



chern. Nun sind Sie in der Lage die wahrscheinlich weltweit verbreitetsten Texteditoren zu bedienen: Vim und seinen Vater Vi. Neben diesen Grundfunktionen bietet Vim aber noch eine lange Liste weiterer Befehle, von denen die wichtigsten in Tabelle 1 aufgeführt sind. Eine vollständige Übersicht erhalten Sie durch aufrufen der Manual-Seiten (siehe Unix 3 in CLB Heft 1/2004) mit `man vim`.

### Textdateien betrachten

Ebenso wichtig wie das Erstellen von Textdateien ist das Betrachten von Textdateien. Die primitivste Möglichkeit bietet der Befehl `cat` (*concatenate*, zusammenhängen), der eigentlich den Inhalt zweier Dateien zusammenfügt. Mit `cat dateiname.txt` wird der Inhalt der Datei namens *dateiname.txt* auf den Bildschirm ausgegeben. Dieser Befehl ist eher ungeeignet bei langen Texten. Natürlich können Sie auch Vim verwenden um Texte zu lesen. Komfortabler ist dagegen `less`. Rufen Sie `less` zusammen mit dem Namen der zu betrachtenden Datei auf: `less dateiname.txt`. Mit den Pfeil- und `PgUp` bzw. `PgDn` Tasten können Sie nun durch die Textdatei scrollen. Durch Drücken der Taste `Q` beenden Sie `less` und gelangen wieder zurück in die Kommandozeile. `less` verwendet übrigens ähnliche Befehle wie Vim. So können Sie mit `/text` nach dem Vorkommen von *text* suchen. Sie sehen, die Anwendung von `less` ist extrem einfach und die Kenntnis der Vim Befehle ist universell einsetzbar. Manchmal möchte man aber nur den Anfang oder das Ende einer Datei betrachten. Auch hierfür gibt es eigene Befehle. Mit `head dateiname.txt` werden die ersten 10 Zeilen der Textdatei namens *dateiname.txt* angezeigt. Sollen mehr oder weniger Zeilen angezeigt werden, so lässt sich dies mit der Option `-n` beziffern. So zeigt `head -n5 dateiname.txt` nur die ersten 5 Zeilen an. Ebenso werden mit dem Befehl `tail dateiname.txt` nur die letzten 10 Zeilen der Datei *dateiname.txt* angezeigt. Auch hier kann die Anzahl der anzuzeigenden Zeilen mit der Option `-n` gesteuert werden.

### Textdateiwerkzeuge

Die große Stärke von Unix und seinen Derivaten liegt in einer Vielzahl kleiner Programme die einem das Leben versüßen (zumindest vereinfachen). In Bezug auf Textdateien haben Sie soeben `less`, `head` und `tail` kennen gelernt. Diese Befehle dienen lediglich dem Betrachten von Textdateien. Mit dem Befehl `sort dateiname.txt` können Sie den Inhalt von Textdateien sortieren. Dies ist insbesondere dann interessant, wenn die Datei z.B. Zahlenkolonnen enthält. Enthält die Datei mehrere Zahlenkolonnen, die z.B. durch ein Leerzeichen oder Tabulator getrennt sind, dann kann man mit der Option `-k` die zu sortierende Datenspalte angeben. `sort -k2 dateiname.`

`txt` sortiert bspw. die Zeilen nach den Einträgen in der zweiten Spalte. Das folgende Beispiel in Terminal 1 soll dies verdeutlichen.

#### Terminal 1

```
01 $ cat zahlen.txt
02 1 2
03 3 9
04 2 4
05 5 0
06 $ sort zahlen.txt
07 1 2
08 2 4
09 3 9
10 5 0
11 $ sort -k2 zahlen.txt
12 5 0
13 1 2
14 2 4
15 3 9
16 $
```

Mit Vim wurde die Datei *zahlen.txt* mit zwei durch Leerzeichen getrennte Datenkolonnen erstellt. In Zeile 1 von Terminal 1 geben wir den Dateiinhalt mit dem Befehl `cat` auf den Bildschirm aus. In Zeile 6 verwenden wir `sort` um die Zeilen zu sortieren. In Zeile 11 von Terminal 1 weisen wir `sort` durch Angabe der Option `-k2` an, die Zeilen nach der zweiten Spalte zu sortieren. Das Ergebnis sehen Sie in den Zeilen 12 bis 15. Bei allen Manipulationen bleibt der Inhalt der Datei *dateiname.txt* unverändert. Die Ausgabe erfolgt ausschließlich auf den Bildschirm. Sie lässt sich aber in eine Datei umleiten (*redirect*). Dazu dient der Befehl `>`. Ja, dies ist kein Druckfehler. Der Befehl besteht tatsächlich aus einem einzelnen Zeichen. D.h., `sort dateiname.txt > sortiert.txt` sortiert den Inhalt der Datei *dateiname.txt* und speichert das Ergebnis in der Datei *sortiert.txt*. Durch die Verwendung von Umleitungen (*redirections*) können Sie die Ausgabe jedes Befehls in eine Datei umleiten – eine Riesenstärke von Unix. Ein weiteres nützliches Werkzeug ist der Befehl `wc` (*word count*). Mit `wc dateiname.txt` lässt sich Anzahl von Zeilen, Wörtern und Zeichen zählen. Wie in Terminal 2 gezeigt, erfolgt die Ausgabe in dieser Reihenfolge, gefolgt von dem Dateinamen.

#### Terminal 2

```
01 $ wc zahlen.txt
02 4      8      16 zahlen.txt
03 $
```

Die Datei *zahlen.txt* enthält aber doch nur 8 und nicht 16 Zeichen. Der Befehl `wc` zählt auch Leerzeichen, Tabulatoren und Zeilenumbrüche als Zeichen.

## Textsuche

Schließlich möchte ich Ihnen noch ein komfortables Textsuchprogramm vorstellen: **grep** (greifen). **grep** sucht in Textdateien nach bestimmten Textstellen. Die Syntax ist: **grep 'muster' dateiname**. Bei der Angabe des Dateinamens können Sie auch Wildcards verwenden, die Sie in Unix 3 (CLB Heft 1/2004) kennen gelernt haben. Zusätzlich bietet **grep** eine Vielzahl von Optionen an. Hier möchte ich nur die Option **-r** (*recursively*) erwähnen, die **grep** veranlasst Dateien im aktuellen und allen Unterverzeichnissen zu durchsuchen. **grep -r 'Baum' \*.txt** etwa durchsucht alle Dateien im aktuellen und allen Unterverzeichnissen die auf *.txt* enden nach Zeilen die das Wort Baum enthalten. Die zutreffenden Zeilen werden auf den Bildschirm ausgegeben. In Terminal 3 suchen wir die Zahl 2 in dem File *zahlen.txt* aus Terminal 1.

### Terminal 3

```
01 $ grep '2' zahlen.txt
02 1 2
03 2 4
04 $
```

Wenn wir in der kommenden Ausgabe über reguläre Ausdrücke (eine Erweiterung von Wildcards) sprechen, werden wir noch einmal auf **grep** zu sprechen kommen. Dann werden Sie in die Lage versetzt werden, nach komplizierten Mustern zu suchen.

h	Cursor nach links bewegen
l	Cursor nach recht bewegen
k	Cursor nach oben bewegen
j	Cursor nach unten bewegen
\$	Cursor an das Zeilenende bewegen
0	Cursor an den Zeilenanfang bewegen
G	Cursor an die letzte Zeile der Datei bewegen
nG	Cursor an die n-te Zeile der Datei bewegen
x	Lösche das aktuelle Zeichen
dd	Lösche die aktuelle Zeile
/text	Suche vorwärts nach text
?text	Suche rückwärts nach text
:s/text1/text2/gc	Ersetzte (substitute) text1 durch text2 im gesamten Dokument und frage vor jeder Ersetzung

### Neue, in dieser Ausgabe behandelte Befehle

cat	Ausgabe des Dateinhaltes auf den Bildschirm
grep	Sucht in Textdateien nach Wörtern
head	Zeigt die ersten 10 Zeilen einer Textdatei an
less	Tool für die Betrachtung großer Textdateien
sort	Sortiert den Inhalt einer Textdatei alphabetisch
tail	Zeigt die letzten 10 Zeilen einer Textdatei an
vi	Startet den Texteditor Vim
wc	Gibt die Anzahl von Zeilen, Wörtern und Zeichen einer Textdatei aus

Tabelle 1: Eine kleine Auswahl des Befehlsschatzes der Texteditoren Vim und Vi.

## Halten Sie ein Ökosystem mit Tieren und Pflanzen in Ihren Händen: Eine Glaskugel im ökologischen Gleichgewicht

**H**ellrote Garnelen, Mikroorganismen und Algen leben gemeinsam in Wasser mit Meerwasser-ähnlicher Salzkonzentration. Sie sind vollständig von Glas umschlossen; es findet kein Gas- oder anderer Stoffaustausch mit der Umwelt außerhalb des Glases statt! Triebfeder für das Leben im Glas ist einzig das eingestrahlte Licht.



Winzige Algen, zum Teil an getrockneten Gorgonien, erzeugen aus Kohlendioxid Sauerstoff. Dazu benötigen sie Lichtenergie. Die Garnelen atmen den Sauerstoff und fressen Algen sowie im Wasser vorhandene Bakterien. Diese wiederum formen die tierischen Abfallstoffe in Nährstoffe für die Algen um. Ebenso erzeugen Garnelen und Bakterien Kohlendioxid für die pflanzlichen Lebensformen...

In solch einer Ecosphere leben die Garnelen typischerweise zwei Jahre, können aber auch bis zu zehn Jahre alt werden. Dafür galt es beispielsweise, Garnelen zu finden, die sich nicht gegenseitig fressen. Auch jeder Besitzer einer Ecosphere muss das Gleichgewicht des Lebens im Auge behalten. So führt zuviel Licht zu starkem Algenwachstum – und darüber hinaus zu für die Garnelen unverträglichen pH-Werten im Wasser. Ebenso ist eine möglichst gleichmäßige Raumtemperatur nötig.

Wie Sie Ihre Ecosphere erhalten: Siehe letzte Seite dieser CLB!