>> vi lernen -- die "Spickzettel"-Methode

[Bitte Kapitel auswählen]

1. Und los geht's

1.1 Einführung

Dieses Tutorial wird Ihnen zeigen, wie man den mächtigen visuellen Editor vi benutzt. Anhand einer speziellen beschleunigten "Spickzettel"-Methode, versucht dieses Tutorial Sie zu einem fortgeschrittenen vi-Benutzer zu machen, und das ohne viel Zeit in Anspruch zu nehmen. In diesem vi-Tutorial werden Sie lernen sich innerhalb von Texten zu bewegen, Text zu bearbeiten, den insert-mode zu benutzen, Text zu kopieren und wieder einzufügen, und natürlich solch wichtige vim-Erweiterungen wie den Visual-Mode und Multi-Window-Editing.

Wenn Sie vi nicht kennen oder Ihnen das Arbeiten damit unbequem erscheint, dann können Sie dieses Tutorial dazu benutzen, um Ihr Arbeiten mit einem der beliebtesten und mächtigsten Editoren im Linux/UNIX-Sektor zu beschleunigen.

1.2 Über den Leitfaden und das vi Lernen beschleunigen

Es gibt eine Sache im Speziellen, die es erschwert, vi zu lernen -- vi hat unglaublich viele Kommandos. Um vi also effektiv nutzen zu können, müssen Sie sich einige davon merken. Dies kann eine lange Zeit in Anspruch nehmen, und eines der Ziele dieses Tutorials ist eben nicht besonders viel Zeit zu beanspruchen. Für uns ist das zu Beginn eine grosse Herausforderung: Wie kann ich Ihnen helfen, diese Kommandos innerhalb kurzer Zeit im Kopf zu haben?

Um dieser Herausforderung entgegenzutreten während wir durch dieses Tutorial voranschreiten, werden wir uns Stückchen für Stückchen einen Spickzettel zusammenstellen. Dieser Zettel wird alle wichtigen vi- Kommandos enthalten. Nachdem Sie dieses Tutorial abgeschlossen haben, wird es Ihnen möglich sein auf diesen Spickzettel zurückzugreifen, wenn Sie ein bestimmtes Kommando vergessen haben. Nach einiger Zeit werden sich die Kommandos in Ihrem Kopf festgesetzt haben, und sie werden immer unabhängiger von diesem Spickzettel werden. Mit der Spickzettel-Methode ist es möglich, vi schneller zu lernen als jemals zuvor.

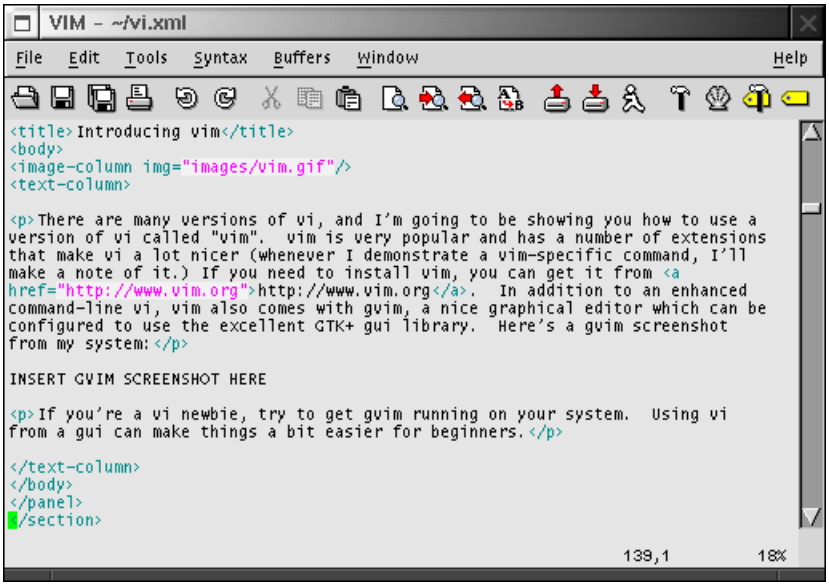
1.3 Der Lernprozess

In diesem Leitfaden werde ich einige Methoden benutzen um Ihnen beim Lernen zu helfen. Zuallererst beschreibe ich, wie ein bestimmtes Kommando funktioniert. Dann werde ich Sie auffordern, einmal selbst in vi dieses Kommando auszuprobieren (zur Übung), und dann werde ich Sie auffordern, das Kommando Ihrem Spickzettel hinzuzufügen. Wenn Sie vi schnell lernen möchten ist es wichtig, dass Sie jeden dieser Schritte auch durchführen. Das Kommando auszuprobieren und dann auf dem Spickzettel zu vermerken wird Ihnen helfen, sich das Kommando zu behalten.

1.4 Einführung in vim

Es gibt viele Versionen von vi, und ich werde Ihnen zeigen, wie man eine Version namens "vim" benutzt. vim ist sehr beliebt und besitzt einige Erweiterungen, die vi viel interessanter machen (wann immer ich ein vim-spezifisches Kommando benutze, werde ich es anzeigen). Wenn Sie vim installieren möchten, dann können Sie es von www.vim.org herunterladen. Zusätzlich zum erweiterten Kommandozeilen-vi, kommt auch gvim mit auf den Rechner, ein netter grafischer Editor, der so eingestellt werden kann, dass er die ausgezeichnete GTK+ GUI-Bibliothek benutzt. Hier ein gvim-Screenshot von meinem System:

Grafik 1: VIM Screenshot



Wenn Sie ein vi-Neuling sind, dann versuchen Sie, gvim auf Ihrem System zum Laufen zu bekommen. Von einer GUI aus sind die Dinge etwas einfacher für Anfänger.

2. Erste Schritte

2.1 Eine Datei benutzen

Bevor Sie vi benutzen um Dateien zu bearbeiten, müssen Sie erst einmal wissen, wie Sie sich mit vi innerhalb des Textes bewegen können. vi hat eine Menge von Bewegungs-Kommandos, und wir werden uns viele von diesen ansehen. Für diesen teil des Tutorials suchen Sie sich bitte eine unwichtige Textdatei und öffnen diese mit vi, indem Sie folgendes am Kommandoprompt Ihrer Shell eingeben:

Befehlsauflistung 1

```
$ vi meinedatei.txt
```

Wenn Sie vim installiert haben, dann geben sie *vim meinedatei.txt* ein. Wenn Sie die Benutzung von gvim bevorzugen, dann geben Sie bitte *gvim meinedatei.txt* ein. meinedatei.txt ist durch den Namen einer einfachen Textdatei auf Ihrem System zu ersetzen.

2.2 vi intern

Nachdem vi geladen ist, sollten Sie einen Teil der Textdatei auf dem Bildschirm sehen. Herzlichen Glückwunsch! - Sie befinden sich nun in vi. Im Gegensatz zu anderen Editoren, befindet sich vi nach dem Start in einem speziellen so genannten "Command-Mode". Das heisst, dass wenn Sie nun *l* auf Ihrer Tastatur drücken, können Sie den Cursor einen Zeichen nach rechts bewegen, anstatt den Buchstaben *l*in den Text einzufügen. Im Command-Mode werden die Buchstaben auf Ihrer Tastatur dazu benutzt, Kommandos an vi zu senden anstatt Buchstaben in den Text einzufügen. Die essentiellsten aller Kommandos sind die zur Bewegung des Cursors. Lassen Sie uns einige ansehen.

3. Im Text bewegen

3.1 Mit vi im Text bewegen, Teil 1

Im Command-Mode können Sie die *h*,*j*,*k* und *l* Tasten benutzen, um den Cursor nach links, unten, oben und rechts zu bewegen. Wenn Sie eine moderne Version von vi benutzen, dann können Sie außerdem auch die Pfeiltasten zu diesem Zweck benutzen. Die Tasten *h*,*j*,*k* und *l* sind recht nützlich, denn wenn Sie sich an diese gewöhnt haben, dann können Sie sich innerhalb der Datei bewegen ohne Ihre Finger vom Hauptfeld der Tastatur nehmen zu müssen. Veruschen Sie *h*,*j*,*k* und *l* (und die Pfeiltasten) zu benutzen, um sich innerhalb des Textes zu bewegen. Versuchen Sie *h* zu benutzen um an den Beginn einer Zeile zu springen. Beachten Sie bitte, dass vi es nicht erlaubt, zum Ende der vorherigen Zeile zu springen, wenn Sie sich am Beginn einer Zeile befinden und dann *h* drücken. Das selbe gilt für das Springen auf die nächste Zeile, wenn Sie sich am Ende einer Zeile befinden.

3.2 Mit vi im Text bewegen, Teil 2

vi bietet spezielle Abkürzungen um zum Anfang oder Ende einer zeile zu springen. Sie können *0* (Null) drücken, um zum ersten Zeichen in einer Zeile zu springen, und sie können *$* benutzen, um zum letzten Zeichen in einer Zeile zu springen. Probieren Sie die Kommandos aus und sehen Sie, was passiert. Da vi so viele nützliche Bewegungs-Kommandos beherrscht, ist er der ideale "Pager" (wie "less" oder "more"). vi als Pager zu benutzen wird Ihnen ausserdem helfen, die Bewegungs-Kommandos schneller zu lernen.

Sie können auch *<Strg>F* und *<Strg>B* benutzen, um sich seitenweise vorwärts und rückwärts zu bewegen. Moderne Versionen von vi (wie vim) erlauben es auch BildAuf und BildAb für diesen Zweck zu benutzen.

3.3 Wortsprünge, Teil 1

vi erlaubt es auch, den Cursor wortweise links oder rechts zu bewegen. Um zum **ersten** Zeichen des nächsten Wortes zu gehen, drücken Sie *w*. Um zum **letzten** Zeichen des nächsten Wortes zu gehen, drücken Sie *e*. Um zum ersten Buchstaben des **vorherigen** Wortes zu gelangen, drücken Sie *b*. Probieren Sie es aus!

3.4 Wortsprünge, Teil 2

Nachdem Sie nun ein bisschen mit den Kommandos zur wortweisen Bewegung herumgespielt haben, haben Sie vielleicht festgestellt, dass vi Wörter wie "foo-bar-oni" als fünf seperate Wörter ansieht. Dies ist so, weil vi von Hause aus Wörter anhand von Leer- **oder** so genannten Delimiterzeichen trennt. Aus diesem Grund wird foo-bar-oni als fünf Wörter angesehen: "foo","-","bar","-" und "oni".

Manchmal ist es genau das, was man möchte, und manchmal kann man dieses Feature überhaupt nicht gebrauchen. Glücklicherweise versteht vi auch das Konzept von "bigword". vi trennt bigwords anhand von **Leerzeichen oder Zeilenumbrüchen**. Dies bedeutet, dass foo-bar-oni zwar als 5 Wörter angesehen wird aber nur als ein bigword.

3.5 Wortsprünge, Teil 3

Um zum nächsten und vorherigen Bigword zu springen, können Sie ein grossgeschriebenes Wortbewegungskommando benutzen. Benutzen Sie *W*, um zum ersten Zeichen des nächsten Bigwords zu springen, *E*, um zum letzten Zeichen des nächsten Bigwords zu springen und *B*, um zum ersten Zeichen des vorhergehenden Bigwords zu springen. Probieren Sie es aus und vergleichen Sie die entsprechenden Wort- und Bigwordbewegungskommandos bis Sie die Unterschiede verstehen.

3.6 Grössere Sprünge

Wir müssen noch ein paar Kommandos mehr behandeln, bevor wir unseren Spickzettel zusammenstellen können. Sie können die *(* und *)* benutzen um zum Anfang des vorhergehenden und des nächsten Satzes zu springen. Zusätzlich dazu können Sie *{* oder *}* benutzen, um zum Anfang des aktuellen bzw. des nächsten Absatzes zu springen. Auch hier ist wieder einmal ausprobieren angesagt.

4. Beenden

4.1 Beenden

Wir haben die grundsätzlichen Bewegungskommandos behandelt, aber es gibt noch weitere Kommandos die Sie kennen müssen. Die Eingabe *:q* beendet vi. Sollte dies nicht funktionieren, dann haben Sie vielleicht die Datei geändert. Um vi nun zu beenden und die evtl. gemachten Änderungen zu verwerfen, müssen Sie *:q!* eingeben. Sie sollten nun wieder zurück auf Ihrer Shell sein.

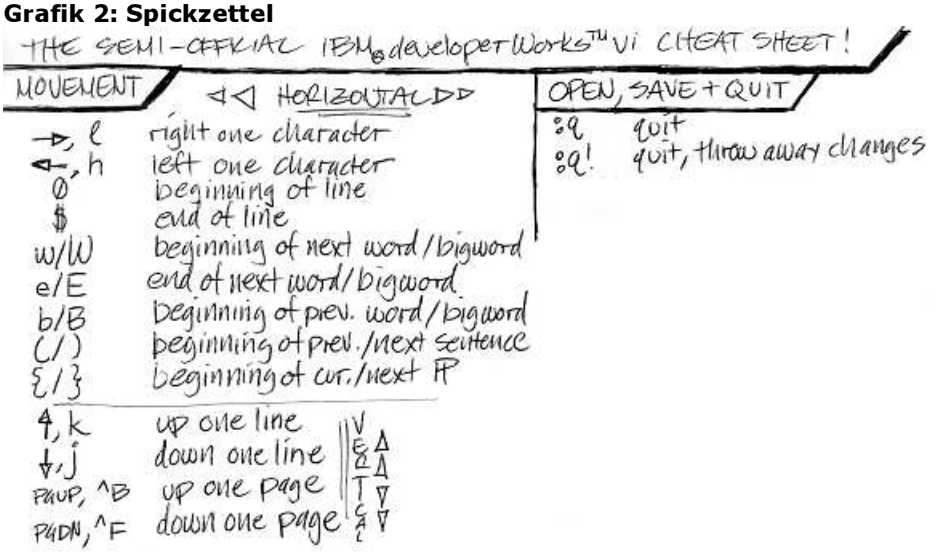
In vi ist jedes mit einem ":" beginnenden Kommandos ein sogenanntes **ex-mode** Kommando. Der Grund hierfür ist, dass vi einen nicht visuellen Editor mit dem Namen **ex** eingebaut hat. Er kann ähnlich wie sed dazu benutzt werden um zeilenbasierte Bearbeitungsfunktionen durchzuführen. Desweiteren kann er auch dazu benutzt werden um vi zu beenden - wie wir eben gerade gesehen haben. Sollten Sie jemals die Taste *Q* drücken, während Sie sich im Kommandomodus befinden, dann wird vi sofort in den ex Modus schalten. Wenn dies passiert, wird Ihnen ein : Prompt präsentiert, das Drücken von Enter wird den ganzen Bildschirm nach oben rollen. Um wieder in den guten alten vi-Modus zurückzukommen, geben Sie einfach nur vi ein und drücken Enter.

5. Der Spickzettel

5.1 Die Anfänge des Spickzettels

Wir haben einige Kommandos behandelt und nun ist es Zeit, diese auf unseren Spickzettel zu übertragen. Für den Spickzettel brauchen Sie ein Blatt Papier (wir werden einige Informationen auf diesem Blatt vermerken!). Hier ist ein Bild meines Spickzettels, nachdem ich alle Kommandos aufgeschrieben habe, die wir bis hierhin behandelt haben. Versuchen Sie bitte, meinem Layout zu folgen, sodass alles auf ein Blatt Papier passt.

Grafik 2: Spickzettel



5.2 Vielseitiger vi

Lassen Sie uns mit unserem Kommandoschnelldurchlauf fortfahren. Im Kommandomodus können Sie zu einer bestimmten Zeile springen, indem Sie *G* verwenden. Um zur ersten Zeile in einer Datei zu springen, geben Sie *1G* ein. Bitte beachten Sie dass grosse G!

Wenn Sie zur nächsten Erscheinung eines bestimmten Textmusters springen möchten, geben Sie */ <regexp>* ein und drücken Sie anschliessend *Enter*. Ersetzen Sie <regexp> mit dem Regulären Ausdruck nach dem Sie suchen. Wenn Sie nicht wissen, wie man reguläre Ausdrücke benutzt, dann haben Sie keine Angst : Die Eingabe von */foo* wird zum nächsten Auftauchen von **foo** springen. Das einzige wo Sie aufpassen müssen sind die speziellen Zeichen **^**, **.**, **$** oder ****. Setzen Sie diesen Zeichen einen Backslash voran (****) und es sollte funktionieren. Zum Beispiel wird */foo\.gif* zum nächsten Auftauchen von von "**foo.gif**" springen.

Um die Suche in Vorwärtsrichtung fortzusetzen, drücken Sie *n* auf der Tastatur. Um rückwärts zu suchen drücken Sie *N*. Wie immer sollten Sie diese Kommandos in Ihrer vi-Umgebung testen. Sie können auch *//* eingeben, um die letzte Suche zu wiederholen.

6. Speichern und Bearbeiten

6.1 Speichern und Speichern unter...

Wir haben behandelt, wie das **ex** Kommando *:q* zu benutzen ist, um vi zu beenden. Wenn Sie die gemachten Änderungen Speichern wollen, dann geben Sie *:w* ein. Wenn Sie die Änderungen in einer anderen Datei speichern wollen, dann geben Sie *:w dateiname.txt* ein, um in der Datei dateiname.txt zu speichern. Wenn Sie Speichern und Beenden wollen, so geben Sie *:x* oder *:wq* ein.

In vim (und anderen fortgeschrittenen vi Editoren wie elvis) können Sie mehrere Puffer auf einmal geöffnet haben. Um eine Datei in einem neuen Fenster zu öffnen, geben Sie *:sp dateiname.txt* ein. dateiname.txt wird in einem neuen geteilten Fenster erscheinen. Um zwischen Fenstern hin und her zu springen, geben Sie *<Strg>w<Strg>w* (also zweimal Strg-w) ein. Alle *:q*, *:q!*, *:w* und *:x* -Eingaben, die Sie machen, werden jeweils nur auf das momentan aktive Fenster angewendet.

6.2 Einfache Änderungen

Nun wird es Zeit einige der einfachen Bearbeitungskommandos kennenzulernen. Die Kommandos, die wir hier behandeln, werden "einfache" Bearbeitungskommandos genannt, da Sie sie im Kommandomodus verwenden. Die komplexeren Bearbeitungskommandos bringen Sie in den Einfügemodus (insert mode) - ein Modus, der es Ihnen erlaubt Text von der Tastatur ganz normal einzugeben. Wir werden gleich darauf zurück kommen.

Versuchen Sie einmal, den Cursor über einige Zeichen zu stellen und die Eingabe von *x* zu wiederholen. Sie werden sehen, dass dies jeweils den aktuellen Buchstaben am Cursor löschen wird. Nun bewegen Sie sich zur Mitte eines Absatzes irgendwo im Text und geben Sie *J* (Gross!) ein. Sie werden sehen, dass dieses Kommando vi anweist, die nächste Zeile an das Ende der aktuellen Zeile anzuhängen. Nun bewegen Sie den Cursor über ein Zeichen und geben *r* ein und dann ein neues Zeichen; Sie werden sehen, dass das alte Zeichen gegen das neu eingegebene ersetzt wurde. Zuletzt bewegen Sie den Cursor zu irgendeiner Zeile im Text und geben Sie *dd* ein. Sie werden sehen, daß *dd* die aktuelle Textzeile löscht.

6.3 Wiederholen und löschen

Sie können jedes Bearbeitungskommando wiederholen, indem Sie . drücken. Wenn Sie damit experimentieren werden Sie merken, dass *dd...* vier Zeilen löschen wird und *J.....* vier Zeilen aneinanderhängen wird. Wie erwartet, bietet vi eine weitere Abkürzung.

Um Text zu löschen können Sie auch das *d* Kommando kombiniert mit Bewegungskommando benutzen. Zum Beispiel wird *dw* alles von der aktuellen Position bis zum Anfang des nächsten Wortes löschen; *d)* wird alles bis zum Beginn des nächsten Satzes löschen und *d}* wird den kompletten Rest des aktuellen Absatzes löschen. Experimentieren Sie mit dem *d* Kommando und den anderen Bearbeitungskommandos bis Sie sich mit ihnen vertraut fühlen.

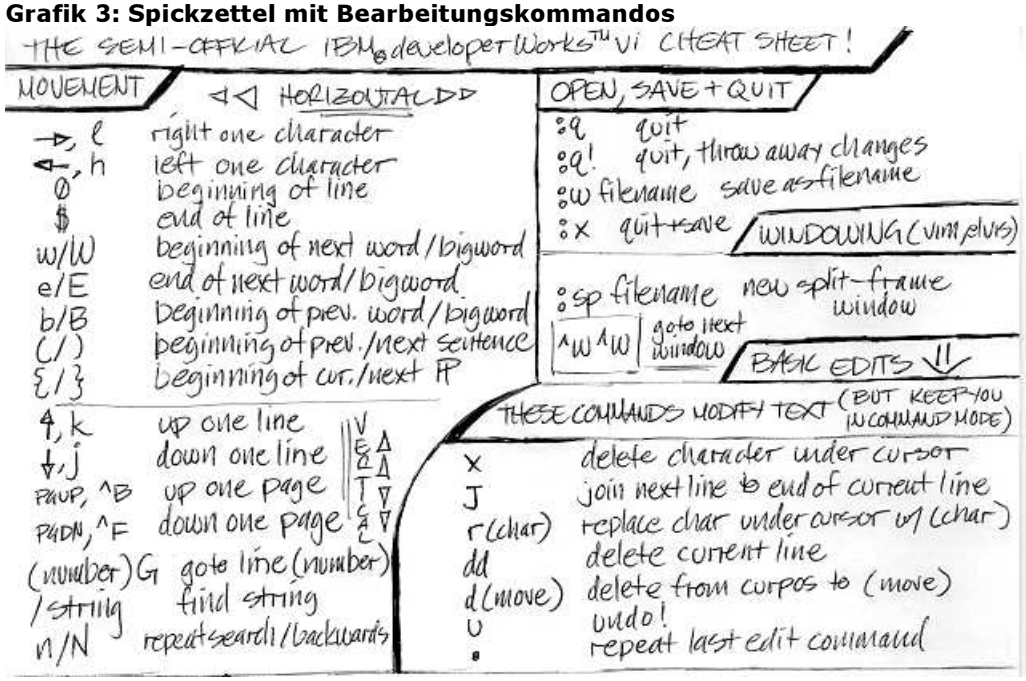
6.4 Rückgängig!

Nun, da wir mit Löschen experimentieren ist es sicherlich genau der richtige Zeitpunkt, auch zu lernen, wie man seine Änderungen wieder rückgängig macht. Durch das Drücken von *u* erlaubte die Originalversion von vi nur den letzten Bearbeitungsschritt wieder rückgängig zu machen. Trotzdem erlauben moderne vi Versionen wie vim mehrere Änderungen durch wiederholtes drücken von *u* auch weitere Änderungen an Ihrer Datei wieder rückgängig zu machen. Versuchen Sie einige der *d* und *u* Kommandos miteinander zu kombinieren.

6.5 Aktualisieren des Spickzettels

Es wird Zeit unseren Spickzettel zu akualisieren! Nachdem Sie alle Kommandos hinzugefügt haben, die wir bisher behandelten, sollte der Spickzettel in etwa so aussehen:

Grafik 3: Spickzettel mit Bearbeitungskommandos



7. Einfügemodus

7.1 Einfügemodus

Bisher haben wir behandelt, wie man sich mittels vi in Texten bewegt, Datei Ein- und Ausgabe realisiert und ein paar grundlegende Bearbeitungskommandos durchführt. Aber ich habe Ihnen bisher immer nicht gezeigt, wie Sie freie Texteingaben machen können. Dies war Absicht, denn vi's Einfügemodus (insert mode) ist zuerst ein bisschen kompliziert. Nachdem Sie sich mit dem Einfügemodus vertraut gemacht haben, wird Ihnen die Komplexität (und Flexibilität) unverzichtbar erscheinen.

Im vi Einfügemodus haben Sie die Möglichkeit, Text direkt am Bildschirm einzugeben - ganz so wie es auch in vielen anderen Editoren und Textverarbeitungen der Fall ist. Sobald Sie Ihre Änderungen gemacht haben, können Sie Escape drücken, um in den Kommandomodus zurückzukehren. Sie kommen in den Einfügemodus, indem Sie *i* oder *a* eingeben. Wenn Sie *i* eingeben, dann wird Ihr Text vor dem aktuellen Zeichen **eingefügt** werden. Wenn Sie *a* eingeben, dann wird Ihr Text hinter dem aktuellen Zeichen**angehängt** werden. Beachten Sie bitte: nachdem Sie Ihren Text eingegeben haben drücken Sie bitte *<ESC>* um in den Kommandomodus zurückzukehren.

7.2 Vorteile des Einfügemodus

Machen Sie weiter und versuchen Sie, die *a* und *i* Kommandos zu benutzen. Drücken Sie entweder *a* oder *i*, geben Sie ein wenig Text ein und dann drücken Sie Escape, um wieder in den Kommandomodus zurückzukehren. Nachdem Sie *a* oder *i* gedrückt haben, versuchen Sie *<ENTER>* einzugeben und passen Sie auf, was passiert. Durch die Benutzung der Pfeiltasten und *<DEL>*-Taste können Sie beachtliche Bearbeitungsschritte machen, ohne ständig in den Eingabemodus und wieder heraus zu springen.

7.3 Einfügeoptionen

Hier kommen noch ein paar andere praktische Möglichkeiten, um in den Einfügemodus zu gelangen. Drücken Sie *A* (Gross!) um - unabhängig von der aktuellen Position in der Zeile - mit dem Anhängen am **Ende** der Zeile zu beginnen. Sie können auch *I* (Gross!) drücken, um den Text am **Anfang** der Zeile einzufügen. Oder *o* um eine neue Leerzeile unter der aktuellen Zeile einzufügen, in die Sie dann Text eingeben können, und *O* (Gross!) um eine neue Zeile über der aktuellen einzufügen. Um die aktuelle Zeile mit einer neuen Zeile zu überschreiben geben Sie bitte *cc* als Kommando. Um innerhalb der Zeille Alles von der aktuellen Position bis zum Ende hin zu löschen können Sie *c$* eingeben. Um umgekehrt alles von der aktuellen Position bis zum Anfang der Zeile zu löschen, geben Sie das Kommando *c0* ein.

Zusätzlich dazu, dass jedes dieser Kommandos eine spezielle Operation durchführt, bringen sie Sie in den Einfügemodus. Nachdem Sie Ihren Text eingegeben haben, drücken Sie *<ESC>* um in den Kommandomodus zurückzukehren.

7.4 Textänderungen

Wir haben das *c* Kommando bisher benutzt, wenn wir *cc*, *c0* und *c$* angewendet haben. *cc* ist eine spezielle Form des Änderungskommandos, ähnlich zu *dd*. Die *c0* und *c$* Kommandos sind Beispiele der Anwendung des Änderungskommandos in Verbindung mit einem Bewegungskommandos. In dieser Form arbeitet *c* gleichwertig zu *d*, mit Außnahme darin bestehend, dass es Sie im Einfügemodus lässt, sodass Sie ersetzenden Text für die gelöschte Region eingeben können. Versuchen Sie einige der Bewegungskommandos mit *c* zu kombinieren und probieren Sie sie mit Ihrer Datei aus (Tipp : *cW*, *ce*, *c(*).

8. Verbund-Kommandos

8.1 Verbund-Kommandos

vi wird **wirklich** sehr mächtig, wenn Sie damit beginnen Verbund-Kommandos (also Kombinationen) wie z.B. *d{* and *cw* zu benutzen. Zusätzlich zu diesen Kommandos können Sie eine Zahl mit einem Bewegungskommando kombinieren, wie zum Beispiel *3w*, welches vi anweist drei Wörter nach rechts zu springen. Hier ein paar Beispiele zu "Bewegungskombinationen": *12b*, *4j*.

Zusätzlich zu (Zahl)(Bewegunskommando)-Kombinationen erlaubt vi auch *d* oder *c* mit einer Zahl oder einem Bewegungskommando zu kombinieren. So wird zum Beispiel *d3w* die nächsten drei Wörter löschen und *d2j* wird die aktuelle und die nächsten beiden Zeilen löschen. Probieren Sie einige *d* und *c*

Kombinationen aus, um ein Gefühl dafür zu bekommen, wie mächtig und schnell das Editieren mit vi sein kann. Sobald Ihnen diese Kommandos normal erscheinen, werden Sie die Fähigkeit besitzen Dateien in unglaublicher Geschwindigkeit zu bearbeiten.

8.2 Aktualisieren des Spickzettels

Es ist wieder einmal Zeit, den Spickzettel zu aktualisieren. Hier sehen Sie, wie er in etwa nun aussehen sollte:

Grafik 4: Spickzettel mit Verbundkommandos

8.3 Produktivitätssteigernde Fähigkeiten

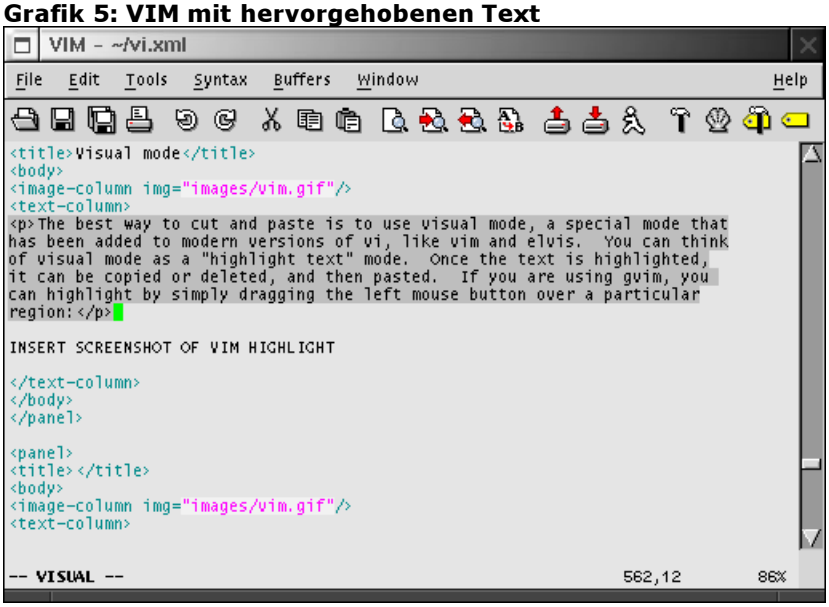
Bisher haben wir behandelt, wie man Bewegungen durchführt, Speichert und Beendet, einfache Änderungen und Löschungen realisiert, und wie man den Einfügemodus benutzt. Mit den Sachen die Sie bisher auf Ihrem Spickzettel haben, sollten Sie schon jegliche Aufgabe erledigen können.

Trotz Alledem: vi hat viele weitere mächtige Kommandos. In diesem Abschnitt werden Sie lernen, wie man **Ausschneidet**, **Kopiert** und **Einfügt**, **Sucht** und **Ersetzt** und **Autoindent** (automatisches Einrücken) benutzt. Diese Kommandos tragen dazu bei, vi spassiger und produktiver zu machen.

8.4 Visual mode

Die beste Möglichkeit zum Ausschneiden und Einfügen ist der Visual Mode, ein spezieller Modus der den modernen Versionen von vi (wie vim und elvis) hinzugefügt wurde. Sie können sich den Visual Mode als einen Modus mit Texthervorhebung vorstellen. Sobald der Text hervorgehoben ist, kann er kopiert oder gelöscht und dann eingefügt werden. Wenn Sie gvim benutzen, dann können Sie Text hervorheben, indem Sie einfach mit der linken Maustaste gedrückt über einen beliebigen Bereich fahren.

Grafik 5: VIM mit hervorgehobenen Text



Desweiteren können Sie auch in den Visual-Mode gehen indem Sie *v* drücken (dies wird für Sie auch die einzige Möglichkeit darstellen, wenn Sie vi aus der Konsole benutzen). Dann können Sie den Cursor bewegen, indem Sie Bewegungskommandos benutzen (normalerweise die Pfeiltasten) und damit einen Textbereich markieren. Sobald er einmal hervorgehoben ist, können wir den Text ausschneiden oder kopieren.

Wenn Sie Text kopieren, dann drücken Sie *y* (das für "yank", also "rausreissen" steht). Wenn Sie Text ausschneiden wollen, dann drücken Sie *d*. Sie werden dann zurück in den Kommandomodus gebracht. Nun bewegen Sie den Cursor zu der Position, an der Sie den ausgeschnittenen oder kopierten Text einfügen wollen und drücken Sie *P* um den Text nach dem Cursor einzufügen, oder drücken Sie *p* um den Text vor dem Cursor einzufügen. Fertig - der Ausschneide-/Kopiervorgang ist schon erledigt! Versuchen Sie ein paar Kopier-, Ausschneide- und Einfügeoperationen bevor Sie zum nächsten Abschnitt fortfahren.

8.5 Replacing text

Um Textmuster zu ersetzen, benutzen wir den **ex**-Modus. Wenn Sie das erste Muster auf der aktuellen Zeile ersetzen möchten, geben Sie *:s/<regexp>/<replacement>/* ein und drücken Sie *<ENTER>*, wobei <regexp> das Muster ist, dass gesucht und ersetzt werden soll und <replacement> das Ersetzmuster darstellt. Um alle entsprechungen auf der aktuellen Zeile zu ersetzen, geben Sie *:s/<regexp>/<replacement>/g* ein. Um jedes Vorkommen innerhalb der Datei zu ersetzen (das will man normalweise), geben Sie *:%s/<regexp>/<replacement>/g* ein. Wenn Sie global alles ersetzen möchten, aber vi vor jeder Ersetzung nachfragen soll, dann geben Sie *:%s/<regexp>/<replacement>/gc* (steht für "bestätigen") ein und drücken Sie *<ENTER>*.

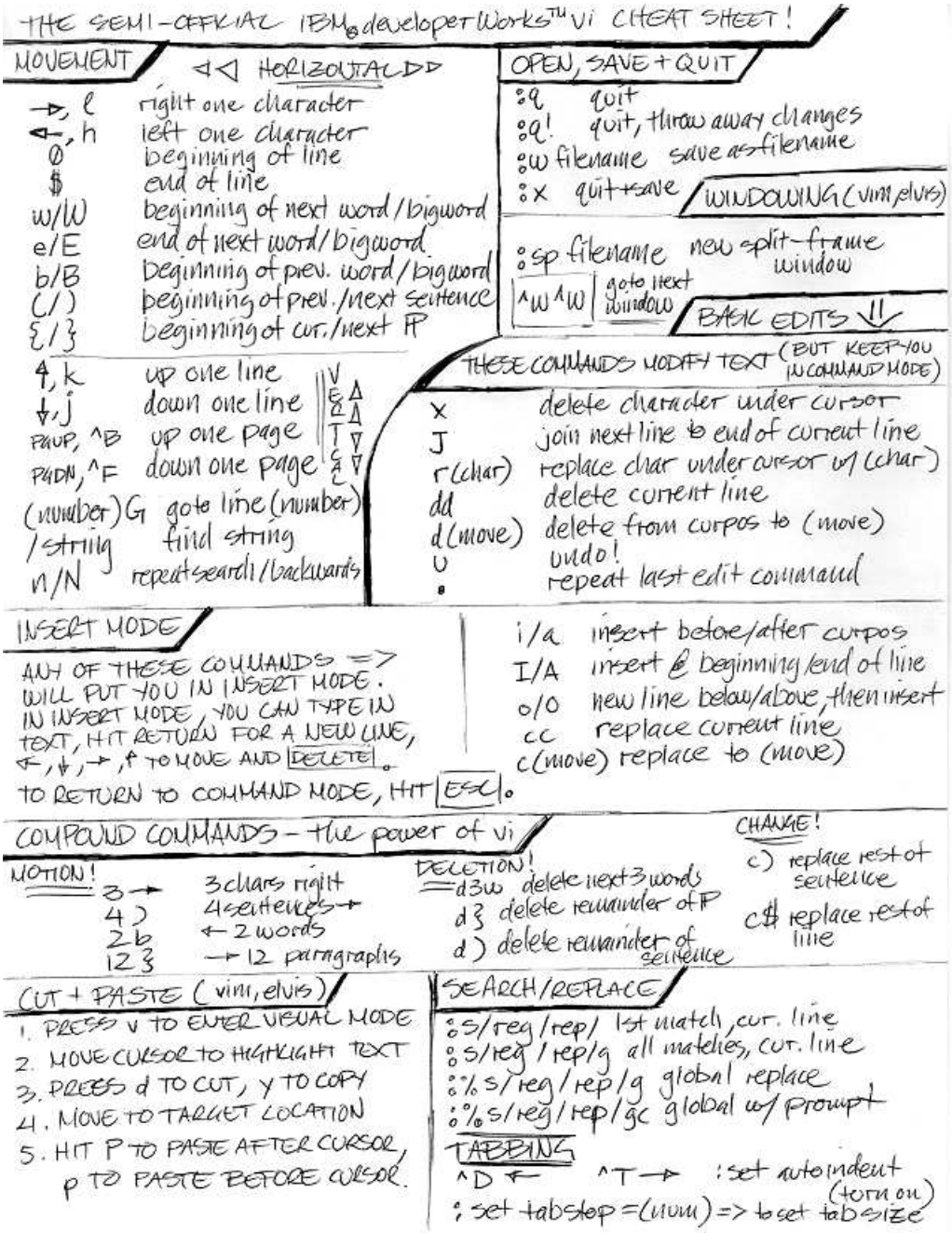
8.6 Indentation

vi beherrscht Automatisches Einrücken, welches als Hilfe gedacht ist, wenn Sie Quellcode editieren. Die meisten modernen Versionen von vi (wie vim) werden diesen Modus automatisch aktivieren, wenn Sie eine Quellcodedatei bearbeiten (wie z.B. eine .c-Datei). Wenn Automatisches Einrücken aktiviert ist, dann können Sie mittels *<Strg>d* die Einrückung eine Stufe nach links bewegen, und mit *<Strg>t* eine Stufe nach rechts. Wenn Automatisches Einrücken nicht automatisch aktiviert wurde, dann können Sie es manuell anschalten indem Sie *:set autoindent* im **ex** Modus eingeben. Sie können vi auch Ihre bevorzugte Tabulatorweite zu benutzen; dies geht mittels des *:set tabstop* Kommandos. *:set tabstop=4* ist recht beliebt.

8.7 Der letzte Spickzettel

Nun gut - dies ist das Ende des Tutorials! Nachdem Sie nun all die fortgeschrittenen Editierkommandos auf dem Spickzettel notiert haben, sollte er in etwa so aussehen:

Grafik 6: Letzter Spickzettel



Haben Sie Ihren Spickzettel immer bei der Hand und fangen Sie an mit vi Dateien zu editieren und E-Mails zu verfassen. Schauen Sie auf den Spickzettel wenn nötig: Innerhalb einer Woche werden Sie feststellen, dass Sie fast alle Kommandos behalten haben und Ihre Produktivität mit vi wird durch die Decke platzen! ;-)

8.8 Ressourcen

Hier sind ein paar Ressourcen, die Ihnen sicherlich weiterhelfen, wenn Sie mehr über vi lernen wollen:

- Die "vi Lovers Home Page", Eine hervorragende Ressource für alles rund um vi.
- Die "vim Homepage" ist der Ort für Infos rund um vim.
- Wenn Sie nach einem guten, altmodischen Buch suchen, dann ist "Learning the vi Editor", 6. Ausgabe sicherlich das Richtige für Sie. Viel Inhalt über vi und vi-Klone.

8.9 Über dieses Dokument

Die ursprüngliche Version dieses Artikels wurde zuerst bei IBM developerWorks veröffentlicht und ist Eigentum von Westtech Information Services. Dieses Dokument ist eine erneuerte Version des Original-Artikels und enthält verschiedene Verbesserungen durch das Gentoo Linux Documentation Team.


W3C
HTML
4.01