieben
- nicht-GUI-Funktionalitäten in GLib ausgelagert
 - Datenstrukturen, Timer, Threads, Stringfunktionen, ...
- objektorientiert, basiert auf GObject

GObject

6

- ermöglicht OO-Programmierung in C
 - Interfaces, Vererbung, Nachrichten
- Basis für alles in GNOME
 - „C with GObject“
- Bindings für andere Sprachen
- Signal-System
- Nachteil: mühselig
 - Vala, GObject Builder

- Standard-Texteditor für GNOME



```

endofline-plugin.c (/home/whagen/proggin/gedit-2.20.0/source/plugins/endofline) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
endofline-plugin.c
static void
change_to_unix_eol (GtkTextBuffer *buffer)
{
    gunichar ch;
    GtkTextIter iter;
    GtkTextIter iter2;

    gtk_text_buffer_get_start_iter (buffer, &iter);

    while (!gtk_text_iter_is_end (&iter))
    {
        ch = gtk_text_iter_get_char (&iter);

        if (ch == '\r')
        {
            /* delete CR */
            iter2 = iter;
            gtk_text_iter_forward_char (&iter2);
            gtk_text_buffer_delete (buffer, &iter, &iter2);

            /* if it wasnt CR+LF encoding, then insert LF */
            if (gtk_text_iter_get_char (&iter) != '\n')
                gtk_text_buffer_insert (buffer, &iter, "\n", 1);
        }
        else
            gtk_text_iter_forward_char (&iter);
    }
}
Ln 75, Col 61 INS

```

gedit

8

- in C geschrieben
- 50k LOC
- Subklassen von GTK+ - Komponenten
- flexibles Plugin-System (C oder Python)
- große Anzahl an Plugins vorhanden
 - Python-Konsole
 - eingebaute Shell
 - Shell-Scripts auf Tastendruck
 - ...

Das Problem

9

- verschiedene Formate für Zeilenumbruch
 - OS2, DOS, Windows: CR+LF (0x0D + 0x0A)
 - Unix: LF (0x0A)
 - frühere MacOS: CR (0x0D)
- Benutzer soll über Format entscheiden können
- Maintainer: Nicht im Hauptprogramm, sondern als Plugin
 - nur für Bruchteil der Benutzer relevant
 - fügt UI-Komplexität hinzu -> gegen GNOME-Konzept

Lösungsidee

10

- Auswahl des gewünschten Formates im Edit-Menü
- automatisches Search & Replace beim Abspeichern
- Soll für alle Kombinationen der Formate funktionieren

Implementierung

11

- Registrierung für „saving“-Signal bei allen Dokumenten
- Sicherstellung, dass neu geöffnete Dokumente erfasst werden
- Formatwahl wird für jedes GeditWindow gespeichert
- Hinzufügen eines Untermenüs mit XML-String
- auf saubere Deinitialisierung achten

Quellen

12

- API-Referenzen
 - <http://library.gnome.org/devel/references>
- Quelltext von gedit sowie Plugins
- <http://www.gnome.org/about/>
- <http://en.wikipedia.org/wiki/GNOME> (28.01.2009)