SelfLinux-0.13.0

Editoren

Autor: Johnny Graber (*linux@jgraber.ch*)
Formatierung: Johnny Graber (*linux@jgraber.ch*)
Lizenz: GFDL

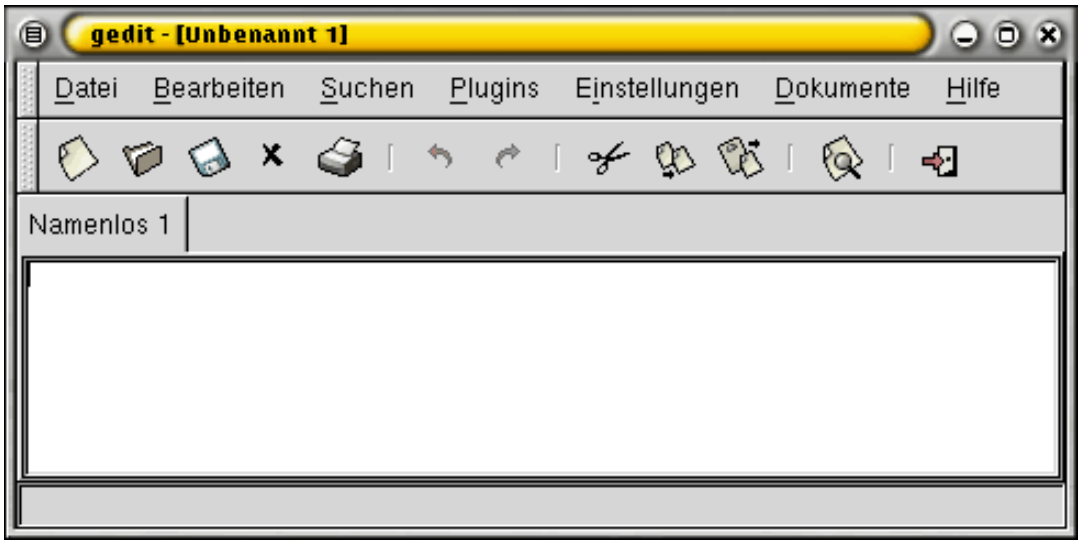
Editoren, genauer Texteditoren, gibt es unter Linux wie Sand am Meer. Da Editoren für das Arbeiten mit Computern unerlässlich sind, haben sich viele Leute ihren eigenen geschrieben und verteilen diesen über das Web. Zudem ist das Erstellen eines einfachen Editors eine beliebte Aufgabe zum Lernen einer Programmiersprache.

Dieses Kapitel soll es dem Leser ermöglichen, mit dem Texteditor seiner Wahl zu arbeiten. Da wir nicht alle Editoren beschreiben können, beschränken wir uns hier auf diese 5.

Mit dieser Auswahl sind fast alle Editoren abgedeckt, da alle anderen dem einen oder anderen davon sehr nahe kommen. Hier gibt es nun zu jedem dieser Editoren eine Übersicht. Für kwrite, emacs und den vi folgt zusätzlich noch eine ausführlichere Erklärung.

Inhaltsverzeichnis

1 gedit



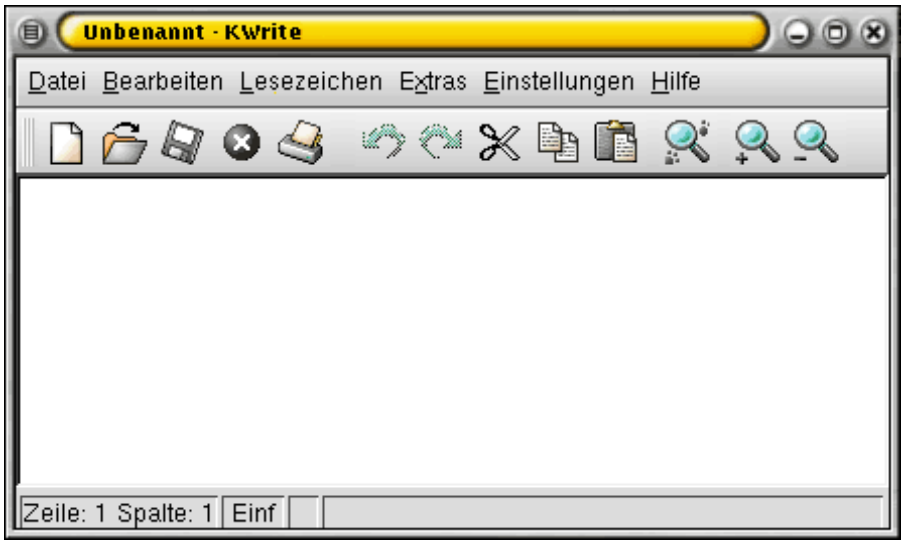
gedit: Klein aber Fein

`gedit` ist ein kleiner und schneller Editor. Wie es das G vermuten lässt, gehört er zum *GNOME Desktop*. Er bietet eine einfache Möglichkeit, kürzere Texte zu schreiben. Die grossen Icons bieten die wichtigsten Befehle gut erreichbar an. Damit kann ohne langes Suchen eine Datei geöffnet, gedruckt oder gespeichert werden. Auch die Möglichkeit, Änderungen rückgängig zu machen oder wieder herzustellen, liegt bereit und ermöglicht einem so ein zielgerichtetes Arbeiten.

Da einem `gedit` nur das Auswählen einer Schriftdarstellung erlaubt, bemerkt man sein hauptsächliches Einsatzgebiet. `gedit` ist gut, um ASCII-Text zu bearbeiten, er soll aber keinesfalls die Aufgaben einer Textverarbeitung übernehmen.

Alle Einstellungen laufen über das gleichnamige Menü. Neben der erwähnten Einstellung für die Schrift kann man auch das Aussehen von gedit festlegen sowie grundlegende Angaben fürs Drucken machen.

2 kwrite



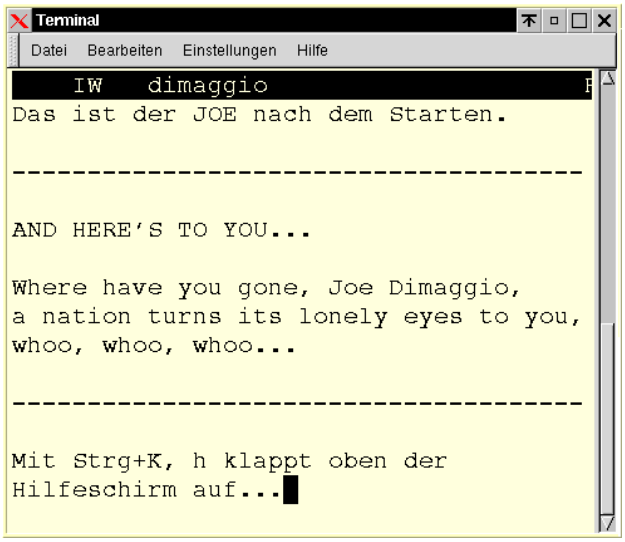
Der vielfältige kwrite in Aktion

Nachdem ein Editor für GNOME vorgestellt wurde, folgt nun mit `kwrite` einer von KDE. kwrite ist vom Funktionsumfang her gedit weit überlegen. Obwohl oberflächlich betrachtet kein grosser Unterschied auszumachen ist, bietet `kwrite` zahlreiche kleine Features, wie das Hervorheben von Code und die Verwendung von Lesezeichen.

Dennoch ist auch `kwrite` keine Textverarbeitung und hat, was das Auswählen der Schrift betrifft, die gleichen Limits wie `gedit`. Da auch deutlich mehr Einstellungen gemacht werden können, gibt es für `kwrite` bei Selflinux einen zusätzlichen Text, der einen tieferen Einblick ermöglicht.

* Das KWrite-Handbuch

3 joe



joe - nicht nur für Notfälle geeinget

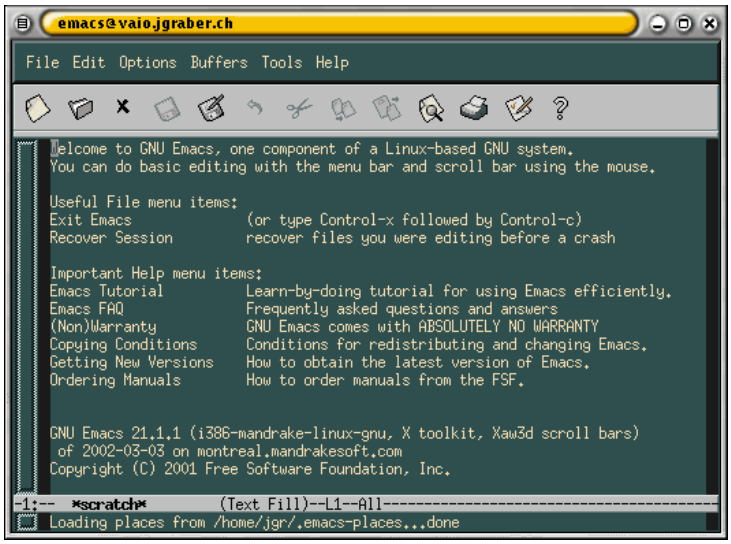
`joe` ist ein einfacher Editor für die Konsole. Er eignet sich insbesondere für User, die sich partout nicht den `vi` aneignen wollen, aber dennoch eine Möglichkeit suchen, von der Konsole aus einfache Bearbeitungen an Textdateien vorzunehmen, z. B. Konfigurationsdateien zu ändern. Mit `vi` gemeinsam hat `joe` den Vorteil, auch dann zu laufen, wenn sonst nichts mehr geht, und die genügsamen Ansprüche an die Hardware. Im Gegensatz zum `vi`, dessen Bedienungskonzept manchem Neuling als recht kryptisch erscheint, ist `joe` aber etwas konventioneller. Wieso der Name joe? joe ist ein rekursives Akronym für "Joe's Own Editor".

`joe` wird einfach von der Konsole aus gestartet mit `joe <Dateiname>` Dann erscheint der Dateiinhalt in Editierbereich. Zum Eingeben einfach die Tastatur verwenden, der Cursor kann mit den Cursortasten bewegt werden.

Mit der Tastenkombination `Strg + k, h` (hintereinander betätigen) Lässt sich ein kleiner Hilfebildschirm anzeigen, der alle Tastenkommandos beinhaltet; sie sind selbsterklärend. Mit `ESC` und anschließendem `,` oder `.` kann durch die Hilfeseiten vor- und zurückgeblättert werden.

`joe` legt von jeder geänderten Datei eine Sicherungskopie an, die aus dem Dateinamen mit angehängtem `~` (Tilde) besteht - für alle Fälle.

4 emacs



Gibt es etwas, das Emacs nicht kann?

Auch wenn `emacs` auf den ersten Blick nicht den Eindruck macht, gibt es nichts, was er nicht kann (ausser vielleicht Kaffee kochen...). Sei es nun das Schreiben von Texten, das Lesen von News oder der tägliche Emailbeantwortung - mit `emacs` ist dies alles möglich. Wären alle diese Funktionen direkt in emacs integriert, wäre dieser ein riesiges und schwerfälliges Monster. Daher sind die meisten Funktionen in Plugins ausgelagert und werden nur bei Gebrauch geladen.

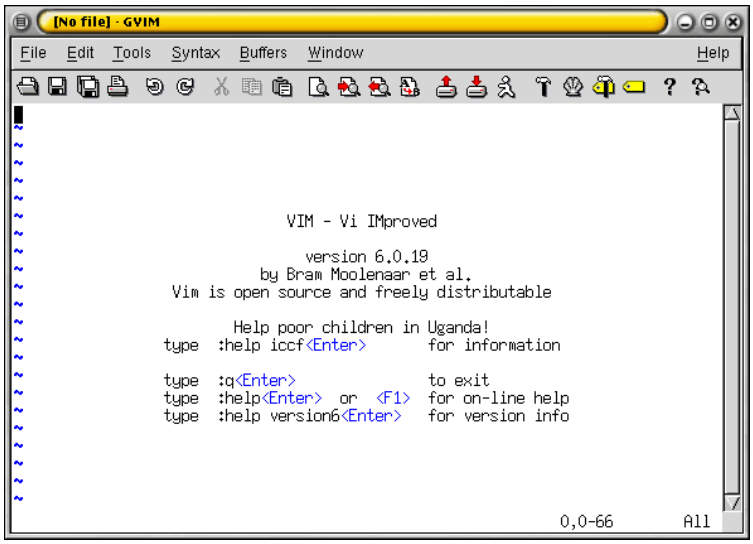
`emacs` hat eine lange Entwicklungsgeschichte. 1976 war `emacs` nur eine Sammlung von Makros, daher auch der Name (Editor MACroS). *Richard Stallman* ist nicht nur der Gründer von GNU, sondern auch der Vater von `emacs`. Nach dem Start von *GNU* war der Verkauf von emacs, der schon damals unter der *GPL* stand, lange Zeit die einzige Einnahmequelle der *FSF*.

Mit dem Ur-emacs hat der heutige aber kaum mehr etwas gemeinsam. Die stetige Weiterentwicklung brachte so viel Neues, dass die momentan aktuelle Version die Nummer 21.3 trägt. Die Entwicklung ist damit aber noch nicht abgeschlossen, und so wird es wohl nicht bei dieser Version bleiben.

Wer sich gerne eingehender mit `emacs` beschäftigen möchte, dem sei der folgende Text von Selflinux wärmstens empfohlen:

* HOWTO für den Emacs Einsteiger

5 vi



gvim bietet den Komfort eines grafischen Editors mit der Vielfältigkeit von vi.

vi ist wie emacs, ein Editor, der einem sehr viele Möglichkeiten bietet. Er ist sehr schnell, aber man muss etliche Stunden aufwenden, um mit ihm angenehm arbeiten zu können. Für diesen Aufwand wird man jedoch dann mit der Arbeitsgeschwindigkeit mehr als nur entschädigt.

Bis es soweit ist, ist es aber ein steiniger Weg. Vieles, was man bisher von Editoren wusste, gilt bei vi nicht mehr. Alles wird mit Hilfe der Tastatur gemacht, die Maus verliert in vi ihre Bedeutung als zentrales Hilfsmittel. Diesem Effekt ist es zu verdanken, dass vi immer zur Verfügung steht. Wenn nichts mehr geht, geht immer noch ein vi. Daher ist er Teil der meisten Rettungssysteme. Da man in einem Notfall nicht noch Zeit hat, sich in vi einzuarbeiten, sollte man dies schon vorher einmal machen.

Damit der Umstieg zu vi nicht zu schmerzhaft ist, gibt es eine Vielzahl von *vi-Clonen*. Der auf dem Screenshot gezeigte gvim ist eine angenehme Mischung aus der Stärke des vi mit der Einfachheit von gedit. gvim kann entweder wie vi über die Tastatur bedient werden oder auch mit Hilfe der Maus.

Wer einmal das Prinzip von vi und all seinen abgeleiteten Brüdern begriffen hat, will es nicht mehr missen. Damit auch die Leser von Selflinux einen tieferen Einblick bekommen, haben wir auch zu vi einen weiterführenden Text:

* Praxisorientiertes vim-Tutorial