

A Einführung

Das wahrscheinlich am häufigsten gebrauchte Tool bei der Linux-Systemadministration ist ein Editor zum Bearbeiten von Dateien – typischerweise macht man das von Kommandozeile aus (also in der Shell, häufig über eine SSH-Verbindung). Unter Linux gibt es zahlreiche Editoren: Vom zeilenorientierten `ed` über `vi`, dem UNIX-Standardeditor bzw. dessen Erweiterung `vim`, bis zu `emacs`.

Die Wahl eines, des “besten” oder des “richtigen” Editors kann zu erbitterten Debatten führen.

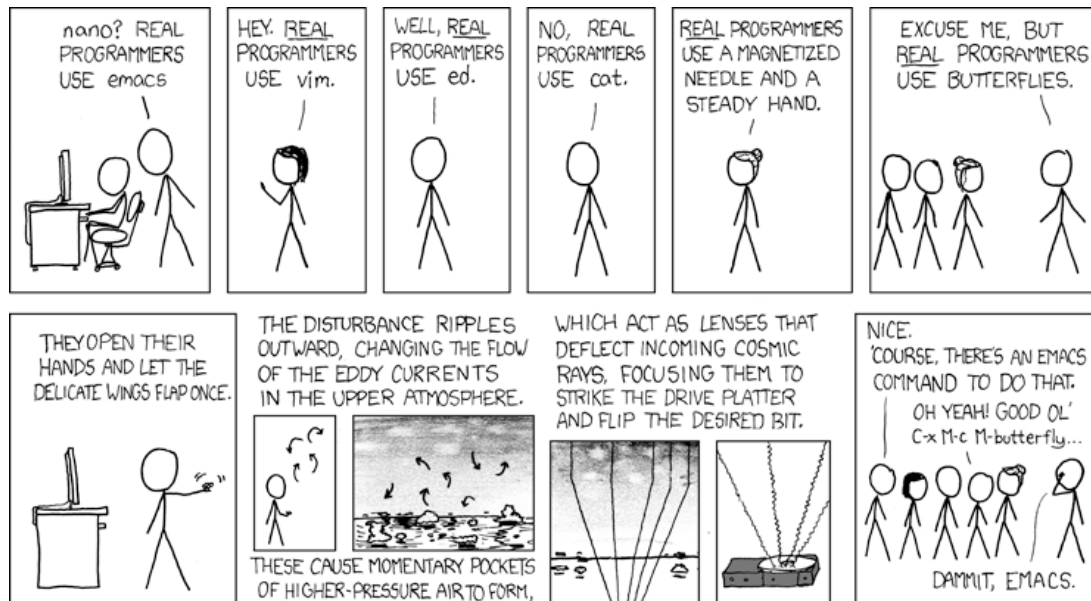


Figure 1: <https://xkcd.com/378/> CC BY-NC 2.5

Zusätzlich bringen die grafischen Oberflächen KDE und GNOME eigene Editoren mit. Automatisiertes d.h. programmgesteuertes Editieren ist übrigens mit dem *Stream-Editor* `sed` aber auch mit `emacs` oder `vi` möglich.

Wir wollen die grundlegende Bedienung einiger Editoren kennenlernen – schließlich sind das wesentliche *Survival Skills* der IT-Spezialisten!

B Übungsablauf

In dieser Übung befassen Sie sich mit den ersten beiden der drei vorgestellten Editoren näher, nämlich dem `nano` sowie dem `vi` (Sie können sich natürlich auch alle 3 anschauen ;-). Bei beiden Editoren sollten Sie:

- die Einführung durchlesen und, so vorhanden, das Tutorial durcharbeiten, dann
- die “Fragen” zum Editor beantworten sowie
- mit einem Editor Ihrer Wahl die *Editieraufgabe* durchführen!

C nano

`nano` ist - wie `vi` - auch bei vielen Distributionen vorinstalliert und zeichnet sich durch sehr einfache Benutzbarkeit aus. Er wurde ursprünglich als Klon des anfangs nicht-freien Editors `pico` entwickelt. Fortgeschrittene Funktionen und Features wie in `vi` oder gar `emacs` sind aber mit `nano` manchmal schwieriger oder gar nicht vorhanden.

C.1 Tutorial/Einführung (nano)

Sie finden die allerwichtigsten Befehle im *Editor Survival Guide*.

Ausführlichere Informationen schlagen Sie im Internet nach – wie beispielsweise unter:

- <https://wiki.ubuntuusers.de/Nano>
- <https://wiki.gentoo.org/wiki/Nano>

C.2 Fragen (nano)

- (1) Kopieren Sie eine längere Datei auf der Kommandozeile in Ihr Heimatverzeichnis, etwa `/etc/services` und öffnen Sie diese mit dem Editor.

Beantworten Sie (z.B. unter Verwendung des *Editor Survival Guide*, der Online-Hilfe im Editor oder durch Web-Recherche) dann - mit Ausprobieren - die Fragen:

- (2) Welche Tastenkombination beendet nano?
- (3) Wie bewegt man sich eine Seite vor und zurück? Kann man mit einem Befehl 5 Seiten vorwärts springen?¹
- (4) Wie kommt man zum Ende der Datei bzw. wieder zurück zum Anfang?
- (5) Wie löscht man eine Zeile und fügt Sie (woanders) wieder ein?
- (6) Wie macht man einen Schritt rückgängig?
- (7) Wie speichert man eine Datei?
- (8) Wie sucht man nach einem Text?
- Suche / weitersuchen
 - Suche / weitersuchen mit RegEx
- (9) Wie ersetzt man einen Text durch einen anderen Text?
- (10) Wie öffnet man eine weitere Datei?
- (11) Wie zeigt man alle Dateien in einem Ordner an und öffnet eine davon?

D vi

`vi`² ist ein text-basierter *Full Screen*-Editor, der Teil des *POSIX*-Standards ist und daher auf praktisch allen UNIX-ähnlichen Systemen, insbesondere auch Linux in der Regel *vorinstalliert* ist. Daher sollte man zumindest Grundkenntnisse der Bedienung kennen, um im Notfall z.B. eine kleine Änderung der Systemkonfiguration in einer Konfigurationsdatei vornehmen zu können.

Die Besonderheit des `vi`-Editors (die anfangs gelernt werden muss) ist, dass er zwei “Betriebsarten” kennt:

1. Befehlsmodus (mit `<Esc>`)
2. Eingabemodus (mit `i`, `I`, `a`, `o`, `O` ...)

Im *Befehlsmodus* kann man navigieren (sich in der Datei frei bewegen), löschen, abspeichern und dergleichen. Um Text einzugeben oder zu überschreiben, muss man in den *Eingabemodus* wechseln, was man z.B. mit ‘`i`’ (für *insert*) machen kann. Nach Eingabe/Änderung des Textes wechselt man stets mit der `<ESC>`-Taste (*Escape*) wieder in den Befehlsmodus.

¹Achtung: nicht jeder Editor kann alles ;-)

²`vi` steht für “visual” – Der Name erinnert daran, wie bahnbrechend innovativ der Editor zum Zeitpunkt seiner Programmierung war

Das ist anfangs etwas ungewohnt, nach einigem Arbeiten tippt man aber “automatisch” nach Ende jeder Eingabe/Änderung die <ESC>-Taste, um wieder in den Befehlsmodus zu wechseln (um dann z.B. woanders hin zu navigieren).

Genau genommen gibt es auch einen *dritten* Modus, den ex-Modus, der einen sehr mächtigen Zeileneditor (namens ex) integriert. Alle Befehle, die mit : beginnen, werden der ex-Kommandozeile übergeben. Dies erfolgt in der Praxis direkt über : aus dem Befehlsmodus und ist daher kein Zusatzaufwand. So kann man beispielsweise mit :w (*write*) die Datei abspeichern und mit :q! (*quit unconditionally*) den Editor ohne Änderungen verlassen.

D.1 Tutorial/Einführung (vi)

(12) Arbeiten Sie das Online-Tutorial durch: <http://www.openvim.com!>

D.2 Fragen (vi)

(13) Kopieren Sie eine längere Datei in Ihr Heimatverzeichnis, etwa /etc/services und öffnen Sie diese mit dem Editor.

Beantworten Sie (z.B. unter Verwendung des *Editor Survival Guide*, der Online-Hilfe im Editor oder durch Web-Recherche) dann - mit Ausprobieren - die Fragen:

- (14) Welche Tastenkombination beendet vi?
- (15) Wie bewegt man sich eine Seite vor und zurück? Und 5 Zeilen?
- (16) Wie kommt man zum Ende der Datei bzw. wieder zurück zum Anfang?
- (17) Wie löscht man eine Zeile und fügt Sie (woanders) wieder ein?
- (18) Wie macht man einen Schritt rückgängig?
- (19) Wie speichert man eine Datei?
- (20) Wie sucht man nach einem Text?
 - Suche / weitersuchen
 - Suche / weitersuchen mit RegEx
- (21) Wie ersetzt man einen Text durch einen anderen Text?

D.3 vi für Fortgeschrittene

- (22) Was bewirkt – im Befehlsmodus – der Befehl ~ (*Tilde*), wenn man ihn über Text (auch mehrfach) ausführt?
- (23) Wie kann man die aktuelle Zeile (kann auch leer sein) mit der Ausgabe eines Befehls (z.B. dem aktuellem Datum von date) ersetzen?
- (24) Wie kann man alle Zeichen von der aktuellen Cursor-Position bis zum Zeilenende *löschen* und woanders einfügen?
- (25) Wie kann man die nächsten 3 Wörter löschen (oder nur kopieren) und woanders einfügen?
- (26) *Bonus*: Wie öffnet man eine weitere Datei (in einem neuen Dateipuffer)?
- (27) *Bonus* Wie zeigt man alle Dateien in einem Ordner und öffnet eine davon?

E Editieraufgabe

Wichtig: Führen Sie diese Aufgabe erst durch, *nachdem* Sie sich mit zumindest den beiden Editoren nano und vi vertraut gemacht haben und deren Fragen durchgearbeitet haben!

Wählen Sie einen Editor aus (Vollprofis können auch den Emacs verwenden – siehe unten – verwenden), mit dem Sie folgende poetische Aufgabe erfüllen:

- (28) Mit welchem Editor möchten Sie die Aufgaben erledigen?
- (29) Sie erhalten eine - leider ein klein wenig verwürfelte Version des Gedichts “Der Zauberlehrling” - die auch noch ein paar Tippfehler enthält - von Johann Wolfgang von Goethe (Datei: zauberlehrling.txt). Bringen Sie das Gedicht in seine ursprüngliche Form (*Tipp:* Es sind nur 2 Absätze am falschen Platz), und korrigieren Sie die paar Tippfehler!
- (30) Realisieren Sie eine IT/Netzwerktechnik-relevante *Modernisierung* des Zauberlehrlings (inklusive Einrückungen). Als *Vorlage* bekommen Sie von Ihrem freundlichen Betreuer eine Hardcopy zur Verfügung gestellt bzw. verwenden die beigefügte Datei netzwerkmeister.png bzw. netzwerkmeister.pdf (und Sie haben gerade kein OCR³ zur Hand - bitte nicht einfach scannen!). Verwenden Sie alle Ihre Editor-Zauberkünste, um Ihren Text an den Inhalt der Hardcopy anzupassen (*Ergebnis mit kleinem Monospace-Font ins Protokoll!!*)!

F Bonus: Emacs

F.1 Über Emacs

Dieser Editor ist ein mächtiges Werkzeug, die Bezeichnung Editor ist für diese Arbeitsumgebung fast eine Untertreibung. Es ist ein sehr umfangreiches Programm mit den wahrscheinlich meisten Features eines einzelnen UNIX-Programms.

Emacs basiert auf einer eigenen Programmiersprache (Lisp bzw. Emacs-Lisp) für Erweiterungen und Makros (die meisten Pakete des Emacs sind in dieser Sprache geschrieben).

Es können beliebig viele Dateien gleichzeitig bearbeitet werden. Neben den Funktionen zum Bearbeiten von Texten (Einfügen, Kopieren, etc.) gibt es eine Undo Funktion (beliebig weit), inkrementelles Suchen (Suchergebnisse werden während der Eingabe des Suchbegriffs angezeigt), Autosave und Menüs.

Die enthaltenen Erweiterungen decken fast alle Bereiche ab: z.B. Programme schreiben, übersetzen und testen (wie eine integrierte Entwicklungsumgebung), alle Arten der Kommunikation (Mail, News, WWW) sind am Textschirm und unter X Windows möglich. Für sehr viele Arbeiten gibt es spezielle Unterstützung (so genannte Modes), z.B. je ein Mode für die gängigen Programmiersprachen (einrücken, farbige Darstellung der Schlüsselwörter etc.), HTML Mode zum Gestalten von WWW-Dokumenten usw.

Häufig verwendete Kommandos sind durch “**shortcuts**” (eine Kombination der Tasten Meta, Ctrl und einer weiteren Taste) abrufbar.

Alles in allem: Es gibt fast nichts, was Emacs nicht kann.

Diese Funktionalität hat auch ihren Preis: Emacs braucht viel Speicher (bei der Installation und bei der Ausführung) und ist für den “Emacs-Neuling” durch die vielen Möglichkeiten eher entmutigend.

³O ptical C haracter R ecognition extrahiert Text aus Bilddateien

F.2 Bearbeiten

Der angezeigte Schirm ist in drei Teile geteilt: den Buffer zum Bearbeiten der Datei, eine Statuszeile und die unterste Zeile zur Eingabe von Kommandos und zur Anzeige von Meldungen.

Die Bewegung des Cursors geschieht am einfachsten durch die Cursortasten.

Außerdem können alle Kommandos in der untersten Zeile direkt eingegeben werden: durch Eingabe von M-x (M steht dabei für Meta – falls sie keine solche Taste auf der Tastatur finden, kann dies entweder durch vorheriges Drücken der ESC oder gleichzeitiges Drücken der ALT Taste ersetzt werden, d.h. ESC X oder ALT X und Eingabe des Kommandonamens. Bei vielen Eingaben funktionieren die TAB Taste zur Ergänzung und die Cursortasten zur Auswahl der vorigen Eingaben (Commandhistory).

Beispiel:

Öffnen einer Datei: M-x find-file. Man kann aber auch C-x C-f (Ctrl - X gefolgt von Ctrl - F) verwenden.

Die Taste C-g bricht jedes Kommando ab.

Die zweite Zeile von unten ist die so genannte Modeline (Statuszeile). Dort findet man Informationen über die aktuelle Datei, den für die Bearbeitung der Datei gewählten Modus usw.

Beispiel:

```
--*- main.C  Top L2 (C++)--
```

** die Datei wurde geändert (%% wenn die Datei nur zum Lesen geöffnet ist)

C++ ausgewählte Modi

L2 aktuelle Zeile

TOP Beginn der Datei

F.3 Text löschen und verschieben

Das Markieren eines Bereiches (Region) geschieht entweder mit der Maus oder mit der Tastenkombination C-@ (oder C-Space) und bewegen des Cursors.

C-w löscht den Bereich, M-w kopiert den Bereich (ohne Löschen) – der Text kommt dabei in den Kill-Ring (Zwischenablage mit beliebig vielen Einträgen). Mit dem Befehl C-y (yank) kann dieser Text an anderer Stelle eingefügt werden; mit M-y können dann alle bisherigen Einträge des Kill-Rings durchgegangen werden.

F.4 Mehr als Dateien – Modes

Der Editor erkennt aufgrund des Dateinamens den Dateityp. Damit können verschiedene Daten auch verschieden gehandhabt werden. Die unterschiedlichen Modi der Bearbeitung können auch durch Eingabe des entsprechenden Kommandos gestartet werden. Emacs unterscheidet dabei zwei Arten:

Major-Mode : Dieser bestimmt die Art der Bearbeitung z.B. C-Mode, HTML-Mode etc. Es kann dabei immer nur einer dieser Modi aktiv sein.

Minor-Mode : Dieser bestimmt zusätzliche Funktionalitäten bei der Bearbeitung, z.B. Overwrite Modus, Fontlock Modus etc. Diese Modi können beliebig miteinander und den Major Modi kombiniert werden.

Text : Fast alle Programmiersprachen und Textbeschreibungssprachen (z.B. HTML, TeX) werden erkannt und durch einen entsprechenden Major-Mode unterstützt: Es gibt spezielle Befehle zum Bearbeiten und Bewegen (z.B.

gehe zum Ende der Funktion) und spezielle Menüs. Meist gibt es Kommandos zum schnellen Einfügen von oft gebrauchten Konstrukten.

Verzeichnisse : Directory Editor (dired-mode): Bearbeiten des Verzeichnisses (Löschen, Umbenennen, Öffnen usw.)

tar (Archiv), ZIP Datei etc. : Bearbeiten der enthaltenen Dateien ähnlich dem dired-mode.

mail : Lesen und Schreiben von Email

compile : Ein eigener Modus zum Compilieren eines Projekts. Durch die Auswertung der Fehlermeldung können die entsprechenden Fehlerstellen schnell angesprungen werden.

font-lock : Farbige Darstellung der Schlüsselworte

F.5 Hilfe

Mit der Eingabe von C-h kommt man in das Hilfesystem.

C-h C-h zeigt die Liste aller Möglichkeiten

C-h t für den Anfänger gibt es auch eine kurze Einführung (Tutorial)

C-h m beschreibt die derzeitigen Modi und die entsprechenden Tastenbelegung

C-h a zeigt alle zu einem Begriff passenden Befehle

M-x apropos erweiterte Suche nach einem Begriff

C-h i startet man das Infosysteminfo, das außer dem Editor viele GNU-Programme beschreibt. Die Information zur Benutzung des Editors ist sehr umfangreich; sie entspricht dem gedruckten Benutzerhandbuch mit ca. 450 Seiten. Zusätzlich gibt es ein ca. 800-seitiges Programmierhandbuch

F.6 Tutorial/Einführung (emacs)

(31) Starten Sie emacs – eventuell muss dazu das (Meta-)Paket emacs nachinstalliert werden.

Man kann emacs als GUI-Programm verwenden oder nur in einem Terminal starten: `emacs -nw`

(32) Wählen Sie das Tutorial und arbeiten Sie alle Schritte durch – wobei die den letzten Teil ab “REKURSIVE EDITIER-EBENEN” überspringen können.

F.7 Fragen (Emacs)

(33) Kopieren Sie - falls noch nicht erfolgt - eine längere Datei in Ihr Heimatverzeichnis, etwa `/etc/services` und öffnen Sie diese mit dem Editor.

Anschließend beantworten Sie die Fragen:

(34) Welche Tastenkombination beendet emacs?

(35) Wie bewegt man sich eine Seite vor und zurück? Und 5 Zeilen?

(36) Wie kommt man zum Ende der Datei bzw. wieder zurück zum Anfang?

(37) Wie löscht man eine Zeile und fügt Sie (woanders) wieder ein?

(38) Wie macht man einen Schritt rückgängig?

(39) Wie speichert man eine Datei?

(40) Wie sucht man nach einem Text?

- Suche / weitersuchen / Inkrementelle Suche⁴
- Suche / weitersuchen / Inkrementelle Suche mit RegEx

(41) Wie ersetzt man einen Text durch einen anderen Text?

(42) Wie öffnet man eine weitere Datei?

(43) Wie zeigt man alle Dateien in einem Ordner und öffnet eine davon?

F.7.1 Emacs für Fortgeschrittene

(44) Was macht M-x kill-rectangle?

(45) Was macht M-x grep-find?

(46) Wann ist heute Sonnenaufgang? (Tipp: M-x calendar, C-h m, C-s sunset, oder direkt M-x sun...) Was würde ein Maya heute sagen? *Bonus:* Wie kann man die für den Sonnenaufgang notwendigen Informationen dauerhaft speichern?

(47) Im Buffer `_scratch_`: M-x picture-mode (Tipp: C-h m)

(48) Was macht M-x hexl-find-file?

(49) M-x tetris, M-x doctor, https://www.gnu.org/software/emacs/manual/html_node/emacs/Amusements.html

G Superbonus: sed

Das Programm sed ist ein (sehr mächtiger) Stream-Editor. Es liest die Eingabedateien (oder die Standardeingabe) zeilenweise und führt entsprechende Veränderungen (Ersetzen, Umordnen mittels Hilfspuffern etc.) durch. Anweisungen können für bestimmte Zeilen bzw. für gewisse Bereiche definiert werden. Sie können mit sed ganze Skripte mit Editier-Anweisungen schreiben!

Beispielhafte Befehle:

`d` : löscht die Zeile (keine Ausgabe)

`s/alt/neu/opt` : ersetzen des Musters

statt / ist jedes andere Sonderzeichen möglich. Die im Muster geklammerten Ausdrücke, können beim Ersetzen durch `\1`, `\2` usw. angesprochen werden

Option `g`: eventuell mehrmals ersetzen

`y/set1/set2/` : ersetze die Zeichen aus set1 durch das entsprechende Zeichen in set2

`w datei` : Ausgabe der Zeile in eine Datei

`h` : Kopiert die aktuelle Zeile in einen Hilfsbereich

`x` : vertauscht aktuelle Zeile und Hilfsbereich

Die Befehle können mit der Option `-e` (auch mehrmals möglich) direkt auf der Kommandozeile angegeben werden oder aus einer Datei gelesen werden.

Beispiel: Aus einer Datei sollen führende Zeilennummern entfernt werden:

```
sed -e 's/^ *[0-9]*: \(.*\) /\1/'
```

⁴d.h. Fundstellen werden schon während der Eingabe des Suchbegriffs angezeigt

G.1 Tutorial/Einführung (sed)

Ein brauchbares Online-Tutorial findet sich z.B. unter <http://www.grymoire.com/Unix/Sed.html#uh-0>

G.2 Aufgaben

- (50) Ersetzen Sie in der zweiten Zeile einer Datei alle a durch A
- (51) Wie kann man in einer Datei den Text HTL durch Höhere Technische Lehranstalt ersetzen (*Tipp: in-place editieren mit -i*)?
- (52) Alternative: auch mit perl kann man *in-place* suchen/ersetzen.