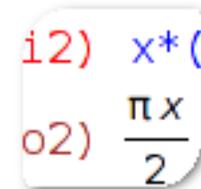


Topthemen dieser Ausgabe

Mathematik mit Open Source

Der Artikel ist eine Reise in die Welt der modernen Mathematik. Die wichtigsten Handwerkszeuge des Mathematikers waren, sind und werden immer sein: Bleistift und Papier. Doch mit der Erfindung des elektronischen Rechners entstanden zahlreiche leistungsfähige und ausgeklügelte Computerprogramme, die den Mathematiker und jeden, der sich dafür hält, bei der alltäglichen Arbeit unterstützen. Ein paar dieser Programme werden hier vorgestellt. Den Start der Reihe macht das Computer-Algebra-System Maxima. ([weiterlesen](#))

Seite 5



Spamfilterung mit bogofilter

Das Aussortieren von unerwünschten E-Mails (Spam) gehört zu den notwendigen und unbeliebten Aufgaben eines Internetnutzers. Normalerweise bietet ein E-Mail-Provider Spam-Filterung an. Eine Alternative oder auch Ergänzung dazu ist es, selber ein Filterprogramm wie beispielsweise bogofilter zu verwenden. ([weiterlesen](#))

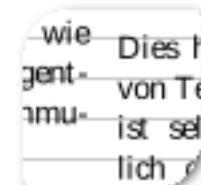
Seite 9



Registerhaltiger Satz mit L^AT_EX

Wer eine aktuelle Zeitschrift auf dem Markt aufschlägt, dem wird wahrscheinlich auffallen, dass bei einem mehrspaltigen Layout benachbarte Zeilen immer auf einer Höhe liegen. Diese Eigenschaft nennt man Registerhaltigkeit. L^AT_EX bietet von sich aus keine Möglichkeit, Texte registerhaltig zu setzen. Der Artikel soll zeigen, wie man dies dennoch mit etwas Handarbeit erhalten kann. ([weiterlesen](#))

Seite 15



Editorial

Neues Team-Mitglied

In der Juni-Ausgabe von **freiesMagazin** [1] hatten wir einen Aufruf veröffentlicht, um Unterstützung für unser Layout-Team zu finden. Wir freuen uns, dass unser Aufruf erfolgreich war und begrüßen Kai Welke ganz herzlich in unserem Team. Wir wünschen Kai viel Spaß und freuen uns auf eine gute Zusammenarbeit!

Rezensenten und Autoren gesucht

Seit Jahren veröffentlicht **freiesMagazin** regelmäßig Rezensionen von Büchern aus dem IT-Bereich. Von den Buch-Verlagen erreichen uns viele Anfragen zum Verfassen von Rezensionen. Meistens müssen wir Anfragen leider ablehnen, da sich kein Autor in unseren eigenen Reihen hierfür finden lässt.

Daher hier unser Aufruf: Wer gerne liest und dazu auch noch schreiben kann, kann sich als Rezensent unter redaktion@freiesMagazin.de bei der Redaktion bewerben. Eine Voraussetzung ist, dass Sie uns eine kleine Kostprobe einer Rezension mitschicken. Danach sollte der Zusendung von Rezensionsexemplaren je nach Fachgebiet und Interessen nichts mehr im Wege stehen. Wir freuen uns über Zuschriften und Anfragen.

Seit unserem erfolgreichen Aufruf im Januar ist unser Vorrat an Artikeln für die nächsten Ausgaben langsam erschöpft. Wer also kein Interesse an Rezensionen hat, aber trotzdem einen Arti-

kel zu **freiesMagazin** beisteuern möchte, findet vielleicht auf unserer Artikelwunschliste [2] sein Fachgebiet wieder und kann einen Artikel dazu schreiben. Auch andere Themen rund um Freie Software sind herzlich willkommen.

Für Interessenten finden sich in der Rubrik Mitmachen [3] auf unserer Homepage und in den Autorenrichtlinien zahlreiche nützliche Hinweise und Hilfestellungen. Für weitere Fragen steht die Redaktion unter redaktion@freiesMagazin.de gerne zur Verfügung.

Neuer Programmierwettbewerb

Und in der Hoffnung, dass wir das Sommerloch sinnvoll nutzen können, versuchen wir ab diesem Oktober auch wieder einen Programmierwettbewerb anzubieten. Im letzten Jahr hat dies aus Zeitgründen nicht geklappt, Themen haben wir aber schon gesammelt [4].

Und nun wünschen wir Ihnen viel Spaß mit der neuen Ausgabe!

Ihre **freiesMagazin**-Redaktion

LINKS

- [1] <http://www.freiesmagazin.de/freiesMagazin-2014-06>
- [2] <http://www.freiesmagazin.de/artikelwuensche>
- [3] <http://www.freiesmagazin.de/mitmachen>
- [4] <http://www.freiesmagazin.de/20130909-ergebnisse-der-abstimmung-fuer-programmierwettbewerb>

Inhalt

Linux allgemein

Der Juni im Kernelrückblick S. 3

Anleitungen

Mathematik mit Open Source S. 5

Spamfilterung mit bogofilter S. 9

Registerhaltiger Satz mit L^AT_EX S. 15

Software

Statische Webseiten mit Pelican erstellen S. 22

Typographie lernen mit Type:Rider S. 26

Planet Explorers S. 28

Retten von verlorenen Daten mit PhotoRec S. 32

Community

Rezension: Geheimnisse eines JavaScript-Ninjas S. 34

Rezension: Inkscape – Der Weg zur professionellen Vektorgrafik S. 36

Magazin

Editorial S. 2

Leserbriefe S. 38

Veranstaltungen S. 40

Konventionen S. 40

Impressum S. 41



Teilen



Kommentieren

Der Juni im Kernelrückblick von Mathias Menzer

Basis aller Distributionen ist der Linux-Kernel, der fortwährend weiterentwickelt wird. Welche Geräte in einem halben Jahr unterstützt werden und welche Funktionen neu hinzukommen, erfährt man, wenn man den aktuellen Entwickler-Kernel im Auge behält.

Linux-3.15-Entwicklung

Eigentlich hatte Torvalds keinen -rc8 [1] vorgesehen, aber letzte Korrekturen für **dcache**, ein System zur Zwischenspeicherung der zuletzt gesuchten Verzeichniseinträge in Dateisystemen, legten einen letzten Zwischenschritt nahe. Und so gab es die Veröffentlichung von Linux 3.15 erst am zweiten Juni-Wochenende [2]. Bei der Freigabe merkte Torvalds allerdings an, dass die Änderungen an **dcache** scheinbar niemanden berührt hatten. Dafür traten jedoch andere Probleme während dieser zusätzlichen Woche auf und trugen damit immerhin zu knapp über 100 zusätzlichen Änderungen bei, die zumeist Fehlerkorrekturen geschuldet waren.

Linux 3.15

Der jüngste Spross aus der Reihe der Linux-Kernel konnte die 70 Tage seines Vorgängers knapp unterbieten, stellt dabei aber trotzdem ein Schwergewicht dar. Fast 15000 Änderungen wurden während der Entwicklungsphase eingebracht und auch die Menge der veränderten Quelltextzeilen liegt weit über dem Durchschnitt der ver-

gangenen Kernel-Versionen, wenn auch nicht auf dem Spitzenplatz. Der Qualität der eingebrachten neuen und verbesserten Funktionen tat dies jedoch keinen Abbruch.

Viele Anwender werden sich schon einmal einmal daran gestört haben, dass das Aufwachen des Rechners aus dem Suspend-Modus mitunter recht lange dauert, wenn klassische Festplatten im Spiel sind. Dies liegt in erster Linie daran, dass das System auf den Treiber für die Festplatten warten muss, bis diese vollständig betriebsbereit sind und z. B. ihre Magnetscheiben auf Drehzahl gebracht haben. Intel-Entwickler haben dieses Problem nun entschärft, indem dieser Treiber nun den Weckruf an die Festplatten absendet und dann nicht weiter blockiert. Weitere Kommandos an die Hardware werden gegebenenfalls zwischengespeichert und ausgeführt, sobald sie verfügbar ist.

Das FUSE-Modul [3] erlaubt das Einbinden von Dateisystemen aus dem Kontext des Anwenders heraus und genießt nicht immer den Ruf, besonders performant zu sein. Dem versucht man nun ein wenig abzuwehren, indem Writeback-Caching für FUSE aktiviert wird, wenn der Nutzer dies als Mount-Option setzt. Hierbei werden Änderungen nicht sofort in das Dateisystem geschrieben, sondern zwischengespeichert, um mehrere Änderungen in einem Rutsch schreiben zu können oder andere Operationen nicht zu blockieren.

Das 20 Jahre alte Dateisystem XFS [4] genießt auch heute noch einen guten Ruf. Ursprünglich für die von SGI vertriebene Unix-Version Irix [5] entwickelt, wurde es 2000 unter die GPL gestellt und fand dann schnell seinen Weg in den Linux-Kernel. Nutzer schätzen insbesondere die Stabilität, Datensicherheit und Leistungsfähigkeit von XFS, die erst von modernen Dateisystemen wie beispielsweise Ext4 übertroffen wurde. Unter Linux 3.15 wird nun ein neues Format für XFS als stabil gekennzeichnet, das bereits in Linux 3.10 aufgenommen worden war [6]. Damit sollen nun nicht nur die abgelegten Daten, sondern auch ihre Metadaten mittels Prüfsummen gegen Verfälschungen geschützt werden. Auch die Funktionen für das Wiederherstellen eines lauffähigen Dateisystems nach einem Ausfall wurden verbessert.

XFS wurde auch mit einer weiteren Funktion bedacht, die das Löschen von Teilen aus einer Datei betrifft. So stehen nun der Funktion **fallocate** zwei neue Methoden zur Manipulation von Dateien zur Verfügung. Die erste dient dazu, um einen zusammenhängenden Bereich einer Datei zu entfernen, ohne Löcher im genutzten Speicherbereich zu hinterlassen. Die zweite setzt einen zusammenhängenden Bereich innerhalb einer Datei auf Null. Diese Methoden sollen schneller arbeiten als die direkte, manuelle Bearbeitung der Dateien. Neben XFS stehen die beiden neuen Flags für **fallocate** auch unter ext4 [7] zur Verfügung, weitere Dateisysteme sollen bald folgen.

Ebenfalls auf Dateien bezogen ist der Systemaufruf `renameat2()`: Er ermöglicht sozusagen das Vertauschen der Namen zweier Dateien. Bislang war hierfür ein Zwischenschritt erforderlich, bei dem eine Datei zeitweise einen Platzhalter-Namen annehmen musste. Der neue Systemaufruf ermöglicht nun beispielsweise, Dateien oder ganze Verzeichnisbäume mit nur einem Systemaufruf durch einen symbolischen Link zu ersetzen, ohne dass dabei Wartezeiten berücksichtigt werden müssen.

Bei der Verwaltung von Arbeits- und Auslagerungsspeicher liegt die Kunst darin herauszufinden, welche Speicherseiten denn nun wirklich benötigt werden und welche ausgelagert werden können. Hierfür pflegt die Speicherverwaltung jeweils eine Liste für inaktive und aktive Daten. Bislang gab es hier die Restriktion, dass die Aktiv-Liste nicht die Größe der Inaktiv-Liste überschreiten darf, was jedoch teilweise zu Problemen führte. Der Entwickler Johannes Weiner bricht diese Beschränkung nun auf und bestimmt die jeweils passende Größe für Inaktiv- und Aktiv-Liste anhand der Zeit, die vergeht, bis eine Speicherseite wieder in den Hauptspeicher zurückgeholt wird. Hierfür waren einige Umbauarbeiten an der Speicherverwaltung notwendig, da diese Informationen bislang nicht protokolliert wurden.

Dies ist nur ein kleiner Überblick über einige der interessanteren Neuerungen. Eine vollständige Liste der Änderungen findet sich auf der englischsprachigen Seite kernelnewbies.org [8].

Entwicklungsversionen von Linux 3.16

Das Merge Window eröffnete Torvalds diesmal bereits mit der Freigabe von Linux 3.15-rc8. Der Grund hierfür lag darin, dass er während seines Urlaubs keine Merge Requests abarbeiten wollte und so den Zwei-Wochen-Zeitraum einfach um eine Woche nach vorne verlegte. Somit lag Linux 3.16-rc1 [9] bereits eine Woche nach der Veröffentlichung von Linux 3.15 vor. Verglichen mit den vorangegangenen Versionen fiel diese erste Entwicklerversion dann auch recht kompakt aus. Die Zahl der Änderungen liegt mit über 12000 Commits zwar im oberen Bereich, doch handelte es sich überwiegend um kleine Änderungen, sodass sich das Gesamtvolumen der geänderten Quelltestzeilen deutlich unter Durchschnitt bewegt.

Gearbeitet wurde zu einem guten Teil im Bereich der Grafik-Treiber, insbesondere an AMDs Radeon-Treiber und dem Nouveau-Treiber für NVIDIA-Grafik-Kerne. Auch die cgroups – Prozess-Gruppen, die der Verwaltung von System-Ressourcen dienen – wurden überarbeitet. Linux 3.16-rc2 [10] konnte besonders viele Änderungen an der SPARC-Architektur [11] aufweisen, die jedoch fast ausschließlich der Beseitigung von Warnmeldungen bei der Kernel-Kompilierung geschuldet waren. Änderungen an der Firmware des Nouveau-Treibers stechen ebenfalls aus den ansonsten eher gleichmäßig verteilten Änderungen heraus. Die dritte Entwicklerversion [12] erweckt bereits den Eindruck, dass sich die Entwicklung beruhigt. Die Änderungen haben zwar ihren Schwerpunkt im Architektur-

Bereich, insbesondere MIPS und PowerPC. Bei der x86-Architektur stechen Korrekturen an der VDSO-Infrastruktur hervor, die es Anwendungen im User-Kontext erlaubt, Funktionen aus dem Kernspace zu nutzen.

LINKS

- [1] <https://lkml.org/lkml/2014/6/1/264> 
- [2] <https://lkml.org/lkml/2014/6/8/70> 
- [3] https://de.wikipedia.org/wiki/Filesystem_in_Userspace
- [4] [https://de.wikipedia.org/wiki/XFS_\(Dateisystem\)](https://de.wikipedia.org/wiki/XFS_(Dateisystem))
- [5] <https://de.wikipedia.org/wiki/IRIX>
- [6] <http://www.pro-linux.de/-0h21526c>
- [7] <https://de.wikipedia.org/wiki/Ext4>
- [8] http://kernelnewbies.org/Linux_3.15 
- [9] <https://lkml.org/lkml/2014/6/16/1> 
- [10] <https://lkml.org/lkml/2014/6/22/5> 
- [11] https://de.wikipedia.org/wiki/Oracle_SPARC
- [12] <https://lkml.org/lkml/2014/6/29/126> 

Autoreninformation

Mathias Menzer ([Webseite](#)) behält die Entwicklung des Linux-Kernels im Blick, um über kommende Funktionen von Linux auf dem laufenden zu bleiben und immer mit interessanten Abkürzungen und komplizierten Begriffen dienen zu können.

 [Teilen](#)

 [Kommentieren](#)

Mathematik mit Open Source von Tobias Kaulfuß

Dieser Artikel ist eine Reise in die Welt der modernen Mathematik. Die wichtigsten Handwerkszeuge des Mathematikers waren, sind und werden immer sein: Bleistift und Papier. Doch mit der Erfindung des elektronischen Rechners entstanden zahlreiche leistungsfähige und ausgeklügelte Computerprogramme, die den Mathematiker und jeden, der sich dafür hält, bei der alltäglichen Arbeit unterstützen. Ein paar dieser Programme werden hier vorgestellt.

Jenseits des Taschenrechners

Ja, es gibt tatsächlich mehr als den obligatorischen Taschenrechner, dessen „Wissenschaftlicher Modus“ schon einen Teil der geneigten Leserschaft in höchste Glücksgefühle versetzen mag: „Was für ein Nerd man doch sei.“ Ein anderer Teil wird wohl an dunkle, leider noch nicht ganz verdrängte Stunden der Schulzeit erinnert. Doch gerade letztere werden gebeten weiter zu lesen. Es gibt auch keine Noten und niemand muss etwas an der Tafel vorrechnen, versprochen.

Die wohl bekanntesten kommerziellen Softwarepakete für Mathematik sind Mathematica, Maple und Matlab.

Mathematica [1] wurde von dem genialen Stephen Wolfram [2] entwickelt, der bereits im Alter von 15 Jahren Artikel zur Teilchenphysik veröffentlichte. Er betreibt auch die „Suchmaschine“ Wolfram

Alpha [3]. Mathematica ist ein Computeralgebrasystem (CAS) [4], das heißt, es kann nicht nur Zahlen verarbeiten, sondern auch Symbole, Buchstaben, Variablen, Funktionen, Matrizen und noch vieles mehr.

Maple [5] ist ebenfalls ein CAS. Entwickelt wurde es in Kanada. Die Kanadier scheinen nicht viele Symbole zu kennen, jedenfalls haben sie kurzerhand ihr Nationalsymbol [6] als Name und Logo für ihre Mathematiksoftware gewählt. Schwer zu sagen, ob ein Mangel an Symbolen der Grund dafür ist, dass Maple preiswerter ist als Mathematica. Der Preis könnte jedoch ein Grund dafür sein, dass Maple etwas häufiger an Universitäten verwendet wird, während Mathematica häufiger im professionellen Umfeld zu finden ist.

Einen etwas anderen Einsatzbereich hat Matlab [7]. Hier stehen wieder Zahlen und Matrizen im Vordergrund. Es dient der numerischen Simulation, Datenerfassung und -analyse. Die Grenzen zwischen den verschiedenen Paketen sind jedoch fließend und viele Probleme lassen sich in allen genannten Programmen lösen.

Back to the root

Okay, die Überschrift ist ein bisschen geschummelt. Normale Benutzerrechte reichen für die Arbeit. Aber wie ein echter Superuser wird zunächst einmal auf der Konsole gehackt.

Die Reise führt zurück in die 60er Jahre des letzten Jahrhunderts und das Open-Source-Programm Maxima [8] wird installiert. Maxima stammt von einem der allerersten Computeralgebrasystemen Macsyma ab.

Nach dem Start begrüßt der Input-Prompt:

```
$ maxima
Maxima 5.32.1 http://maxima.sourceforge.net
using Lisp SBCL 1.1.14
Distributed under the GNU Public License
. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
```

Einfache Rechnungen

Das erste Kommando soll sein: `1+2`; bestätigt wird mit `Enter`:

```
(%i1) 1+2;
(%o1) 3
(%i2)
```

Man beachte: Kommandos werden immer mit Semikolon beendet. Eingabezeilen erkennt man an dem kleinen `i`; Ausgabezeilen, also die Ergebnisse der Berechnungen, an dem kleinen `o`. Beide werden fortlaufend nummeriert.

Addition, Subtraktion und Multiplikation beherrscht Maxima aus dem Effeff. Auch Rechnen mit Klammern gehört zu den leichteren Übungen:

```
(%i2) 5-7;
(%o2)  - 2
(%i3) 7*6;
(%o3)  42
(%i4) 4*(3+2);
(%o4)  20
```

Die erste kleine Überraschung erlebt man, wenn man versucht zu dividieren:

```
(%i5) 10/2;
(%o5)  5
(%i6) 11/2;
(%o6)  --
      2
(%i7) 10/15;
(%o7)  -
      3
(%i8) 5*(3/2);
(%o8)  --
      2
(%i9) 5+(3/2);
(%o9)  --
      2
```

Maxima gibt einen Bruch aus, freundlicherweise soweit wie möglich gekürzt. Möchte man eine De-

zimalzahl ausgeben, verwendet man die Funktion **float()**. (Floating Point Number = Fließkommazahl).

```
(%i10) float(13/2);
(%o10)  6.5
(%i11) float(sqrt(2));
(%o11)  1.414213562373095
```

sqrt() berechnet die Quadratwurzel (Square Root) einer Zahl. Maxima kennt Konstanten wie die Kreiszahl Pi und Standardfunktionen wie Sinus und Kosinus.

```
(%i12) float(%pi);
(%o12)  3.141592653589793
(%i13) sin(%pi);
(%o13)  0
(%i14) cos(3*%pi);
(%o14)  - 1
(%i15) sin(float(%pi));
(%o15)  1.224646799147353e-16
```

Die Zeile **o15** zeigt, dass das Rechnen mit Dezimalzahlen mit Vorsicht zu genießen ist: $1.224646799147353e-16$ bedeutet $1.224646799147353 \cdot 10^{-16} = 0.000000000000000122466799147353$. Das ist zwar eine sehr kleine Zahl, jedoch nicht genau 0, was laut Zeile **o13** ein anderes Ergebnis ist (und es gibt gute Gründe anzunehmen, dass 0 richtig ist). Das Problem ist, dass Pi unendlich viele Nachkommastellen hat, der gewöhnliche PC aber nicht unendlich viel Arbeitsspeicher, um all diese Ziffern speichern zu können. Auch steht

nur selten unendlich viel Zeit zur Verfügung, um bei Bedarf alle Ziffern zu berechnen. Man muss sich also mit einem Kompromiss begnügen: Nachkommastellen werden irgendwann abgeschnitten, was zu ungenauen Ergebnissen führt.

Symbolische Gleichungen

Man darf sich nicht Algebrasystem nennen, wenn man nicht das Rechnen mit Buchstaben beherrscht:

```
(%i16) 2*a;
(%o16)  2 a
(%i17) (x^2+y)^3;
              2
(%o17)  3 (y + x )
(%i18) expand((a+b)^2);
              2      2
(%o18)  b + 2 a b + a
(%i19) expand((a+b)^10);
      10      9      2 8      3 7
(%o19) b + 10 a b + 45 a b + 120 a b
          4 6      5 5      6 4
      + 210 a b + 252 a b + 210 a b
          7 3      8 2      9      10
      + 120 a b + 45 a b + 10 a b + a
```

Die Funktion **expand()** multipliziert die Klammer aus, wie man leicht in Zeile 18 anhand der ersten Binomischen Formel sieht.

Faktorisierung

```
(%i20) u: 10!;
(%o20)  3628800
(%i21) factor(u);
```

```
(%o21)      8 4 2
(%i22) expand(%o21);
(%o22)      3628800
```

In Zeile **o20** passieren zwei Dinge: Die Fakultät von 10 wird berechnet, das heißt: $10! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot 9 \cdot 10 = 3628800$. Diese Zahl wird dem Buchstaben **u** zugewiesen und kann dann mit dieser Abkürzung verwendet werden. In Zeile **o21** wird die Zahl **u** = **3628800** faktorisiert, also in Primzahlen zerlegt, deren Produkt u ist. Es gilt, die Ausgabe richtig zu lesen. Hier steht: $3628800 = 2^8 \cdot 3^4 \cdot 5^2 \cdot 7$. Die Probe folgt in Zeile **o22**, wo das Ergebnis **o21** ausmultipliziert wird.

Es ist ein sehr wichtiges Resultat der Mathematik, dass sich jede natürliche Zahl eindeutig in ein Produkt von Primzahlen zerlegen lässt. Das heißt, man kann jede Zahl faktorisieren, und für jede Zahl tauchen immer die gleichen Primzahlen auf. (Wenn die Zahl selbst schon eine Primzahl ist, dann besteht die Faktorisierung natürlich nur aus einer Primzahl, nämlich der Zahl selbst.) (Ja, dieser letzte Satz strotzt vor bestechender Logik!)

In der Mathematik nennt man dies „Fundamentalsatz der Arithmetik“ oder „Hauptsatz der elementaren Zahlentheorie“. Diese martialischen Begriffe geben schon einen kleinen Hinweis auf die Bedeutung der Aussage. Ganz praktische Anwendung findet der Satz in den modernen Verschlüsselungsverfahren, die tagtäglich bei der Arbeit mit dem Computer genutzt werden, z. B. beim Online-

banking. Der RSA-Algorithmus [9] basiert darauf, dass es sehr leicht ist, zwei Zahlen zu multiplizieren, aber sehr schwer eine große Zahl in ihre Primfaktoren zu zerlegen. Und wenn hier „große Zahl“ steht, dann ist eine große Zahl gemeint, eine Zahl mit hunderten oder gar tausenden Ziffern.

Differentialrechnung

Zurück zu Maxima: Für das Rechnen mit Funktionen bietet das CAS vielfältige Möglichkeiten.

```
(%i23) f(x) := x^2+x-1;
(%o23)      2
          f(x) := x + x - 1
(%i24) f(2);
(%o24)      5
(%i25) solve(f(x)=11,x);
(%o25)      [x = 3, x = - 4]
(%i26) integrate(f(x),x);
(%o26)      3 2
          x  x
          -- + -- - x
          3 2
(%i27) diff(f(x),x);
(%o27)      2 x + 1
(%i28) integrate(f(x),x,1,2);
(%o28)      17
          --
          6
(%i29) 'integrate(f(x),x);
(%o29)      /
          [ 2
          I (x + x - 1) dx
          ]
          /
```

Zunächst wird die Funktion $f(x) = x^2 + x - 1$ definiert. Die Funktion wird an der Stelle $x=2$ ausgewertet, man erhält den Wert 5. Umgekehrt werden die x-Stellen gesucht, an denen **f** den Wert **11** annimmt, das ist für $t < x = 3 >$ und $x = -4$ der Fall. Die Funktion **integrate()** bildet eine Stammfunktion [10] nach der gewählten Variablen **x**, während **diff()** die Ableitung [11] bildet. Über gibt man **integrate()** zwei Integrationsgrenzen, dann wird das bestimmte Integral berechnet, was, wie jeder weiß, dem Flächeninhalt zwischen Funktionsgraphen und x-Achse entspricht. In Zeile **o29** sieht man, dass das einfache Anführungszeichen ' die Berechnung unterdrückt und nur den Ausdruck ausgibt. In diesem Fall mit einem wunderschön gezeichneten Integralzeichen.

Graphen zeichnen mit Gnuplot

Hat man Gnuplot [12] installiert, lassen sich mit Maxima Funktionsgraphen zeichnen. Das Bild kann sogar (mehr schlecht als recht) in der Konsole ausgegeben werden:

```
(%i30) plot2d(sin(x),[x,0,2*%pi],[↵
gnuplot_term,dumb]);
sin(x)
1 +-----+
| ++ ++ + + + + |
| + ++ |
| + + |
0.5 |++ + +|
|+ + |
|+ ++ |
|+ + |
0 |+.....++.....+|
```

```

|      +      + |
|      ++     + |
|      +      + |
-0.5 | +      +      ++ |
|      +      + |
|      ++     + |
| + + + ++ ++ + |
-1 +-----+
    0 1 2 3 4 5 6
      x
(%o30) /home/benutzername/maxout.gnuplot
/home/benutzername/maxplot.txt
    
```

Die Option `[x,0,2*%pi]` sorgt dafür, dass die Funktion nach der Variablen `x` im Intervall `[0, 2Pi]` gezeichnet wird. Für die Ausgabe auf der Konsole sorgt die Option `[gnuplot_term,dumb]`.

Es ist auch möglich eine Bilddatei zu erzeugen:

```

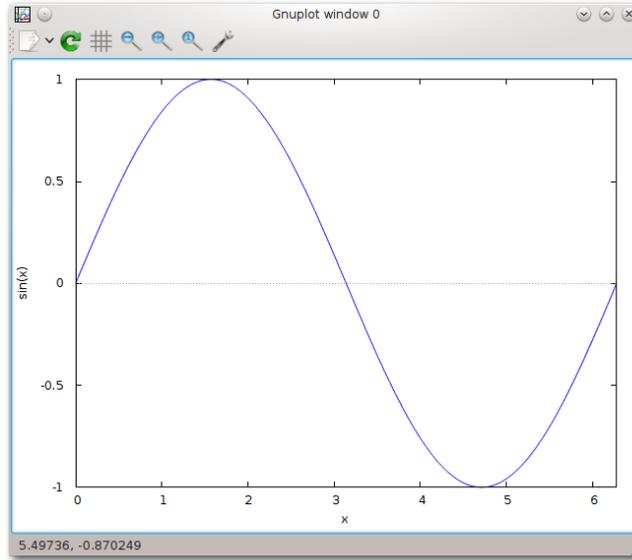
(%i31) plot2d(sin(x),[x,0,2*%pi]);
(%o31) /home/benutzername/maxout.~
gnuplot_pipes
    
```

Gnuplot erstellt jeweils Dateien im persönlichen Verzeichnis des Benutzers.

Beendet wird Maxima mit dem Befehl `quit()`.

Grafische Oberfläche

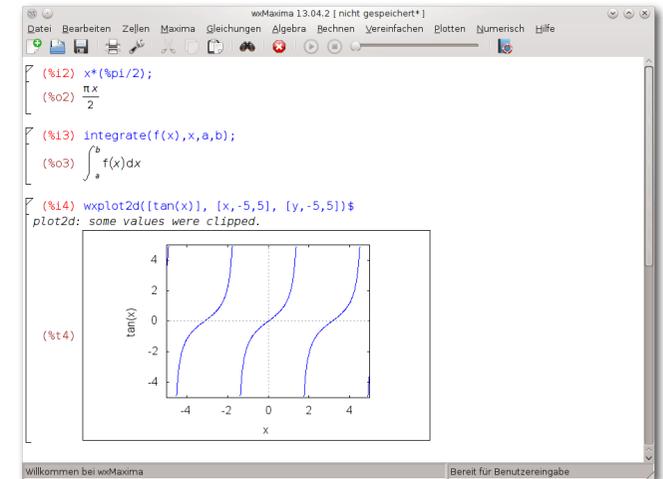
Zurück in der Gegenwart gibt es für all diese mathematischen Spielereien eine moderne grafische Oberfläche: wxMaxima [13].



Sinuskurve mit Gnuplot.

Sie basiert auf der Klassenbibliothek wxWidgets [14]. wxMaxima enthält zahlreiche Menüs, die die Bedienung vereinfachen. Die Anweisungen werden in Zellen geschrieben. Um eine neue Zelle zu erzeugen, klickt man in einen leeren Bereich der Arbeitsfläche. Das Menü „Zellen“ bietet verschiedene Optionen, um Zellen auszuwerten.

- [9] https://de.wikipedia.org/wiki/Binomische_Formel
- [10] <https://de.wikipedia.org/wiki/RSA-Kryptosystem>
- [11] <https://de.wikipedia.org/wiki/Stammfunktion>
- [12] <https://de.wikipedia.org/wiki/Differentialrechnung>
- [13] <http://www.gnuplot.info/>
- [14] <http://andrejv.github.io/wxmaxima/>
- [15] <http://www.wxwidgets.org/>



wxMaxima.

LINKS

- [1] <http://www.wolfram.com/mathematica/>
- [2] https://de.wikipedia.org/wiki/Stephen_Wolfram
- [3] <https://www.wolframalpha.com/>
- [4] <https://de.wikipedia.org/wiki/Computeralgebrasystem>
- [5] <http://www.maplesoft.com/products/Maple/>
- [6] <https://de.wikipedia.org/wiki/Kanada#Kultur>
- [7] <http://www.mathworks.de/products/matlab/>
- [8] <http://maxima.sourceforge.net/>

Autoreninformation

Tobias Kaulfuß (Webseite) studiert Mathematik, interessiert sich aber auch für Informatik und Physik.

Teilen

Kommentieren

Spamfilterung mit bogofilter von Dr. Diether Knof

Das Aussortieren von unerwünschten E-Mails (Spam) gehört zu den notwendigen und unbeliebten Aufgaben eines Internetnutzers. Normalerweise bietet ein E-Mail-Provider Spam-Filterung an. Eine Alternative oder auch Ergänzung dazu ist es, selber ein Filterprogramm zu verwenden.

Die Verwendung eines eigenen Filters hat den Vorteil, dass dieser auf den individuellen E-Mail-Verkehr spezialisiert werden kann. So sind für einen Sparkassen-Kunden vermutlich alle E-Mails zur Postbank Spam. Und für einige Personen sind alle englischsprachigen E-Mails Spam. Da Spam sich immer wieder verändert, sollte sich auch der Filter immer wieder anpassen. Darum muss sich der Anwender selber sorgen.

In diesem Artikel wird gezeigt, wie das Programm bogofilter [1] zum automatischen Erkennen von Spam eingesetzt werden kann. Analog lassen sich auch andere Programme wie spamassassin [2] verwenden. Bogofilter ist in den gängigen Distributionen im gleichnamigen Paket enthalten.

Funktionsweise

Bogofilter verwendet einen Bayes-Filter zur Spam-Erkennung. Vereinfacht beschrieben ist das Verfahren des Bayes-Filters wie folgt:

Zum Trainieren des Filters sortiert der Anwender zuerst E-Mails in Spam (unerwünschte) und Ham

(erwünschte). Der Filter zählt nun für jedes Wort wie häufig es in Spam und in Ham vorkommt.

Beispiel: Verteilung von Worten

Wort	Spam	Ham	insgesamt
Viagra	200 E-Mails	1 E-Mail	201 E-Mails
Linux	10 E-Mails	900 E-Mails	910 E-Mails
Bank	6 E-Mails	4 E-Mails	10 E-Mails

Zur Filterung einer neuen E-Mail wird auf Grundlage der in der E-Mail enthaltenen Worte die Bayes-Formel [3] verwendet, um die Wahrscheinlichkeit für Spam zu ermitteln.

Beispiele:

- Ist in der neuen E-Mail das Wort „Viagra“ enthalten, ist diesbezüglich die Wahrscheinlichkeit auf Spam $200/201 = 99,5\%$.
- Für „Linux“ ist die Wahrscheinlichkeit hingegen 1,1% und für „Bank“ 60%.

Aus den Wahrscheinlichkeiten auf Grundlage der einzelnen Worte wird eine Spam-Wahrscheinlichkeit für die gesamte E-Mail berechnet. Bogofilter ermittelt analog eine Wahrscheinlichkeit dafür, dass die E-Mail Ham ist. Die beiden Wahrscheinlichkeiten werden zu einem Wert „Bogosity“ zwischen 0 und 1 kombiniert. Dabei steht ein Wert von nahe 0 für Ham, von nahe 1 für Spam und für etwa 0,5 für eine unbekannte Einordnung. Bogofilter kategorisiert demnach die

E-Mails in die drei Kategorien „Spam“, „Ham“ und „Unbekannt“.

Für eine genauere Erläuterung verweist das Handbuch von bogofilter ([man bogofilter](#)) am Ende auf Artikel über verwendete Ansätze.

Anwendung in der Konsole

Im folgenden Beispiel wird das Verfahren von bogofilter verdeutlicht. Dabei wird von einer leeren Datenbank ausgegangen, die wie angegeben trainiert wird. Die verwendeten Programme bogofilter, bogolexer und bogoutil sind übrigens Bestandteil des bogofilter-Pakets. Damit die Ausgabe übersichtlich ist, wird kein echtes E-Mail-Format verwendet, die Worte werden nur durch einen Zeilenumbruch („\n“) getrennt.

Der Befehl

```
$ echo "Viagra\nLinux\nBank" | bogolexer
```

gibt aus, welche Worte (tokens) von bogofilter identifiziert werden und somit für die Filterung verwendet werden. Aufgrund der Vereinfachung in der Eingabe werden die Worte als Kopfzeilen einer E-Mail betrachtet und erscheinen mit vorangestelltem **head:**. Dieses wird im Folgenden ignoriert.

Mit den folgenden beiden Befehlen registriert man das Wort „Viagra“ als Spam und „Linux“ als Ham

(die Option `-s` steht für „Spam“, `-n` für „nicht Spam“).

```
$ echo "Viagra" | bogofilter -s
$ echo "Linux" | bogofilter -n
```

Bei Bedarf wird die Wortdatenbank unter `~/.bogofilter/wordlist.db` automatisch angelegt.

Den Inhalt der Wortdatenbank gibt man mit dem Befehl

```
$ bogoutil -d ~/.bogofilter/wordlist.db
```

aus. Die Ausgabe **head:Viagra 1 0** bedeutet, dass Viagra einmal in einer als Spam registrierten E-Mail vorkam und keinmal in einer als Ham registrierten.

Nachdem der Filter trainiert ist, kann eine E-Mail klassifiziert werden. Wie oben bereits erwähnt, verwendet bogofilter die drei Klassifikationen „Spam“, „Ham“ und „Unbekannt“.

Der Befehl

```
$ echo "Viagra" | bogofilter -T
```

zeigt mit der Ausgabe **S 0.991605** an, dass für den Text „Viagra“ die Klassifikation „Spam“ ist, mit der angegebenen Bogosity. Analog wird „Linux“ als Ham klassifiziert. Dementsprechend gibt der folgende Befehl

```
$ echo "Linux\nViagra" | bogofilter -T
```

die Klassifikation „Unbekannt“ aus, da die Bogosity dicht bei 1/2 liegt. Werden ausschließlich die Spam-Worte betrachtet („Viagra“), ist dies Spam; werden ausschließlich die Ham-Worte betrachtet („Linux“), ist dies Ham. Durch die Kombination beider Wahrscheinlichkeiten zur Bogosity heben sich beide Klassifikationen größtenteils auf.

Mit dem Befehl

```
$ echo "Bank" | bogofilter -s
```

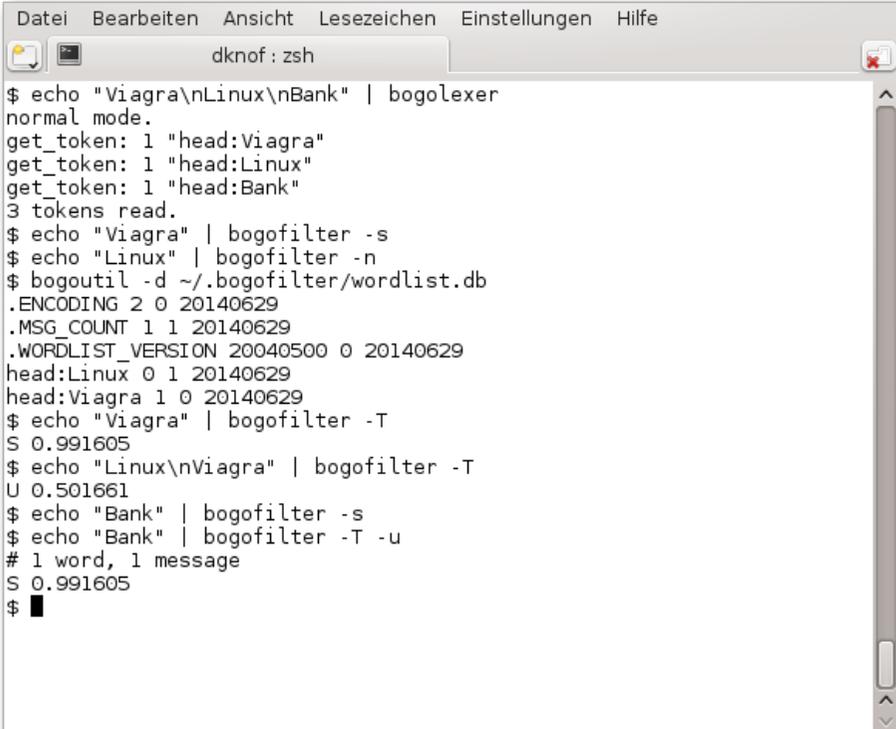
registriert man „Bank“ als Spam. Der Befehl

```
$ echo "Bank" | bogofilter -T -u
```

klassifiziert „Bank“ als Spam und führt eine automatische Registrierung als Spam durch. Danach ist die Ausgabe des Befehls

```
$ bogoutil -d ~/.bogofilter/wordlist.db
```

„Bank 2 0“, weil „Bank“ nun zweimal als Spam klassifiziert wurde.



```
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
dknof: zsh
$ echo "Viagra\nLinux\nBank" | bogolexer
normal mode.
get_token: 1 "head:Viagra"
get_token: 1 "head:Linux"
get_token: 1 "head:Bank"
3 tokens read.
$ echo "Viagra" | bogofilter -s
$ echo "Linux" | bogofilter -n
$ bogoutil -d ~/.bogofilter/wordlist.db
.ENCODING 2 0 20140629
.MSG_COUNT 1 1 20140629
.WORDLIST_VERSION 20040500 0 20140629
head:Linux 0 1 20140629
head:Viagra 1 0 20140629
$ echo "Viagra" | bogofilter -T
S 0.991605
$ echo "Linux\nViagra" | bogofilter -T
U 0.501661
$ echo "Bank" | bogofilter -s
$ echo "Bank" | bogofilter -T -u
# 1 word, 1 message
S 0.991605
$
```

Aufruf in der Konsole. 

Eine Klassifikation von „Bank“ als Spam entfernt man mit dem Befehl

```
$ echo "Bank" | bogofilter -S -n
```

und klassifiziert „Bank“ gleichzeitig einmal als Ham. Daher gibt der folgende Befehl

```
$ bogoutil -d ~/.bogofilter/wordlist.db
```

die Ausgabe **Bank 1 1** aus. So wie `-S` der Gegenpart zu `-s` ist, ist `-N` der Gegenpart zu `-n`.

Zu guter Letzt löscht der Befehl

```
$ rm ~/.bogofilter/wordlist.db
```

die Wortdatenbank, damit der produktive Einsatz unvorbelastet gestartet werden kann.

Anwendung in KMail

Der E-Mail-Client KMail [4] enthält einen Assistenten zur Spam-Bekämpfung, der bei der Einrichtung von Filterregeln für die Verwendung von bogofilter hilft. Für eine allgemeine Einführung in den Filtermechanismus von KMail siehe [freiesMagazin 05/2014 \[5\]](#).



Assistent Seite 3. 

Der Assistent lässt sich im Menü „Extras → Anti-Spam-Assistent“ aufrufen. Er bietet eine Auswahl der zur Verfügung stehenden Systeme an, darunter auch bogofilter, sofern das Programm installiert ist.

Im zweiten Schritt erscheinen zwei Abfragen für den Filter: diese entsprechen den Filteraktionen von den Spam-Filtern. Damit die Spam-E-Mails alle im Ordner Spam landen, muss hier der Ordner von „Papierkorb“ in „Spam“ geändert werden.

Im dritten Schritt des Assistenten werden die Filter angezeigt, die mit Betätigung von „Fertigstellen“ erzeugt werden: „Bogofilter Check“, „Spam-Bekämpfung“, „Als Spam klassifizieren“ und „Als Nicht-Spam klassifizieren“.

Der Assistent legt nun die Filter an. Es sind ganz normale Filter, die in den Filterregeln angeschaut sowie verändert werden können:

Der Filter „Bogofilter Check“ wird auf alle E-Mails bis zur Größe von 250 kB angewendet. Diese E-Mails werden durch das Programm **bogofilter -p -e** geleitet. Dieser Aufruf ergänzt die Kopfzeilen der E-Mail um das

Feld „X-Bogosity“, das enthält, ob die E-Mail als Spam eingeschätzt wurde. Weiterhin ist eingestellt, dass auch die folgenden Filter geprüft werden.

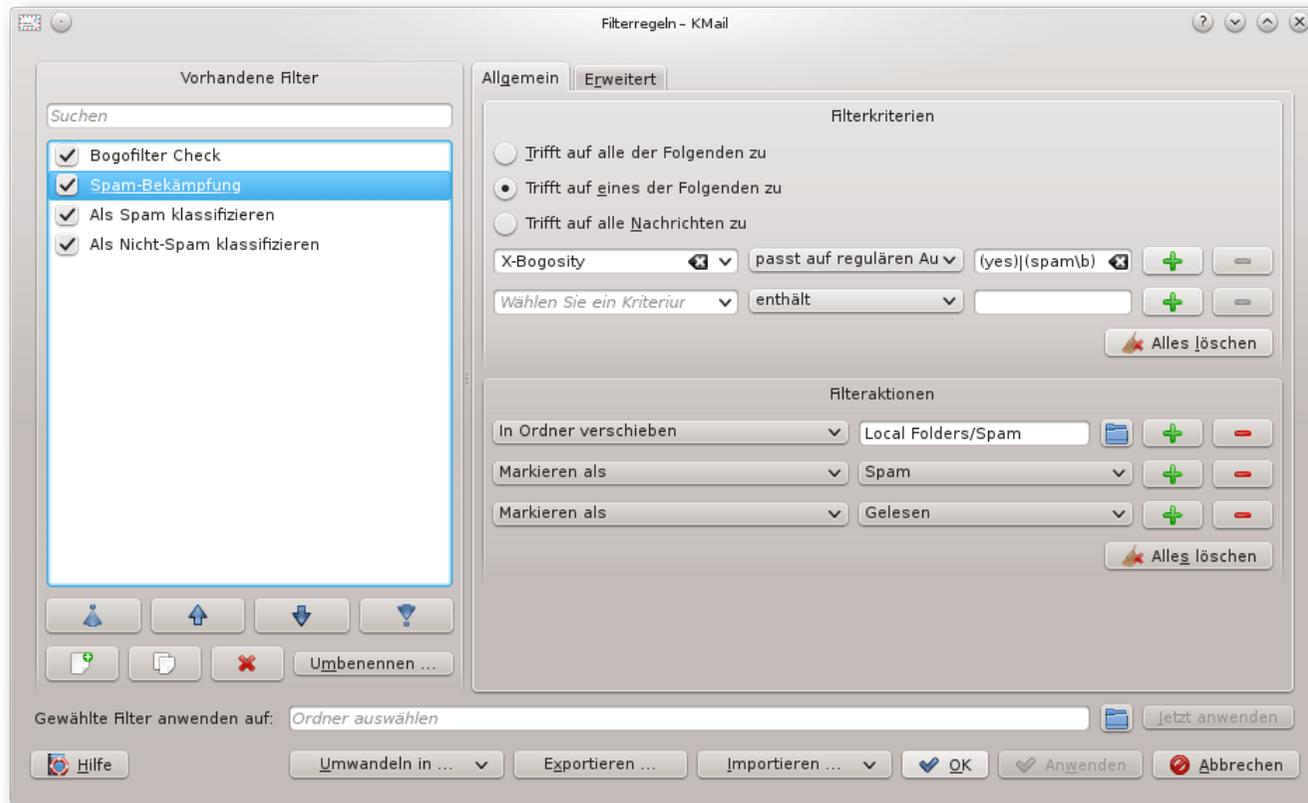
Der Filter „Spam-Bekämpfung“ prüft anschließend das Feld „X-Bogosity“. Wenn dort ein Spamverdacht eingetragen ist, wird die E-Mail in den Ordner Spam verschoben, als Spam markiert und als gelesen gesetzt.

Die Filter „Als Spam klassifizieren“ und „Als Nicht-Spam klassifizieren“ sind Filter, die nicht auf eingehende E-Mails angewendet werden, sondern nur manuell über das Menü „Filteraktionen anwenden“ und über die Werkzeugleiste aufgerufen werden können. Dazu gibt es in der Werkzeugleiste die beiden weitere Schaltflächen „Spam“ und „Ham“.

Über die Schaltflächen werden die markierten E-Mails als Spam beziehungsweise Ham registriert und eine entsprechende Funktion wird durchgeführt.

Wird beim Filter „Als Spam klassifizieren“ die Filteraktion „In Ordner verschieben“ durch „Nachricht löschen“ ersetzt, werden manuell als Spam klassifizierte E-Mails per Knopfdruck entfernt.

Wenn der Filter „Als Nicht-Spam klassifizieren“ um das Filterkriterium „In Ordner verschieben Posteingang“ ergänzt wird, lassen sich irrtümlich als Spam klassifizierte E-Mails mit einem Klick auf



Filter „Spam-Bekämpfung“.

„Ham“ aus dem Spam-Ordner in den Posteingang verschieben und gleich als Ham klassifizieren.

Angepasste Filter

Mit den vom Assistenten erzeugten Filtern ist für die Registrierung von E-Mails noch viel Handarbeit nötig. Diese wird sich der Anwender oft sparen. Um alle E-Mails in der Datenbank von bogofilter zu registrieren, ist es daher am einfachsten, die Registrierung

automatisch vom Filter durchführen zu lassen. Dazu sind die folgenden Änderungen am Filter vorzunehmen:

Den Filter „Spam-Bekämpfung“ ergänzt man um die Filteraktion „Befehl ausführen **bogofilter -s**“.

Den Filter „Spam-Bekämpfung“ dupliziert man und benennt ihn in „Ham-Registrierung“ um. Die

Filterbedingung ändert man in „X-Bogosity passt auf regulären Ausdruck (no)|(ham)|(unsure)b“ (nur der reguläre Ausdruck wird verändert). Alle Filteraktion löscht man und fügt als einzigen Filter „Befehl ausführen **bogofilter -n**“ ein. Die weiteren Einstellungen „Diesen Filter bei der manuellen Filterung anwenden“ und „Bearbeitung hier beenden, falls Filterbedingung zutrifft“ wählt man ab.

Beim Filter „Als Spam klassifizieren“ ändert man die zweite Filteraktion in „Befehl ausführen **bogofilter -N -s**“, fügt also die Option **-N** hinzu.

Entsprechend ändert man beim Filter „Als Nicht-Spam klassifizieren“ die zweite Filteraktion in „Befehl ausführen **bogofilter -S -n**“.

Zum Verständnis der Änderungen hilft es, sich zu überlegen was in den folgenden Fällen passiert:

- Eine Spam-E-Mail wird empfangen und als Spam klassifiziert.
- Eine Spam-E-Mail wird empfangen und nicht als Spam klassifiziert (falsch-Positiv-Klassifizierung). Anschließend wird der Filter „Als Spam klassifizieren“ manuell aufgerufen.
- Eine Ham-E-Mail wird empfangen und als Ham oder Unbekannt klassifiziert.
- Eine Ham-E-Mail wird empfangen und als Spam klassifiziert (falsch-Negativ-Klassifizierung). Anschließend wird der Filter „Als Nicht-Spam klassifizieren“ manuell aufgerufen.

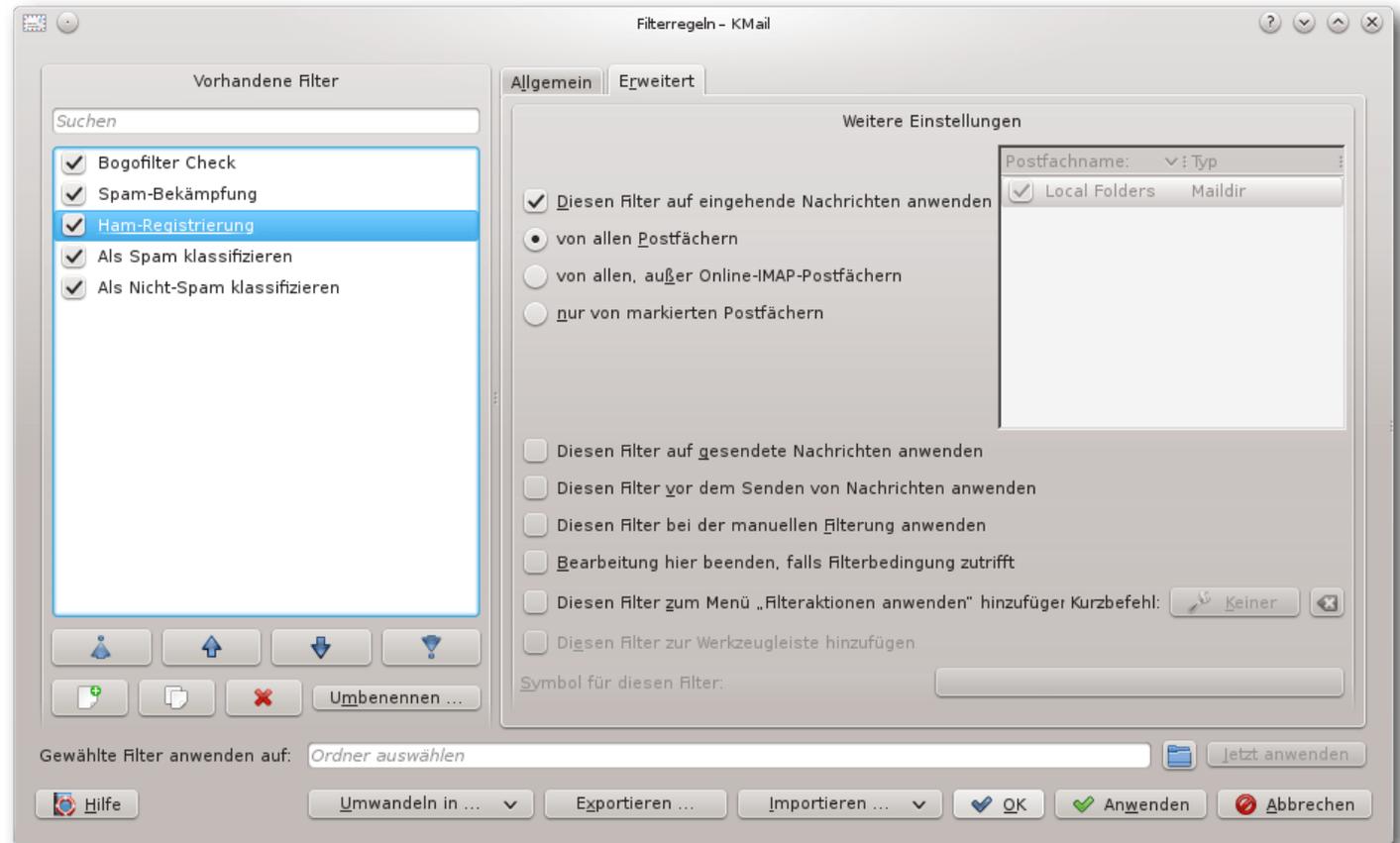
Bogofilter bietet mit dem Schalter **-u** die Möglichkeit, E-Mails entsprechend der Klassifizierung zu registrieren (siehe oben). Dies wird hier nicht verwendet, da ansonsten E-Mails, die weder als Spam noch als Ham erkannt werden, nicht klassifiziert werden. Aber gerade für als Unbekannt klassifizierte E-Mails ist eine Registrierung wichtig, damit der Filter immer besser wird.

Um E-Mails von bestimmten Personen nie als Spam einzuordnen, kann man die Filterkriterien bei „Bogofilter Check“ entsprechend ergänzen. Alternativ kann man direkt darauf folgend einen separaten Filter erstellen, in dem das Feld „X-Bogosity“ entfernt wird.

Grenzen von bogofilter

Bogofilter ist ein Filter, der ausschließlich Text und keine Metadaten zur Filterung heranzieht; außerdem werden Anhänge (wie PDF-Dokumente) nicht berücksichtigt. Für eine Filterung anhand der Anhänge hilft ein Anti-Viren-Filter (siehe entsprechenden Assistenten von KMail) sowie einfache manuelle Filter, die nach .pdf.exe oder Ähnlichem suchen.

Damit der Filter gut zwischen Spam- und Ham unterscheiden kann, muss er trainiert werden. Dies kann und sollte im laufenden Betrieb geschehen (über die Werkzeugleiste), sorgt aber erst einmal für Frust, da anfangs einige E-Mails falsch klassifiziert werden.



Erweiterte Einstellungen des modifizierten Filters „Als Spam klassifizieren“. 🔍

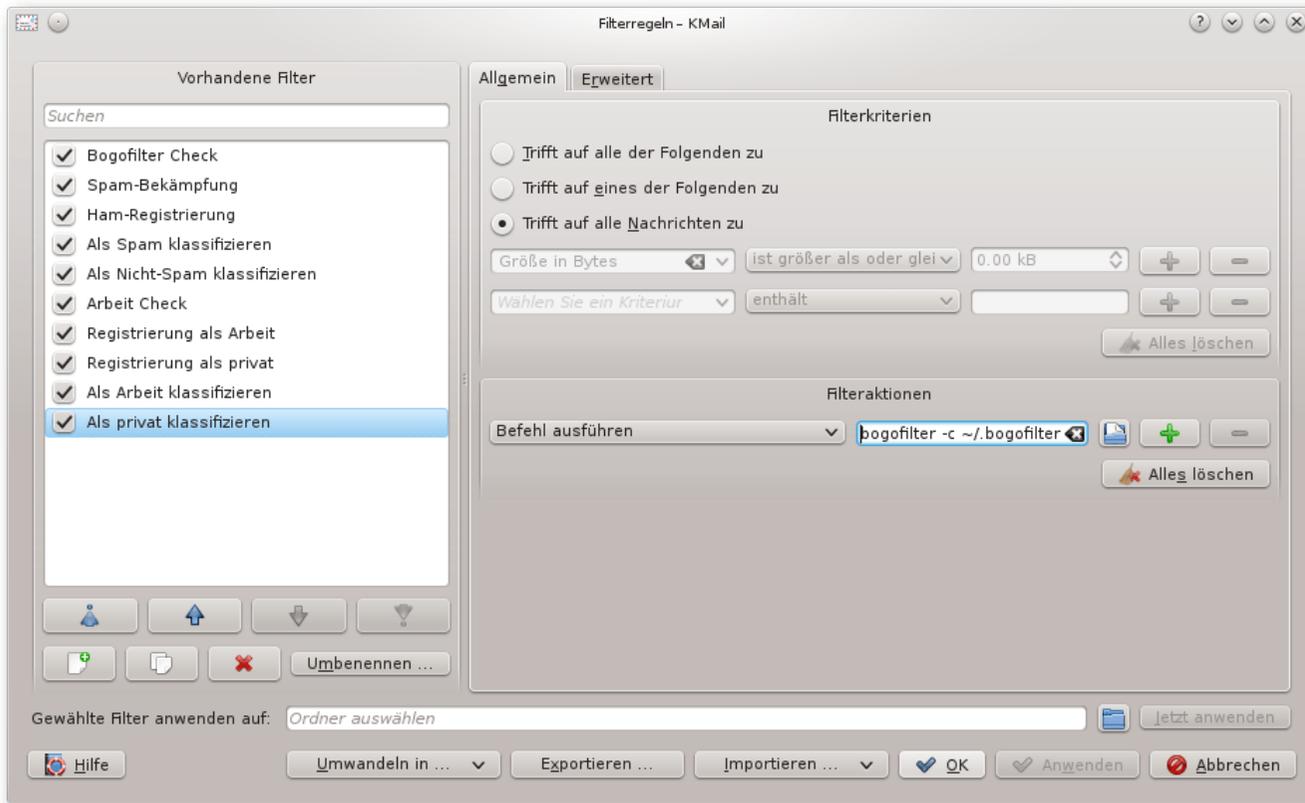
Andere Spam-Filter wie spamassassin verwenden neben dem Bayes-Filter noch weitere Methoden wie statische Regeln ähnlich zu Virensignaturen und Online-Abfragen (Prüfsummen und Mail-Server).

Bogofilter arbeitet ausschließlich lokal, damit kann es nicht eingesetzt werden, wenn die Web-Anwendung vom Provider verwendet wird.

Verschlüsselte E-Mails können nur über die Kopfdaten der E-Mail gefiltert werden, da der E-Mail-Text für bogofilter Kauderwelsch ist. Dies ist für die Praxis kein Problem, da Spam bislang nicht verschlüsselt versendet wird.

Weitere Anwendung

Bogofilter ist ein Programm, das der Unix-Philosophie „Mache nur eine Sache und mache



Zweite Filterung: Arbeit/privat. 🔍

sie gut“ folgt. Bogofilter kann gut E-Mails anhand des E-Mail-Texts in zwei Klassen Spam und Ham einordnen.

Bogofilter lässt sich daher nicht nur für die Spam-Bekämpfung verwenden sondern kann allgemein verwendet werden, um E-Mails abhängig vom Inhalt in zwei Gruppen zu sortieren. Als Beispiel sollen E-Mails automatisch in „Arbeit“ und „privat“ sortiert werden. So ein Filter funktioniert

allerdings nur dann gut, wenn die Inhalte der E-Mails nicht zu ähnlich sind. Dafür wird eine gesonderte Konfigurationsdatei `~/.bogofilter/Arbeit.cf` erstellt, die auf eine separate Wortdatenbank `/home/user/.bogofilter/Arbeit_wordlist.db` (hier muss `user` durch den Benutzernamen ersetzt werden, da `~` beim Test nicht funktionierte) und eine eigene Kopfzeile „X-Bogosity-Arbeit“ verweist. Nun können die E-Mail-Filter zur Spam-Bekämpfung kopiert und entspre-

chend angepasst werden, alle Aufrufe von bogofilter müssen dabei um ein weiteres Argument `-c ~/.bogofilter/Arbeit.cf` ergänzt werden.

Fazit

Bogofilter erzielt nach einer kurzen Lernphase (ein paar hundert E-Mails) gute Ergebnisse beim Sortieren der E-Mails in Spam und Ham. Die Einbindung in E-Mail-Clients funktioniert gut, bei KMail wird sie durch einen Assistenten unterstützt. Andere Spam-Filter verwenden noch weitere Verfahren, die einerseits die Spam-Erkennung verbessern können, andererseits aber deren Komplexität erhöhen. Wichtig ist, den Bayes-Filter immer weiter zu trainieren.

LINKS

- [1] <http://bogofilter.sourceforge.net/>
- [2] <https://spamassassin.apache.org/>
- [3] https://de.wikipedia.org/wiki/Bayesscher_Filter#Mathematische_Grundlage
- [4] <http://kde.org/applications/internet/kmail/>
- [5] <http://www.freiesmagazin.de/freiesMagazin-2014-05>

Autoreninformation

Dr. Diether Knof ist seit 1998 Linux-Anwender. In seiner Freizeit entwickelt er das freie Doppelkopfspiel [FreeDoko](#).

✉ [Teilen](#)

💬 [Kommentieren](#)

Registerhaltiger Satz mit L^AT_EX von Dominik Wagenführ

Wer eine aktuelle Zeitschrift auf dem Markt aufschlägt, dem wird wahrscheinlich auffallen, dass bei einem mehrspaltigen Layout benachbarte Zeilen immer auf einer Höhe liegen. Diese Eigenschaft nennt man Registerhaltigkeit [1]. L^AT_EX bietet von sich aus keine Möglichkeit, Texte registerhaltig zu setzen. Der Artikel soll zeigen, wie man dies dennoch mit etwas Handarbeit erhalten kann.

Wofür ist Registerhaltigkeit gut?

Viele Leser werden sich vielleicht fragen, wozu man bei seinen L^AT_EX-Dokumenten überhaupt Registerhaltigkeit benötigt. Ähnlich wie viele, die mit L^AT_EX erstellten Dokumente als typographisch schön empfinden, gibt es Menschen, die auch Registerhaltigkeit schön finden. Andere wiederum sehen darin eher einen optischen Makel, da das Layout nicht mehr so lebendig wirkt, weil die Zeilen in ein starres Raster gepresst sind.

Es gibt aber auch einen praktischen Nutzen: Bei sehr dünnen Papierseiten ist es störend, wenn die die Zeilen der Rückseite zwischen die Zeilen der Vorderseite fallen. Daher bemüht man sich hier besonders um die Einhaltung des Registers, sodass die Zeilen von Vorder- und Rückseite auf einer Höhe liegen.

Hintergrund für den Artikel ist das Linux-Magazin *freiesMagazin* [2]. Dieses wird seit seiner zweiten

Ausgabe im Jahr 2006 mit L^AT_EX gesetzt. Im Laufe der Zeit hat es sich immer mehr gewandelt und wird heute nebed anderen Formaten als PDF-Version im DIN-A4-Querformat mit drei Spalten gesetzt.

Bis zur Ausgabe 04/2013 (einschließlich) gab es im Magazin keine Registerhaltigkeit. Die Spalten wurden immer bis zum Spaltenende gestreckt, sodass sie zwar unten und oben bündig waren, in der Mitte gab es zwischen Absätzen aber immer wieder variable Abstände, wenn beispielsweise Überschriften oder Codeblöcke eingebettet waren. Um ein einigermaßen gleichmäßiges Layout zu erzeugen, wurde sehr viel manuell mit vertikalen Abständen nachjustiert, was eine mühsame Arbeit war.

Auf der DANTE-Frühjahrstagung (siehe *freiesMagazin* 04/2013 [3]) hielt Patrick Gundlach einen (ungeplanten) Vortrag zum Thema Registerhaltigkeit und zeigte einen Weg, wie man die L^AT_EX-Elemente im Text so ausrichten kann, dass sie alle auf einer Grundlinie liegen. Dieses Konzept wurde daraufhin bei *freiesMagazin* aufgegriffen und stellt die Grundlage für diesen Artikel dar.

Die Registerhaltigkeit erleichtert das Layout des Magazins ungemein, da man sich nicht mehr darum kümmern muss, in die vertikalen Abstände

inels

ner Dialog, dessen Inhalt mit Javascript erstellt wird. Nach dem Modul `panel` [2] wird ein Panel:

```
require("panel");
panel("self");

els.Panel ({
  self.data.url("panel.~
```

Hinweis: Man kann `http://...org` auch direkt als `content` Umweg über `panel.html` ist auf satzfunktionen notwendig.

Damit der `iframe` [4], in dem gezeigt wird, auch den gesamten Inhalt ausnutzt, werden ein paar Zeilen der Datei `panel.css`, die ebenfalls `data` abgelegt ist, festgelegt:

Bis April 2013: Benachbarten Zeilen liegen nicht auf einer Höhe. 🔍

s beim ersten Update dasselbe gilt auch hierfür stellt die in scheidung dar, nur e Software in den aufzunehmen. Das n Video-Formaten

scheidung von ande

Grafische Oberfläche
GNOME wird in openSUSE ausgeliefert. Der Schlag laut Aussagen der serung der Umgebun

Registerhaltigkeit: Benachbarten Zeilen liegen auf einer Höhe. 🔍

manuell eingzugreifen – zumindest meistens. Es gibt immer wieder mal Ausnahmefälle, wo L^AT_EX doch anderer Meinung ist, was die Abstände angeht.

Beispiel

Als Beispiel für anfängliche Nicht-Registerhaltigkeit soll folgendes kleines Code-Beispiel dienen:

```

\documentclass[landscape,parskip=half-]{\screpr}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
\usepackage{multicol}
\usepackage{blindtext}
\usepackage{graphicx}
\usepackage[draft,columns=1]{typogrid}

\begin{document}
\pagestyle{empty}
\raggedcolumns

\begin{multicols*}{3}
\section*{Blindtext}
\blindtext
\begin{enumerate}
\item Eins
\item Zwei
\end{enumerate}
\blindtext
\subsection*{blindtext}
\blindtext\par
\includegraphics[width=0.95\linewidth]{\bild.png}\par
\blindtext
\end{multicols*}

\end{document}

```

Listing 1: *register-beispiel.tex*

Das Dokument ist im A4-Querformat gesetzt, damit man drei Spalten noch lesbar nebeneinander

setzen kann. Für die Darstellung des Textkörpers wird das L^AT_EX-Paket **typogrid** genutzt. Alternativ wäre auch das Paket **showframe** möglich gewesen, was zusätzlich noch Kopf- und Fußzeile sowie den Rand einrahmt. Das Dokument wird dann noch mit Blindtext gefüllt.

Das Kommando `\raggedcolumns` ist für spätere Zwecke zwingend notwendig. Ansonsten fügt L^AT_EX in den Zwischenräumen vertikale Abstände ein, sodass die Spalten immer bis zu Spaltenende gestreckt sind. Gerade dieses Verhalten will man beim registerhaltigen Satz unterbinden und stattdessen selbst die Abstände definieren. Die Stern-Version der **multicols**-Umgebung wird nur genutzt, damit die Spalten nicht gleichmäßig sondern fortlaufend aufgefüllt werden.

Hilfslinien einblenden

Bevor man sich an den registerhaltigen Satz wagt, sollte man Hilfslinien einblenden, sodass man optisch prüfen kann, ob die einzelnen Elemente auch zur Grundlinie ausgerichtet sind. Ein Rahmen um den Textkörper wird bereits angezeigt.

Für die Hilfslinien gibt es theoretisch ein paar hilfreiche L^AT_EX-Pakete [4] [5].

graphpap

Das Paket **graphpap** [6] könnte vermutlich ein Gitternetz darstellen. Leider gibt es keine Dokumentation zu dem Paket, sodass nicht bekannt ist, welche Argumente der Befehl `\graphpaper` benötigt.

pagegrid

Das **pagegrid**-Paket [7] erlaubt die Darstellung eines Gitternetzes. Eine einigermaßen sinnvolle Verwendung wäre:

```

\usepackage[t1,step=\baselineskip,arrows=false]{pagegrid}

```

Über die Option **t1** wird nur ein Gitter (anstatt zwei) angezeigt. Die Schrittweite **step=\baselineskip** ermöglicht die Darstellung der Grundlinie.

Das Problem ist aber, dass zum einen die ganze Seite mit dem Gitter belegt wird und nicht nur der Textkörper. Zusätzlich lassen sich die horizontalen Linien nicht ausschalten, was die Lesbarkeit des Textes erschwert. Und das dritte Problem ist, dass man keinen Offset definieren kann, sodass die Linien mit dem Textkörper anfangen.

Insofern ist das Paket **pagegrid** nicht geeignet.

vgrid

Das Paket **vgrid** [8] stellt die Grundlinie im Text wiederholt dar und kann einfach über

```

\usepackage{vgrid}

```

eingebunden werden. Die Darstellung erfolgt dabei nur im Textkörper und auch nur vertikal wie gewünscht. Leider ist aber kein Offset möglich, was dazu führt, dass auch die erste Zeile in einer Spalte nicht auf der Grundlinie liegt.

Blindtext	ich schreibe: »Dies ist ein Blindtext«	sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein.
Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	»Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	
	blindtext	
	Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	
1. Eins		
2. Zwei		
Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.		

Anzeige der Grundlinien mit `vgrid`, leider ohne Offset. 🔍

graphpaper

graphpaper ist leider kein Paket, was man auf CTAN herunterladen kann. Der Code stammt von Heiko Oberdiek [9] und wurde von Patrick Gundlach angepasst und auf der DANTE-Frühjahrstagung (siehe oben) vorgestellt.

Aufgrund der Größe des Listings, wird es hier nur verlinkt: graphpaper.sty.

Wenn man den Code als **graphpaper.sty** abspeichert und dann per

```
\usepackage{graphpaper}
```

einbindet, wird die Grundlinie in einem hellen Grau angezeigt und ist so ausgerichtet, dass die Zeilen per Standard auf der Grundlinie liegen, wenn kein \LaTeX -Element den Inhalt verschiebt. Im Beispiel liegen die ersten Zeilen der Spalte 2 und 3 auf der Grundlinien, in Spalte 1 verschiebt der Abstand unter einer Überschrift das Register. Zusätzlich werden noch Zeilennummern angezeigt, was hilfreich ist, wenn sich mit anderen Bearbeitern über bestimmten Zeilen austauschen will.

Anpassung der \LaTeX -Elemente

Wie gesagt gibt es kein einzelnes \LaTeX -Paket, das man einbinden kann, um Registerhaltigkeit

Blindtext	ich schreibe: »Dies ist ein Blindtext«	sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein.
Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	»Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	
	blindtext	
	Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.	
1. Eins		
2. Zwei		
Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig ob ich schreibe: »Dies ist ein Blindtext« oder »Huardest gefburn«?. Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muß keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie »Lorem ipsum« dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.		

Anzeige der Grundlinien mit **graphpaper**. 🔍

zu erzeugen, da bei sehr vielen Makros vertikale Abstände eingefügt werden. Die Idee ist daher, dass man die verwendeten Elemente (d. h. Überschriften, Aufzählungen, Listings, Bilder etc.) neu definiert bzw. die vertikalen Abstände manuell festlegt. Hierfür gibt es sehr oft vorhandene \LaTeX -Pakete, die dies ermöglichen.

Absatzabstand

Für das Beispiel wurde mit Absicht eine Version gewählt, die neue Absätze nicht einrückt, wie dies vielleicht von einigen wissenschaftlichen Arbeiten bekannt ist, sondern einen vertikalen Abstand zwischen zwei Absätzen lässt [10].

Damit der Abstand zwischen zwei Absätzen korrekt ist, definiert man diesen einfach als den Abstand zwischen zwei Grundlinien:

```
\setlength{\parskip}{\baselineskip}
```

In Spalte 3 des Beispiels sieht man dann schon, dass alle Zeilen auf der Grundlinie liegen. Hier ist die Registerhaltigkeit also bereits erreicht.

Überschriften

Auch der vertikale Abstand vor und nach Überschriften muss neu definiert werden, da die Voreinstellung nicht mit dem Register zusammenpasst. Für die Anpassung kann das Paket **titlesec** [11] benutzt werden. Die Abstände für eine Überschrift legt man dann per

```
\titlespacing{\section}{0pt}{-4.4pt}{-\baselineskip}
\titlespacing{\subsection}{0pt}{-0.4pt}{-\baselineskip}
```

fest. Das Makro **\titlespacing** hat dabei vier Argumente. Der erste Wert entspricht der linken Einrückungen, der zweite dem Abstand vor und der dritte dem Abstand nach der Überschrift. Die angegebenen Werte wurden durch Testen festgelegt.

Beachten sollte man, dass diese keine relativen Werte darstellen, die bei jeder Schriftgröße funktioniert. Ändert man beispielsweise die Schriftgröße auf 12pt, wären die Werte

```
\titlespacing{\section}{0pt}{-7.6pt}{-1.1\baselineskip}
\titlespacing{\subsection}{0pt}{-3.7pt}{-\baselineskip}
```

Wendet man diese Definitionen auf das Beispiel an, sieht man, dass die zweite Spalte mit der Subsection-Überschrift korrekt ausgerichtet wird. In Spalte 1 wird die erste Überschrift aber dennoch nicht korrekt ausgerichtet ist. Grund ist, dass die über **\titlespacing** eingestellten Werte nur ein **\vspace** (ohne Stern [12]) vor einer Überschrift einfügen. Steht die Überschrift am Spaltenanfang, wird der Abstand daher ignoriert.

Abhilfe schafft hier die manuelle Korrektur, indem man vor den Überschriften manuell den Abstand verringert:

```
\vspace*{-2\baselineskip}%
\section*{Blindtext}
```

Aufgrund der Behandlung der obersten Zeile in **T_EX** [13], müssen hier zwei Grundlinienhöhen abgezogen werden. Leider gibt es keinen einfachen Weg, automatisch zu berechnen, dass es sich um die oberste Zeile in einer Spalte handelt, daher muss man diese Abstandkorrektur immer manuell angeben bzw. definiert sich ein neues Section-Makro und eine entsprechende Option (siehe hierzu auch „Variable Argumente in **L^AT_EX** nutzen“, **freiesMagazin** 08/2011 [14]):

```
\usepackage{xkeyval}
```

```
\makeatletter
\define@boolkey{Section}{top}[true]{%
  \columnbreak%
  \vspace*{-2\baselineskip}%
}
\makeatother

\newcommand{\Section}[2][ ]{%
  \setkeys{Section}{#1}%
  \section*{#2}%
}

\Section[top]{Blindtext}
```

Wird die Option **top** angegeben, wird zum einen die Spalte abgebrochen und zum anderen die Überschrift oben bündig gesetzt.

Auf die gleiche Art müsste man eine Kompensation einbauen, wenn die Überschrift über mehrere Zeilen geht. Darin sieht man auch, dass die Lösung nicht allgemeingültig ist.

Aufzählungen und Auflistungen

Für die Umdefinition von Aufzählungen und Auflistungen nutzt man am besten das **enumitem**-Paket [15]. Hierüber kann man sehr gut die vertikalen (und horizontalen) Abstände einstellen. Eine eigene Aufzählungsumgebung könnte so aussehen:

```
\usepackage{enumitem}
```

```
\newenvironment{Enumerate}[1][]{%
  \par
  \begin{enumerate}[topsep=-0.21\~
  baselineskip,noitemsep]%
}{%
  \end{enumerate}%
}
```

Mit **noitemsep** gibt es zwischen zwei Einträgen keinen Abstand mehr. Der Wert für **topsep** fügt über und unter der Aufzählung den angegebenen Abstand ein. Der Wert wurde erneut durch Ausprobieren festgelegt, bei einer anderen Schriftgröße ist er kleiner oder größer. Zusätzlich wird vor der Aufzählung mit **\par** der aktuelle Absatz beendet, da sonst die Abstände nicht stimmen.

Analog kann man sich auch eine neue **Itemize**-Umgebung definieren.

Exkurs: Abstände berechnen

Bisher konnte man alle vertikalen Abstände sehr leicht umdefinieren bzw. beeinflussen, sodass alle Zeilen immer auf der Grundlinie liegen. Bei Bildern hat man dabei aber ein Problem, da je nach Bildgröße bzw. genauer Bildhöhe der Abstand unter dem Bild bis zur nächsten Grundlinie stark variieren kann.

Die Idee, in so einem Fall eine Registerhaltigkeit zu erhalten, ist einfach: Verschiebe das aktuelle L^AT_EX-Element (nach dem Bild), sodass es auf der Grundlinie liegt. Hierfür müsste man nur ausrechnen, wo das aktuelle Element gerade gedruckt

werden soll, den Abstand zur nächsten Grundlinie ausrechnen und dann das Element dahin verschieben.

gridset

So eine Berechnung gibt es bereits im Paket **gridset** [16], das von Markus Kohm entwickelt wurde. Es stellt einen Befehl **\vskipnextgrid** zur Verfügung, der einen vertikalen Abstand einfügt, sodass das nächste Element auf der Grundlinie liegt.

Eine Beispielanwendung, um den ersten Absatz auf die Grundlinie zu hieven, wäre:

```
\begin{multicols*}{3}
\section*{Blindtext}
\vskipnextgrid
\blindtext
```

Übersetzt man das Beispieldokument neu, sieht man keine Veränderung zum Original. Übersetzt man aber noch einmal, wird der Absatz korrekt auf die Grundlinie geschoben.

Der Grund für dieses Verhalten ist die Funktionsweise von **\vskipnextgrid**. Dieses speichert die aktuelle Position über das Kommando **\savepos** zuerst in die zugehörige **aux**-Datei. Erst bei einer zweiten Übersetzung wird diese Position herangezogen, um den Abstand einzufügen.

Hat man also viele solche Elemente auf der Seite, passiert es, dass es zahlreiche L^AT_EX-Übersetzungen

benötigt, ehe alle Objekte an ihrem Platz sind. Der zweite L^AT_EX-Lauf verschiebt den ersten Absatz an seine korrekte Stelle. Eine zweite, in der **aux**-Datei gespeicherte Position ist damit gegebenenfalls hinfällig, weil sich der Text verschoben hat. So benötigt man einen dritten Lauf etc.

In der Folge führt das leider zu einem unschönen Effekt, den man an diesem Beispiel sieht:

```
\vskipnextgrid
\includegraphics[width=0.95\linewidth]{~
bild.png}\par
\vskipnextgrid
```

Das Beispieldokument kann man nun endlos übersetzen, es wird sich nie ein finaler Zustand ohne Änderung einstellen. Der Text springt bei jedem Lauf immer zwischen zwei Versionen hin und her, weil sich die Abstände jedes Mal neu berechnen.

Eine ähnliche Version hat Patrick Gundlach auf der DANTE-Frühjahrstagung (siehe oben) vorgestellt, bei der die aktuelle Position ebenfalls mit **\zsavepos** bestimmt wurde. Diese hatte aber den gleichen Nachteil, dass man sehr oft übersetzen muss und es ggf. zu obigen Sprüngen im Text und Endlosübersetzungen kommen kann.

Direkte Berechnung

Das Problem der Berechnung mit **\zsavepos** ist die Zwischenspeicherung des Wertes in der **aux**-Datei und die Benutzung im nächsten L^AT_EX-Lauf (siehe oben). Es gibt aber auch eine andere Mög-

lichkeit, dies während eines L^AT_EX-Laufes zu berechnen.

Mithilfe von `\pagetotal` kann man die aktuelle Position des Textes bestimmen. Das ist aber nicht ganz korrekt, sondern es gibt die aktuelle Position am Anfang des Paragraphen an, in dem der Text steht. Das ist aber kein Hindernis, denn man möchte ja immer nur ganze L^AT_EX-Elemente (Aufzählungen, Absätze, Bilder etc.) an der Grundlinie ausrichten.

Über folgende Rechnung kann man dann bestimmen, wie weit der nächste Block verschoben werden muss:

$$b - \left(p - \left\lfloor \frac{p}{b} \right\rfloor \cdot b \right)$$

wobei p `\pagetotal` entspricht und b `\baselineskip`. Die Formel gibt effektiv nur das an, was als Idee am Anfang des Kapitels geschrieben wurde.

Für die Berechnung wird das Paket `fp` [17] benutzt. Zusätzlich muss man von den diversen Längen, die in Point (pt) angegeben sind, die Einheit entfernen, wozu das Hilfsmakro `\getlength` dient. Danach kann man effektiv sehr leicht die Modulo-Operation mit Floating-Point-Zahlen erstellen. Die Abfrage, ob das Zwischenergebnis 0 ist, ist notwendig, da man ansonsten jeden Eintrag, der bereits exakt auf der Grundlinie liegt, um eine Zeile verschieben würde.

Umgesetzt in L^AT_EX-Code ist dies:

```
\usepackage{fp}

\makeatletter
\newcommand*{\getlength}[1]{\strip@pt#1}
\makeatother

\newcommand{\FPmod}[3]{%
  \FPdiv{\fmFPdivTmp}{\getlength{#2}}{\getlength{#3}}%
  \FPtrunc{\fmFPtruncTmp}{\fmFPdivTmp}{0}%
  \FPMul{\fmFPMulTmp}{\fmFPtruncTmp}{\getlength{#3}}%
  \FPsub{\fmFPsubTmp}{\getlength{#2}}{\fmFPMulTmp}%
  \FPifzero{\fmFPsubTmp}%
  \setlength{#1}{0pt}%
  \else%
  \FPsub{\fmFPsubTmpTwo}{\getlength{#3}}{\fmFPsubTmp}%
  \setlength{#1}{\fmFPsubTmpTwopt}%
  \fi%
}

\newlength{\vskiplength}
\newlength{\vskipsavepos}

\newcommand{\vskipnextline}{%
  \setlength{\vskipsavepos}{\pagetotal-11pt}%
  \FPmod{\vskiplength}{\vskipsavepos}{\baselineskip}%
  \vspace{\vskiplength}%
}
```

Für das Makro `\vskipnextline` werden zwei Längen `\vskipsavepos` und `\vskiplength` definiert, welche die gespeicherte Position und den effektiven einzufügenden vertikalen Abstand erhalten. Wichtig ist, dass von der aktuellen Position noch einmal die aktuelle Schriftgröße abgezogen werden muss. Der Wert `\f@size` kann nicht benutzt werden, da dieser bei 11pt auf 10.95 steht

und damit nicht exakt den Offset trifft, den es abzuziehen gilt.

Anpassung der L^AT_EX-Elemente (Fortsetzung)

Bilder

Für das Bild kann man sich nun ein Makro definieren, so dass der Text nach Bildern immer auf der Grundlinie liegt, egal, wie man das Bild skaliert.

```
\newcommand~
{\IncludeImage}[2][]{%
  \includegraphics
    [#1]{#2}\par%
  \vskipnextline%
}
```

Weitere Elemente

Listings lassen sich mit dem Paket `listings` [18] und der Umgebung `lstnewenv`

`ironment` definieren. Wichtig ist, dass man `aboveskip` und `belowskip` auf 0pt setzt und danach die Rahmenstärke und den Randabstand der Listingsbox mittels `\vspace` manuell hinzufügt, da diese nicht mitgezählt werden.

Gleiches gilt ebenfalls auch für Tabellen, die man sich eigens definieren kann.

Auf die gleiche Art und Weise kann man sich weitere LaTeX-Elemente umdefinieren. Manchmal gibt es bereits Pakete, mit denen man den vertikalen Abstand einstellen kann. Manchmal benötigt man `\skipnextline`. In manchen Fällen hilft aber gar nichts bis auf die direkte manuelle Korrektur mit `\vspace`.

Fazit

Registerhaltiger Satz mit LaTeX ist möglich, aber auch mit manueller Arbeit verbunden, wie man sieht. Wenn man aber einmal die notwendigen Umgebungen und Befehle definiert hat, sind diese in der Regel stabil und können lange genutzt werden. Eine allgemeingültige Lösung, mit der LaTeX automatisch alle Element an der Grundlinie ausrichtet, gibt es aber bisher noch nicht.

Auf der DANTE-Frühjahrstagung 2014 in Heidelberg gab es auf die Vorstellung des Themas viel Resonanz, sodass es besser wäre, mehr mit den Box-Mechanismen von LaTeX zu arbeiten anstatt die Absätze einfach nur zu verschieben, da dies auch nicht immer korrekt funktioniert. Ein Folge-

artikel zu dem Thema liegt also im Bereich des Möglichen, sodass es weniger Sonderfälle zu betrachten gibt.

LINKS

- [1] <http://www.typolexikon.de/r/registerhaltigkeit.html>
- [2] <http://www.freiesmagazin.de/>
- [3] <http://www.freiesmagazin.de/freiesMagazin-2013-04>
- [4] <http://dante.ctan.org/tex-archive/help/Catalogue/bytopic.html#gridlayout>
- [5] <http://dante.ctan.org/tex-archive/help/Catalogue/bytopic.html#grid>
- [6] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/graphpap.html> 
- [7] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/pagegrid.html> 
- [8] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/vgrid.html> 
- [9] <http://tex.stackexchange.com/a/70108> 
- [10] [https://de.wikipedia.org/wiki/Absatz_\(Text\)](https://de.wikipedia.org/wiki/Absatz_(Text))
- [11] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/titlesec.html> 
- [12] https://de.wikibooks.org/wiki/LaTeX-Wörterbuch:_vspace
- [13] <http://texhacks.blogspot.de/2010/12/vspace-is-broken.html> 
- [14] <http://www.freiesmagazin.de/freiesMagazin-2011-08>
- [15] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/enumitem.html> 
- [16] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/gridset.html>
- [17] <http://www.ctan.org/pkg/fp> 
- [18] <http://dante.ctan.org/tex-archive/help/Catalogue/entries/listings.html> 

Autoreninformation

Dominik Wagenführ (Webseite) ist Chefredakteur von freiesMagazin und hilft seit den Anfangstagen von freiesMagazin das Magazin in LaTeX zu setzen.

 [Teilen](#)

 [Kommentieren](#)

$$\begin{aligned}
 4) \quad 3 \times 9 &= ? \\
 &= 3 \times \sqrt{81} = 3\sqrt{81} = 3\sqrt{\overset{27}{81}} = 27 \\
 &\quad \quad \quad \begin{array}{r} 6 \\ 21 \\ 21 \\ \hline 0 \end{array}
 \end{aligned}$$

“3x9” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/759/>

Statische Webseiten mit Pelican erstellen von Sujeevan Vijayakumaran

Heutzutage werden viele Webseiten im Internet dynamisch generiert, seien es Nachrichten-Seiten, Social Networks oder auch Blogs. Statische Webseiten gehören eigentlich schon fast der Vergangenheit an. Ein Vertreter um statische Blogs zu generieren ist Pelican.

Statisch?

Webseiten werden entweder dynamisch oder statisch erzeugt. Dynamische Webseiten findet man quasi an jeder Ecke des Internets, seien es Blogs, die mit Content-Management-Systemen wie Wordpress, oder Foren, die mit phpBB laufen. Die einzelnen Seiten werden bei dynamischen Webseiten in der Regel beim Aufruf der Webseite generiert. Als bloßer Besucher einer Webseite merkt man in der Regel keinen großen Unterschied; im Hintergrund passieren dann allerdings doch ein paar Dinge. In Blogs oder Foren werden bei einem Seitenaufruf unter anderem Datenbank-Abfragen getätigt, welche Artikel oder Kommentare aus der Datenbank lesen. Mit diesen Daten wird anschließend auf dem Server der HTML-Code für diese spezifische Seite generiert und dem Besucher zurückgeliefert. Häufig werden dabei die Seiten in den Cache [1] geschrieben, um bei einer gleichen Abfrage die Seite schneller zurückliefern zu können.

Bei statischen Webseiten läuft das ganze anders ab. Die HTML-Seiten werden hierbei nämlich

nicht bei jedem Abruf dynamisch erzeugt, sondern liegen als einzelne HTML-Seiten schon vorab auf dem Server. Wenn man also einen statischen Blog betreiben will, dann schreibt man den Blogpost auf dem heimischen Rechner und erzeugt dort die fertigen HTML-Seiten, welche dann auf den Webserver hochgeladen werden. Pelican kommt genau hier ins Spiel. Mit Pelican ist es möglich, solche statischen Blogs zu erstellen. Statische Blogs haben diverse Vor- und Nachteile. Der wohl größte Vorteil ist, dass statische Blogs sehr schnell sind und nicht von anderen Diensten im Hintergrund, etwa von Datenbanken, ausgebremst werden. Daraus resultiert auch eine geringere Serverlast, da weniger Dienste laufen müssen, weil nur fertige HTML-Dateien zurückgeliefert werden. Ein Webserver benötigt daher beispielsweise kein PHP und MySQL-Modul. Nachteilig bei einem statischen Blog ist, dass man fast keine Interaktion mit Lesern führen kann. Kommentare sind dadurch von Haus aus nicht möglich. Umgehen kann man dies allerdings mit externen Diensten.

Allgemeines und Features

Pelican ist in Python [2] geschrieben und ermöglicht das Generieren von statischen Webseiten. Um den Inhalt einer Webseite zu erzeugen, müssen die einzelnen Seite in einer Auszeichnungssprache vorliegen. Pelican unterstützt drei verschiedene Auszeichnungssprachen: reStructuredText [3], Markdown [4] und AsciiDoc [5].

Installation

Die Installation von Pelican ist über zwei Wege möglich. Entweder man nutzt die Python-Tools **easy_install** oder **pip**, um sich die aktuelle Version zu installieren. Durch das Ausführen des Befehls

```
$ easy_install pelican
```

oder

```
$ pip install pelican
```

wird die aktuellste Version 3.3 von Pelican auf dem System installiert. Alternativ kann man auch Pelican über die Paketverwaltung der Distribution installieren. Unter Ubuntu heißt das Paket beispielsweise **python-pelican**. Der Nachteil ist hierbei, dass man dadurch unter Umständen nicht die aktuellste Version erhält. Im Repository von Ubuntu 14.04 ist allerdings die aktuelle Pelican-Version 3.3 verfügbar.

Erste Webseite anlegen

Nach der Installation kann man recht schnell eine einfache Seite mit Pelican erzeugen, dazu gibt es den Befehl „pelican-quickstart“, der ohne zusätzliche Parameter aufgerufen werden kann. Beim Ausführen des Befehls werden viele Fragen gestellt, die beantwortet werden müssen, um den ersten Blog zu erstellen. Als Beispiel wird hier ein Blog namens „freiesBlog“ angelegt.

```
$ pelican-quickstart
Welcome to pelican-quickstart v3.3.0.

This script will help you create a new Pelican-based website.

Please answer the following questions so this script can generate the files
needed by Pelican.

> Where do you want to create your new web site? [.] freiesBlog
> What will be the title of this web site? Ein freies Blog
> Who will be the author of this web site? svij
> What will be the default language of this web site? [en] de
> Do you want to specify a URL prefix? e.g., http://example.com (Y/n)
> What is your URL prefix? (see above example; no trailing slash) http://example.com
> Do you want to enable article pagination? (Y/n) Y
> How many articles per page do you want? [10] 10
> Do you want to generate a Fabfile/Makefile to automate generation and publishing? (Y/n) Y
> Do you want an auto-reload & simpleHTTP script to assist with theme and site development? (Y/n) Y
> Do you want to upload your website using FTP? (y/N) N
> Do you want to upload your website using SSH? (y/N) Y
> What is the hostname of your SSH server? [localhost] server
> What is the port of your SSH server? [22] 22
> What is your username on that server? [root] svij
> Where do you want to put your web site on that server? [/var/www] /var/www
> Do you want to upload your website using Dropbox? (y/N) N
> Do you want to upload your website using S3? (y/N) N
> Do you want to upload your website using Rackspace Cloud Files? (y/N) N
Done. Your new project is available at /home/sujee/Repositories/freiesBlog
```

Während der Erstellung werden also viele Dinge abgefragt, die direkt beim Beginn relevant sind. Alles lässt sich allerdings auch nachträglich ändern. Die wichtigsten Fragen sind hier der Pfad für die Dateien, der Titel der Webseite, der

Autorenname sowie die Domain. Sehr praktisch ist, dass man direkt vorgeben kann, auf welchem Server die Webseite hochgeladen wird. Egal ob FTP [6] oder SSH [7], beides kann einfach angegeben werden. Im Beispiel wird SSH verwendet.

Pelican hat den Ordner **freiesBlog** erzeugt, in dem alle nötigen Dateien der Webseite liegen. Das sind zum einen der Ordner **content** und zum anderen der Ordner **output**. In **content** liegen die Rohinhalte der Webseite und unter **output** anschließend die fertig generierten HTML-Seiten der Webseite, die auf einen Server hochgeladen werden können.

Weiterhin sind zwei Konfigurationsdateien vorhanden: **pelicanconf.py** und **publishconf.py**. In der Datei **pelicanconf.py** werden eher grundsätzliche Eigenschaften der Webseite konfiguriert, während unter **publishconf.py** die Konfigurationen zum Veröffentlichen der Webseite gespeichert sind. Näheres dazu folgt im Verlaufe des Artikels. Übrig bleiben noch drei Dateien: **fabfile.py**, **Makefile** sowie **develop_server.sh**. Mit dem Python-Tool **Fabric** ist es möglich die **fabfile.py**-Datei zu nutzen. Äquivalent dazu gibt es **Makefile**, welches mit dem **Make**-Kommando genutzt werden kann. Sowohl mit **make** als auch mit **fabric** lassen sich ziemlich ähnliche Dinge erledigen. Aus diesem Grund wird hier grundsätzlich nur auf **make** eingegangen.

Durch

```
$ make html
```

werden die HTML-Dateien aus den angelegten Seiten generiert. Im Hintergrund wird allerdings **pelican** aufgerufen. Sowohl **Makefile** als auch

fabfile.py rufen im Hintergrund den Befehl **pelican** mit diversen Parametern auf. Beide Tools dienen ausschließlich dazu, ein einfaches Interface zur Nutzung zu bieten. Mit **make html** werden ein einziges Mal die HTML-Dateien der Webseite erzeugt; wenn man allerdings häufiger etwas ändert und die Änderungen begutachten möchte, dann bietet es sich an, **make regenerate** aufzurufen. In dem Falle werden Änderungen an den Quelldateien automatisch erkannt und die Webseite wird neu generiert. Noch praktischer ist es, wenn die Webseite mittels eines einfachen HTTP-Webserver über den Browser erreichbar ist. Dafür gibt es den Befehl

```
$ make serve
```

Dieser startet einen Server auf Port 8000, so dass die Webseite unter der URL <http://localhost:8000/> erreichbar ist. Möchte man nun **make regenerate** und **make serve** kombinieren, gibt es den Befehl **make devserver**, bei dem sowohl die HTML-Dateien generiert werden, als auch ein Server gestartet wird. Der Server lässt sich anschließend mit

```
$ ./develop_server.sh stop
```

wieder stoppen.

Mit diesen einfachen Befehlen kann man mit Pelican arbeiten. Wenn man beim Einrichten der Webseite die SSH-Zugänge konfiguriert hat, kann man die Webseite mit **make rsync_upload** ganz

einfach auf den Server hochladen. Die Webseite ist dann sofort online.

Der Inhalt

In Pelican wird zwischen zwei verschiedenen Inhaltstypen unterschieden. Dies sind zum einen Seiten („pages“) und zum anderen Artikel.

Wie zu Beginn dieses Artikels schon erwähnt, lassen sich Artikel in reStructuredText, Markdown oder AsciiDoc schreiben. Exemplarisch als Erläuterung folgt nun ein Artikel in Markdown.

```
Title: Pelican-Beispielartikel in
Markdown
Category: Sonstiges
Date: 20-14-27-04 13:37

# Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur
adipisici elit, sed eiusmod tempor
incididunt ut labore ...
```

Das obige Beispiel zeigt, wie ein Artikel in Markdown aussehen kann. Vor dem eigentlichen Artikel folgt der Header, in dem einige Informationen angegeben werden können. **Title** gibt selbstredend den Titel des Blog-Artikels an. Mit Einträgen in **Category** können Kategorien eingetragen werden, in welchen der Artikel erscheinen soll. Die Kategorien enthalten dann beim Erzeugen der HTML-Dateien den entsprechenden Artikel. Jede Kategorie besitzt zusätzlich einen RSS-Feed. Der dritte Punkt ist hier das Datum, welches das Veröf-

fentlichungsdatum des Blog-Artikels darstellt. Die drei aufgeführten Punkte sind nicht die einzigen; es gibt auch noch einige weitere Daten, die man angeben kann, wie etwa einen speziellen Slug [8], einen Autor oder auch eine Zusammenfassung des Artikels.

Der eigentliche Text enthält in diesem Beispiel lediglich eine Markdown-Syntax, nämlich das **#**. Dies ist das Markdown-Zeichen, welches eine Überschrift der ersten Ebene darstellt. Es kann in Artikeln aber auf die volle Markdown-Sprache [9] zurückgegriffen werden.

Neben Artikeln können auch normale Seiten angelegt werden, wie etwa eine Impressum-Seite. Dazu reicht es, wenn man einen **pages**-Ordner innerhalb des **content**-Ordners erstellt und dort die relevanten Seiten anlegt.

Einstellungen

Die Einstellungsmöglichkeiten von Pelican sind sehr vielfältig. Die Basis-Konfigurationsdatei ist standardmäßig **pelicanconf.py**. Dort sind unter anderem auch die Optionen enthalten, die man beim Anlegen des Blogs spezifiziert hat.

Wenn der Blog bloß von einer einzigen Person befüllt wird, bietet es sich an einen Default-Autor zu setzen. Dafür steht die Option **AUTHOR** zur Verfügung. Weiterhin kann man in **SITEURL** den Seitennamen anpassen, sofern man den ursprünglichen Namen beim Erzeugen des Blogs ändern möchte.

Wenn man von einer anderen Blog-Software wechselt oder besondere Wünsche für die URLs der einzelnen Seiten und Artikel hat, bietet es sich an, die URLs entsprechend zu konfigurieren.

Die URL-Konfiguration kann beispielsweise wie folgt aussehen:

```
ARTICLE_URL = 'blog/{date:%Y}/{date:%m~
}/{date:%d}/{slug}/'
ARTICLE_SAVE_AS = 'blog/{date:%Y}/{date~
:%m}/{date:%d}/{slug}/index.html'
```

Alle Blog-Artikel liegen unter dem Sub-Directory **blog**. Dahinter folgen die Zahlen für das Jahr, Monat, Tag und schlussendlich der Slug. Die resultierende URL sieht also so aus: **/blog/2014/04/27/suchen-und-finden-mit-ack-statt-grep/**. Der Blog-Artikel wird allerdings in einer einzelnen HTML-Datei gespeichert, sodass Pelican für jeden einzelnen Blog-Artikel die Ordner-Struktur erzeugt und dort eine **index.html**-Datei anlegt.

Die vollständigen Einstellungsmöglichkeiten von Pelican findet man in der Dokumentation [10].

Plug-ins

Die Basisfunktionalität von Pelican lässt sich mit Hilfe von Plug-ins erweitern. Hierfür gibt es ein offizielles GitHub-Repository, welches von den Pelican-Entwicklern gepflegt wird. Plug-ins werden aber hauptsächlich von externen Helfern eingereicht. Im Repository [11] kann man stöbern und diverse Plug-ins ausprobieren. So gibt

es beispielsweise Plug-ins zum Erzeugen von PDF-Dateien [12] aus den Artikeln, oder ein Plug-in zum einfachen Einbinden von YouTube-Videos [13].

Themes

Ebenfalls in einem GitHub-Repository [14] finden sich eine größere Anzahl an Themes, die man für seinen Pelican-Blog einsetzen kann. Nachdem Herunterladen des Repositorys reicht es, wenn man ein Theme in der Konfigurationsdatei spezifiziert.

Falls man nicht eines der vorgegebenen Themes nutzen will, kann stattdessen man auch ein eigenes Theme für Pelican entwickeln. Wie das geht, kann man ebenfalls in der Dokumentation [15] nachlesen.

Fazit

Mit Pelican lässt sich schnell und einfach ein statischer Blog aufsetzen und betreiben. Er kommt mit allen Vor- und Nachteilen von statischen Blogs daher. Wer eine sinnvolle Kommentarfunktion vermisst, muss sich mit externen Diensten aushelfen, wie zum Beispiel Disqus [16]. Für bereits existierende Blogs gibt es auch ein Importwerkzeug [17], welches Artikel in das Pelican-Format überträgt. Mit den zahlreichen Konfigurationsmöglichkeiten ist es zudem möglich, seinen Blog bis ins Detail so anzupassen, wie man es möchte.

LINKS

[1] <https://de.wikipedia.org/wiki/Cache>

[2] <https://www.python.org/> 

[3] <http://docutils.sourceforge.net/rst.html> 

[4] <http://daringfireball.net/projects/markdown/> 

[5] <http://www.methods.co.nz/asciidoc/> 

[6] https://de.wikipedia.org/wiki/File_Transfer_Protocol

[7] http://de.wikipedia.org/wiki/Secure_Shell

[8] https://de.wikipedia.org/wiki/Clean_URL

[9] <http://markdown.de/>

[10] <http://docs.getpelican.com/en/3.3.0/settings.html> 

[11] <https://github.com/getpelican/pelican-plugins> 

[12] <https://github.com/getpelican/pelican-plugins/tree/master/pdf> 

[13] https://github.com/kura/pelican_youtube 

[14] <https://github.com/getpelican/pelican-themes> 

[15] <http://docs.getpelican.com/en/3.3.0/themes.html> 

[16] <http://disqus.com/> 

[17] <http://docs.getpelican.com/en/3.3.0/importer.html#import> 

Autoreninformation

Sujeevan Vijayakumaran (Webseite) setzt seit einiger Zeit auf einen statischen Blog. Zuerst nutzte er Blogofile, stieß dann aber auf die bessere Alternative Pelican.

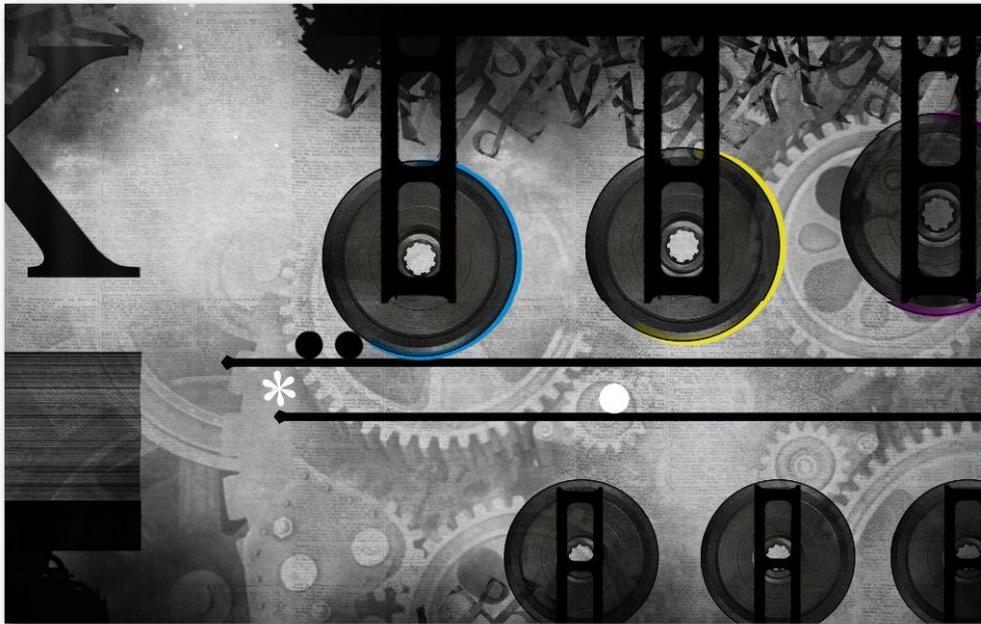
 [Teilen](#)

 [Kommentieren](#)

Typographie lernen mit Type:Rider von Dominik Wagenführ

In einem der letzten Humble Bundle [1] befand sich ein Spiel namens Type:Rider [2]. Da es sich nicht um ein simples Geschicklichkeitsspiel handelt, soll es im Artikel kurz vorgestellt werden.

In Type:Rider versucht man mit zwei rollenden Kreisen (in Anlehnung an einen umgekippten Doppelpunkt) sich seinen Weg durch die einzelnen Levels zu bahnen. Effektiv kann man sich nur mit den Pfeiltasten rollend und mit der Leertaste hüpfend fortbewegen. Dies ist also alles andere als besonders. Besonders ist aber der Levelaufbau.



Der Offsetdruck stört beim Weiterkommen. 

Leveldesign und Aufgabe

Jedes Level besteht aus vier Abschnitten und beschäftigt sich mit einer bestimmten Epoche der Typographie [3]. Von den ersten Höhlenmalereien über die Hieroglyphen zum Buchdruck, der Typenradschreibmaschine und dem Internet. Jedes Level ist dabei nach einer bekannten Schriftart benannt. Es fängt mit Gothic und Garamond an, später trifft man natürlich auch Times und Helvetica und alles findet in Pixel seinen Abschluss.

Die Level sind dabei nicht mit Wiesen und Wäldern angehäuft, wie man das von anderen Geschicklichkeitsspielen gewohnt ist, sondern werden entweder komplett nur aus Buchstaben der jeweiligen Schriftart oder aus bekannten Objekten der jeweiligen Epoche dargestellt. So muss man beispielsweise über die Leitung eines Morseapparates springen oder mit der Type einer Schreibmaschine eine Kugel werfen. Die Macher haben sich dabei ein sehr abwechslungsreiches und



Type:Rider-Titelbild. 

schönes Leveldesign ausgedacht. Ebenso unterlegt die Musik das Spielgeschehen sehr gut.

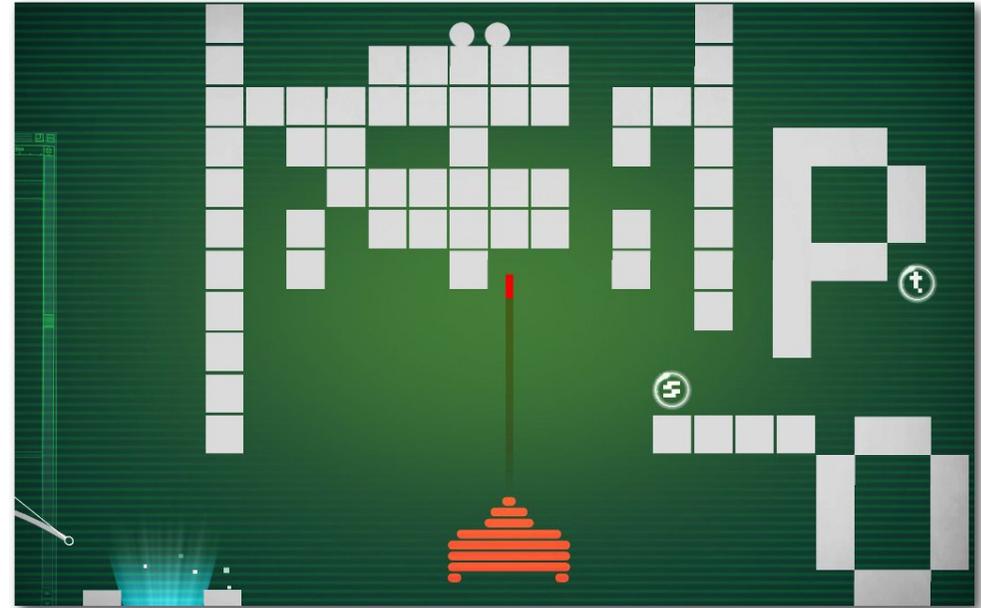
In jedem Level muss man die einzelnen Buchstaben von A-Z aufsammeln. Zusätzlich gibt es noch ein verstecktes Kaufmanns-Und & [4] zu finden. Daneben kann man mehrere Sterne einsammeln, die jeweils einzelne Informationen zur Epoche bieten. So lernt man etwas über verschiedene Druckverfahren oder auch über wichtige Leute hinter den Schriften oder Erfindungen.

Fazit

Wer sich für Typographie interessiert und am besten noch gerne spielt, ist mit Type:Rider gut beraten. Alternativ kann man auch sein Kind spielen lassen und führt sich nur die Hintergrundinformationen später alleine zu Gemüte. Finanziert wurde das Spiel vom europäischen Kultursender arte [5].



Der 20er-Jahre-Film „Metropolis“ bekommt auch einen Platz im Spiel. 🔍



Das Pixel-Level enthält viele Computerspiel-Anspielungen. 🔍

Leider gibt es das Spiel nicht mehr DRM-frei im Humble Store zu kaufen, sondern nur auf Steam [6]. Bei Publisher Bulkypix [7] findet man auch eine Mobilversion für Android und iOS.

Die Sprache des Spiels ist dabei Englisch, was vor allem für die zahlreichen Hintergrundinformationen etwas schade ist. Die Spieldauer ist ungefähr zwei bis drei Stunden, sodass sich Type:Rider auch für zwischendurch gut eignet.

LINKS

- [1] <https://www.humblebundle.com/> 🇬🇧
- [2] <http://typerider.arte.tv/>
- [3] <https://de.wikipedia.org/wiki/Typographie>

- [4] <https://de.wikipedia.org/wiki/Kaufmanns-Und>
- [5] <http://www.arte.tv/de>
- [6] <http://store.steampowered.com/app/258890>
- [7] <http://www.bulkypix.com/game/typerider> 🇬🇧

Autoreninformation

Dominik Wagenführ (Webseite)

spielt sehr gerne unter Linux und interessiert sich auch noch für Typographie, weswegen ihm das Spiel sehr gelegen kam.

✉ Teilen

💬 Kommentieren



IF YOU REALLY HATE SOMEONE, TEACH THEM TO RECOGNIZE BAD KERNING.

“Kerning” © by Randall Munroe
(CC-BY-NC-2.5), <http://xkcd.com/1015/>

Planet Explorers von Marc Volker Dickmann

Spiele, die das besonders erfolgreiche Spiel Minecraft als Vorbild haben, gibt es wie Sand am Meer. Das sich noch in der Entwicklung befindende Spiel Planet Explorers [1] ist jedoch um Längen komplexer und bietet zahlreiche zusätzliche Funktionen.

Minecraft [2] ist ein Spiel, das nach dem Sandkastenprinzip [3] funktioniert. Bei diesem Spielprinzip kann der Spieler selbst bestimmen, was er gerade machen möchte. Dazu kann der Spieler die Welt selbst gestalten und nutzt dabei Blöcke, die als Objekt im Spiel auftreten.

Da die gesamte Welt aus solchen verschiedenen Blöcken besteht, kann der Spieler Blöcke in der



Das Sandbox-Spiel „Minecraft“ besteht aus vielen Klötzchen.

© Kenming Wang (CC-BY-SA 2.0) 

Welt abbauen und wieder hinsetzen und so eine eigene Welt erschaffen. Dabei werden nur wenige Ziele vorgegeben und das Spiel ist unendlich lange spielbar.

Planet Explorers erinnert zur Zeit an einen Massively Multiplayer Online Role-Playing [4]-ähnlichen Ansatz wie in Minecraft.

Geschichte

Planet Explorers bietet neben einem Bau- und einem Adventure- auch einen Story-Modus an.

Der Spieler bekommt nach der Erstellung seines Charakters erst einmal ein kleines Einleitungsvideo zu sehen. In diesem wird der Absturz von Raumschiffen auf einem fremden Planeten gezeigt. Der Spieler erwacht danach in der Nähe eines der abgestürzten Raumschiffe und redet erst einmal mit einer Frau, die ebenfalls in dem gestrandeten Raumschiff anwesend war.

Einem klassischen Rollenspiel ähnlich sind die Quests Aufgaben, die der Spieler lösen muss. Dabei gibt es neben den Hauptquests auch einige weitere

Untermissionen, die es zu lösen gilt. Sollte der Spieler einmal einen Dialog starten, der zu einer Hauptquest führt, wird zuvor nachgefragt, ob die aktuelle Nebenmission abgebrochen werden soll.

Hauptsächlich bestehen die Nebenmissionen aus dem Zusammensuchen von verschiedenen Rohstoffen oder dem Kombinieren solcher zu neuen Gegenständen. Eine Hauptmission kann dagegen zum Beispiel darin bestehen, eine bestimmte Person an einen Ort zu eskortieren. Mit den Hauptmissionen geht die Geschichte weiter und häufig wird der Spieler dabei auch an einen neuen Ort geführt.

Diese einleitende Geschichte ist interessant und führt den Spieler auch ausgezeichnet in die Steuerung und Benutzung des Spiels ein.

Adventuremodus

Im sogenannten Adventuremodus geht es einzig um das Erkunden der Welt. Hierzu müssen die Rohstoffe vom Spieler selbst abgebaut werden.

Der Spieler startet hier in einem Dorf, in dem schon einige Nicht-Spieler-Charaktere (NPC) sesshaft sind. Die NPCs haben in der Regel ein eigenes Haus, in dem sie residieren. Der Spieler kann die Häuser nach Belieben betreten und zum Beispiel abbauen.



Nach dem Absturz erblickt der Spieler eine neue Welt. 🔍

Gefahren stellen hier insbesondere die zahlreichen Kreaturen dar, die in der Welt herumstornern. Besonders große Feinde greifen das Dorf auch an und können die Häuser der NPCs zerstören. Damit der Spieler nicht alleine gegen die Monster kämpfen muss, helfen die NPCs dem Spieler und bekämpfen auch selbst Feinde.

Stirbt der Spieler einmal, kann das Spiel mithilfe des Einsatzes von bestimmten Gegenständen fortgesetzt werden. Gleiches gilt auch für NPCs,

die mit einem Rechtsklick wiederbelebt werden können.

Einige Personen betreiben einen Shop. Als Währung gilt dort rohes Fleisch, das der Spieler durch Verkaufen von Gegenständen oder durch Töten von Tieren bekommen kann.

Baumodus

Wer keine Ressourcen abbauen möchte, kann sich am Baumodus erfreuen. Der Baumodus bie-

tet unbegrenzte Vorräte an Rohstoffen. So können eindrucksvolle Gebäude erschaffen werden, für die man außerhalb dieses Modus wohl Jahre gebraucht hätte.

Das Spiel ermöglicht über den in allen Spielmodi verfügbaren Editor auch die Erschaffung eigener Objekte. Dies können Waffen, Fahrzeuge, Raumschiffe und vieles mehr sein. Dabei gibt es auch die Möglichkeit, vorgefertigte Teile zu benutzen. Einige Teile sind auch für besondere Objekte wie beispielsweise ein Fahrzeug notwendig.

Ist ein Objekt einmal fertig, kann es in einer Datei gespeichert werden und sogar in die Welt eingefügt werden. Dafür müssen sich außerhalb des Baumodus die vorgegebene Anzahl an Rohstoffen im Inventar des Spielers befinden.

Gestaltung

Die Gestaltung der Spielwelt ist recht gut gelungen und bietet eine Vielzahl an unterschiedlichen Objekten. Die natürlichen Objekte wie Bäume sind zwar realitätsfremd, sind aber als Bäume noch erkennbar und fügen sich recht gut in die Umgebung ein, sodass die Welt wie aus einem Guss wirkt.

Für die Bedienoberfläche wird eine an Science-Fiction angelehnte Gestaltungsweise verwendet. Das Gleiche gilt für die Bekleidung mancher Personen. Gleichzeitig erinnert die Oberfläche der Bedienelemente auch etwas an die Gestaltung von The Sims 3.

Steuerung

Die Steuerung funktioniert generell ganz gut, auch wenn noch der eine oder andere Bug vorhanden ist.

Das Spiel unterstützt zwar Controller wie ein Xbox 360 Gamepad, jedoch fehlen noch die passenden Konfigurationen. Wer trotzdem nicht auf ein Pad verzichten möchte, kann über die Einstellungen selbst die entsprechenden Tastenbelegungen festlegen.

Zum aktuellen Zeitpunkt lässt sich Planet Explorers mit Maus und Tastatur jedoch noch am besten spielen.

Grafik

Grafisch sieht Planet Explorers dank Voxel-Technik [5] recht gut aus. Die Farben sind satt und bieten dem Spieler ein anschauliches Bild von der Welt und den Objekten.

Die Effekte sind ebenfalls sehr gut gelungen. Besonders fällt einem hier der Übergang zwischen dem Sichtfeld und dem Nichts auf. Die Welt scheint dabei in der Ferne zu verschwimmen, wodurch ein sehr realistischer Eindruck vermittelt wird.

Die Wetterlage spiegelt sich nicht nur in der Landschaft sondern auch in dem gelungenem Himmel wider. Dieser weist auch realitätsnahe Wolken auf. Für diese Darstellung ist jedoch ein recht guter PC Voraussetzung.



NPCs stehen dem Spieler helfend zur Seite. 

Informationen und Details

Das Spiel wird von dem Entwicklerstudio Panthea Games entwickelt und seit dem 11. März 2014 auch auf Steam [6] vertrieben. Dort kann es für derzeit 22,99 € (Stand Mai 2014) als Early-Access-Version bezogen werden und steht für die Plattformen Linux, MacOS X und Windows bereit. Neben der Möglichkeit alleine zu spielen, gibt es auch einen Mehrspieler-Modus, in dem die Welt gemeinsam mit anderen Spielern erkundet werden kann.

Der Soundtrack ist passend und lässt auch ein längeres Spielen ohne Abschalten der Musik zu. Auch die Geräusche überzeugen und passen zum jeweiligen Objekt.

Als Engine setzt Planet Explorers die bei Linux-Spielen inzwischen häufig benutzte Unity-3D-Engine [7] ein.

Mit dem Kauf der aktuellen Version erhält der Spieler auch alle Updates und spätere Versionen

kostenlos, sodass er das Spiel nach der Veröffentlichung nicht noch einmal kaufen muss. Das ist bei Spielen wie diesem wichtig, da hier häufiger Updates für neue Gegenstände und Fehlerbehebungen kommen werden.



Eine aufgeräumte Oberfläche zeigt wichtige Informationen zum Spieler an. 

Für zukünftige Versionen von Planet Explorers sind schon einige Features angekündigt worden. Es soll beispielsweise später möglich sein, einen Krieg mit Aliens zu führen oder Missionen auf den Monden zu lösen. Auch ein Editor, mit dem sich eigene Geschichten erstellen lassen, ist bereits angekündigt. Einen Überblick über geplante Features findet der Spieler auf Steam.

Eine Besonderheit, die auch die relativ hohen Hardware-Anforderungen erklärt, ist der Einsatz der Voxel-Technik. Diese ermöglicht das Anpas-

sen jedes Objektes und der Welt bis ins kleinste Detail.

Eine Grafikkarte wie NVIDIA kann über die Einstellungen auch zum Rechnen genutzt werden. Dabei wird das vom Hersteller NVIDIA zur Verfügung gestellten Werkzeug CUDA [8] eingesetzt.

Die Anforderungen an das Linux-System sind im Detail:

- OS: Linux Kernel 3.2+
- Prozessor: i5+
- Arbeitsspeicher: 6 GB RAM
- Grafikkarte: NVIDIA GTX 670+, AMD HD7000+
- Netzwerk: Breitband-Internet
- Speicher: 8 GB
- Zusätzlich: OpenCL-kompatible CPU oder GPU (DX10.1)

Fazit

Planet Explorers bietet einen großen Umfang, eine recht interessante Geschichte und weist eine gute Grafik auf. Doch der Preis ist keine Summe für einen Zwischendurch-Kauf, sodass sich das Spiel eher für Fans von Sandbox-Spielen empfiehlt. Auch ist an einigen Stellen zu merken, dass das Spiel noch in der Entwicklung steckt.

Die verschiedenen Spielmodi bringen Abwechslung und ermöglichen eine nahezu unbegrenzte Spieldauer.

Insgesamt befindet sich Planet Explorers auf einem guten Weg und ist am Ende sicherlich auch für einige Spieler durchaus interessant, die sonst keine besondere Leidenschaft für Sandbox-Spiele besitzen.

LINKS

- [1] <http://planetexplorers.pathea.net/> 
- [2] <https://minecraft.net/> 
- [3] <https://de.wikipedia.org/wiki/Open-World-Spiel>
- [4] https://de.wikipedia.org/wiki/Massively_Multiplayer_Online_Role-Playing_Game
- [5] <https://de.wikipedia.org/wiki/Voxel>
- [6] <http://store.steampowered.com/news/12611/?l=german>
- [7] [https://de.wikipedia.org/wiki/Unity_\(Spielengine\)](https://de.wikipedia.org/wiki/Unity_(Spielengine))
- [8] <https://de.wikipedia.org/wiki/CUDA>

Autoreninformation

Marc Volker Dickmann ([Webseite](#)) nutzt privat seit vielen Jahren ausschließlich Linux-Systeme. Er beschäftigt sich insbesondere mit Spielen, die einen neuen und anderen Ansatz bieten und kann dabei auf viele Jahre Spielerfahrung zurückgreifen.

 [Teilen](#)

 [Kommentieren](#)

Retten von verlorenen Daten mit PhotoRec von Hans-Joachim Baader

Das Programm PhotoRec kann versehentlich gelöschte Dateien, aber auch Dateien von defekten Datenträgern retten. Anders als sein Name suggeriert, ist es nicht auf Bilder beschränkt, sondern für Dateien aller Art geeignet.

Redaktioneller Hinweis: Der Artikel „Retten von verlorenen Daten mit PhotoRec“ erschien erstmals bei Pro-Linux [1].

Ein fehlgeschlagenes Experiment mit Btrfs und einem Dateisystem-Cache mit dm-cache resultierten im unerwarteten Verlust einer rund 2,2 TB großen, gut gefüllten Partition. Das Löschen des Caches, als die Daten noch nicht in das Btrfs-Dateisystem geschrieben waren, stellte sich als grober Fehler heraus. Das Resultat war ein so gründlich zerstörtes Btrfs, dass sich selbst mit den Werkzeugen **btrfs-debug-tree** und **btrfs-find-root** nichts mehr rekonstruieren ließ.

Natürlich sollte man keine wertvollen Daten auf experimentellen Dateisystemen halten. Daher waren rund drei Viertel der Daten, darunter die virtuellen Maschinen, gesichert und leicht wiederherzustellen. Der Rest war ersetzbar. Für einige Audio-Dateien, die zur Bearbeitung hier zwischengespeichert wurden, sollte aber einmal PhotoRec auf die Partition losgelassen werden, um vielleicht etwas Arbeit bei der Wiederherstellung zu sparen.

Einleitung

PhotoRec [2] wurde, wie der Name andeutet, ursprünglich in erster Linie zum Retten von gelöschten Bildern aus dem Speicher von Digitalkameras geschrieben. Es kann inzwischen aber auch viele andere Dateitypen wiederherstellen, einige hundert sind bereits eingebaut.

PhotoRec macht sich zunutze, dass die Daten oftmals lückenlos hintereinander geschrieben werden, zumindest auf Linux-Dateisystemen. Es sucht daher Sektor für Sektor nach bekannten Datei-Headern. Erkennt es einen solchen, versucht es, die ganze Datei aus den anschließenden Sektoren auslesen. Es agiert völlig unabhängig vom Dateisystem und kommt daher mit allen Dateisystemen zurecht, auch mit zerstörten, neu formatierten usw. Auch versehentlich gelöschte Dateien kann PhotoRec wieder herstellen.

Installation

PhotoRec kann im Quellcode oder als Bestandteil von Testdisk [3] heruntergeladen werden. Den Quellcode sollte man mit folgender Sequenz kompilieren können, wie auf der Entwicklerseite [4] von Testdisk erklärt wird:

```
$ cd testdisk
$ mkdir config
$ autoreconf --install -W all -I config
$ ./configure
$ make
```

Das scheiterte im Test zwar am **configure**, das mit einer Fehlermeldung endete, aber es gibt auch Binärpakete von Testdisk [5]. Entpackt man dieses Paket, so kann man PhotoRec installieren, aber es geht auch ohne Installation.

```
$ cd testdisk-6.14
$ ./photorec_static
```

Bedienung

Das Programm bringt eine Terminal-Oberfläche, in der man zunächst die Partition oder Image-Datei auswählt, die man durchsuchen will. Dann wählt man, ob das Dateisystem zur Familie ext2/ext3/ext4 gehört oder ein anderes ist. Nachdem man noch das Zielverzeichnis gewählt hat, das ausreichend Platz bieten muss, kann man die Suche starten. Gefundene Dateien werden in den Verzeichnissen **recup_dir.0**, **recup_dir.1** usw. abgelegt. Nach jeweils 1000 gefundenen Dateien legt PhotoRec ein neues Verzeichnis mit einer um 1 erhöhten Nummer an.

Die Suche kann auf einem Medium im Terabyte-Bereich einige Tage in Anspruch nehmen; bei den für Kameras usw. gängigeren Größen dauert sie von einigen Minuten bis zu etwa einer Stunde. Gefundene Dateien erhalten von PhotoRec eine Endung, die dem erkannten Dateityp entspricht. Den Dateinamen oder gar die Verzeichnishierarchie kann PhotoRec allerdings nicht mehr herstellen, da es keine Inodes [6] analysiert.

Dementsprechend muss man die gefundenen Dateien von Hand überprüfen und ihnen wieder einen sinnvollen Namen geben. Bei Bildern und Videos ist die Identifikation offensichtlich: Man schaut sie sich an. Audio-Dateien kann man in einem Player abspielen. Um zu prüfen, ob Audio-dateien fehlerfrei sind, kann man sie in Audacity öffnen. Dort sieht man korrupte Daten auf einen Blick. Sind die Daten beschädigt, hat man Pech gehabt, eine Abhilfe ist in diesem Fall leider nicht mehr möglich.

Eine kleine Orientierung bietet dabei der erste Buchstabe des Dateinamens: PhotoRec extrahiert auch verkleinerte Vorschaubilder, wo möglich, und speichert diese unter Namen ab, die mit **t** beginnen, beispielsweise **t7228086.jpg**. „Normale“ Dateien beginnen mit **f**. Auch das Erstellungsdatum der Datei wird unter Umständen wiederhergestellt; zumindest auf einem FAT-Dateisystem klappte das. Auf anderen Dateisystemen geht das nicht, da die Information wie der Dateiname im möglicherweise nicht mehr vorhandenen Inode gespeichert ist.

Testweise wurde PhotoRec auf die 4 GB große CF-Karte einer Kamera angewendet. Das ergab über 1000 gefundene Bilder, die ältesten davon fast sechs Jahre alt. Stichproben zeigten, dass die Bilder größtenteils unbeschädigt waren. Eine vollständige Historie aller Aufnahmen ergab sich allerdings nicht, da zweifellos zwischenzeitlich viele Bilder überschrieben worden waren. Doch wenn man nur die neuesten Bilder retten will, was der

Normalfall sein dürfte, hat man ausgezeichnete Chancen.

Anders sieht es bei stark beschädigten Dateisystemen aus, wie dem eingangs erwähnten. Dabei war die Suche nur mäßig erfolgreich. Nach fünf Tagen Suche waren 7,5 Mio. Dateien in knapp über 15.000 Verzeichnissen extrahiert. Trotz dieser Datenflut war es ein Leichtes, die für den Test relevanten .wav-Dateien darin zu finden, da ihre Zahl sich auf ein paar Dutzend belief. Viele davon waren jedoch korrupt und nicht mehr brauchbar. Der Grund für den geringen Erfolg dürfte in erster Linie die Größe der Dateien (um 400 MB) sein, die eine sequentielle Abspeicherung ohne Lücken unwahrscheinlich macht.

Die Definition neuer Dateiformate, die PhotoRec erkennen soll, ist leider anders als bei **file** nicht mit Konfigurationsdateien gelöst, sondern im Quellcode beschrieben. Daher ist eine Erweiterung zwar möglich, erfordert aber etwas C-Programmierkenntnisse.

Fazit

PhotoRec ist ein hervorragendes Programm, um einen letzten Versuch zu unternehmen, verlorene Daten zu retten. Dabei muss man zwar mit der Einschränkung leben, dass es die Original-Dateinamen nicht rekonstruieren kann, aber in vielen Fällen kann es fast Wunder wirken. Gute Chancen hat man bei kleinen Dateien wie zum Beispiel JPEG-Bildern, da es prinzipbedingt um so besser funktioniert, je kleiner die Dateien sind.

Wer Dateien wiederherstellen will, die nicht auf Flash-Medien liegen, sollte es zuerst mit spezialisierten, auf das jeweilige Dateisystem zugeschnittenen Werkzeugen versuchen. Erst wenn das nicht möglich ist, sollte als letztes Mittel der Wahl PhotoRec zum Einsatz kommen.

Der Entwickler von PhotoRec nimmt auch Spenden entgegen und es dürfte eine Menge dankbarer Benutzer geben, die hier gerne etwas beitragen.

LINKS

- [1] <http://www.pro-linux.de/artikel/2/1680/retten-von-verlorenen-daten-mit-photorec.html>
- [2] <http://www.cgsecurity.org/wiki/PhotoRec> 
- [3] <http://www.cgsecurity.org> 
- [4] <http://www.cgsecurity.org/wiki/Developers> 
- [5] http://www.cgsecurity.org/wiki/TestDisk_Download 
- [6] <https://de.wikipedia.org/wiki/Inode>

Autoreninformation

Hans-Joachim Baader ([Webseite](#))

befasst sich seit 1993 mit Linux. 1994 schloss er erfolgreich sein Informatikstudium ab, machte die Softwareentwicklung zum Beruf und ist einer der Betreiber von Pro-Linux.de.



Teilen



Kommentieren

Rezension: Geheimnisse eines JavaScript-Ninjas von Jochen Schnelle

Bücher zum Thema JavaScript gibt es viele. Solche, die auch fortgeschrittene Themen behandeln, gibt es schon weniger. Und Bücher, die das Zeug zum Klassiker haben, sind rar. Das vorliegende Buch „Geheimnisse eines JavaScript-Ninjas“ hat dabei durchweg das Potential, ein Klassiker zu werden.

Redaktioneller Hinweis: *Wir danken dem mitp-Verlag für die Bereitstellung eines Rezensionsexemplares.*

Einer der Autoren des Buchs ist John Resig, Schöpfer und ehemaliger Hauptentwickler der beliebten und viel eingesetzten JavaScript-Bibliothek jQuery [1] – gute Voraussetzungen also für ein fundiertes Buch zum Thema JavaScript.

Zielgruppen

Die Zielgruppe des Buchs ist ziemlich weit gespannt, vom fortgeschrittenen Anfänger bis hin zum „Programmierprofi“. Dies wird dadurch erreicht, dass der gebotene Inhalt im Verlauf des Werks immer fortgeschrittener wird. Programmierer mit tiefergehenden JavaScript-Kenntnissen könnten dabei die ersten Kapitel schnell lesen oder ganz auslassen und kommen dann später auf ihre Kosten, während fortgeschrittene Einsteiger das Buch wohl (erst mal) nicht zu Ende lesen, sondern erst zu einem späteren Zeitpunkt wieder darauf zurück greifen werden.

Viergeteilter Inhalt

Das Buch ist in vier Sektionen unterteilt. In der ersten Sektion werden ein paar grundlegende Dinge er- und geklärt. Beispielsweise, wie die im Verlauf des Buchs zu findenden Codebeispiele mit Hilfe von „Assertions“ (auf Deutsch: Zusicherung, Sicherstellung) getestet werden. Wovon die Autoren auch von Anfang bis Ende regen Gebrauch machen.

Der zweite Teil, der ca. 45% des 468 seitigen Buchumfangs ausmacht, beschäftigt sich mit JavaScript-Funktionen, Closures, Objektorientierung und Prototypen, regulären Ausdrücken, Threads und Timer. In diesen Kapiteln wird eine fundierte Wissensbasis für den effizienten Einsatz von JavaScript gelegt.

Der dritte und vierte Teil behandeln dann fortgeschrittene Themen wie Codeauswertung zur Laufzeit, Umgang mit CSS, DOM-Manipulation, Ereignis-Behandlung sowie Cross-Browser-Strategien. Zu letzteren gibt es ein eigenes Kapitel, wobei das Thema auch immer wieder in den diversen anderen Kapiteln aufgegriffen wird.

Praxisnahe Beispiele

Über das ganze Buch hinweg verteilt findet der Leser viele Beispiele, die oft sehr praxisnah und anschaulich sind. Teilweise ist der Code den JavaScript-Bibliotheken jQuery und Prototype entnommen. Alle enthaltenen Beispiele sind gut kom-

mentiert und werden im anschließenden Text nochmals fundiert besprochen und erläutert. Des Weiteren sind alle Listings vollständig abgedruckt, so dass diese auch „offline“ problemlos nachvollziehbar sind.

Zu bemängeln ist hier nur eine Kleinigkeit, nämlich dass die neben den Listings abgedruckten Erklärungen des Codes teilweise schwer zu lesen sind. Dies liegt daran, dass die Schrift recht klein ist und auch statt in schwarz in einem Mittelgrau gedruckt wurde, was zu recht schwachen Kontrasten führt.

Sprache und Übersetzung

Die Übersetzung des Buches ist sehr gut gelungen. Es liest sich sehr flüssig und ist gut verständlich. Prinzipiell kann man den Text auch schnell lesen, falls nicht das Verständnis des Inhalts im Weg steht.

An einigen Stellen im Buch finden sich auch Grafiken und Diagramme. Diese sind durchweg in Graustufen abgedruckt. Leider ist es aber wohl so, dass die Originalgrafiken farbig waren. Durch die Reduzierung auf Graustufen haben diese dadurch enorm an Lesbarkeit verloren haben. Hier hätte man etwas mehr Umsicht bei der Umsetzung walten lassen können. Da aber keine der Grafiken für das Verständnis des Inhalts wirklich wichtig ist, ist dies eher ein Schönheitsfehler.

Buchinformationen

Titel	Geheimnisse eines JavaScript-Ninjas [2]
Autor	John Resig, Bear Bibeault
Verlag	mitp-Verlag, 2014
Umfang	468 Seiten
ISBN	978-3-8266-9714-2
Preis	34,99 € (Print), 29,99 € (EPUB, PDF)

Kritik

Wie in der Einleitung schon gesagt: Das Buch hat absolut das Zeug zum Klassiker. Die Autoren geben einen tiefen und sehr fundierten Einblick darin, wie man effizientes und gutes JavaScript programmiert.

Auch der Teil zur Cross-Browser-Programmierung, der im Verlauf des Buchs wiederholt auftaucht, ist hervorragend gelungen. Schon alleine dieser würde den Kauf des Buches rechtfertigen, wenn man eine weite Spanne von Browsern unterstützen will oder muss.

Zusammenfassung

Wer mit JavaScript programmiert und seine Fähigkeiten verbessern möchte, dem sei dieses Buch wärmstens empfohlen. Eigentlich dürfte es in keiner Bibliothek eines JavaScript-Programmierers fehlen. Wer das Buch gelesen hat und den Inhalt in seinem Programmcode umsetzt, der ist in der Tat dem Level eines JavaScript-Ninjas nahe.

Redaktioneller Hinweis: Da es schade wäre, wenn das Buch bei Jochen Schnelle im Regal verstaubt, wird es verlost. Die Gewinnfrage lautet:

„Wie bereits erwähnt ist einer der Buchautoren der Erfinder der JavaScript-Bibliothek jQuery. Im Monat welchen Jahres erschien das erste stabile Release 1.0 von jQuery?“

Die Antwort kann bis zum **14. Juli 2014, 23:59 Uhr** über die Kommentarfunktion oder per E-Mail

an redaktion@freiesMagazin.de geschickt werden. Die Kommentare werden bis zum Ende der Verlosung nicht freigeschaltet. Das Buch wird unter allen Einsendern, die die Frage richtig beantwortet haben, verlost.

LINKS

- [1] <http://www.jquery.org> 
 [2] http://www.it-fachportal.de/shop/buch/Geheimnisse_eines_JavaScript-Ninjas/detail.html,b191209

Autoreninformation 

Jochen Schnelle ([Webseite](#)) interessiert sich generell für Programmiersprachen und verwendet JavaScript für eigene, web-basierte Programme.

 [Teilen](#)

 [Kommentieren](#)



“Stephen Hawking” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/799/>

Rezension: Inkscape – Der Weg zur professionellen Vektorgrafik von Jens Dörpinghaus

Das Buch richtet sich nach eigener Aussage an Einsteiger, die Grundlagen im Umgang mit Inkscape lernen möchten. Der Autor, Uwe Schöler, bezeichnet sich selbst als „begeisterten Open-Source-Anhänger“. Dargestellt wird sowohl die aktuelle, stabile Version, als auch die kommende Version 0.91, wobei sich der Autor der Entwicklerversion bediente.

Redaktioneller Hinweis: *Wir danken dem Hanser-Verlag für die Bereitstellung eines Rezensionsexemplares.*

Was steht drin

Nach einer kurzen Einleitung, in der nicht nur Hinweise zum Text selbst stehen, sondern auch Informationen zu Inkscape und dem SVG-Format gegeben werden, folgt direkt ein Kapitel „Schnelleinstieg“. Dort werden zunächst die wichtigsten Grundlagen beschrieben, also die Programmoberfläche, sowie Formen und Werkzeuge von Inkscape. Der eigentlichen Programminstallation wird allerdings für die drei Betriebssysteme Windows, Linux und Mac OS X nur insgesamt eine knappe Seite spendiert. Abgeschlossen wird dieses Kapitel mit einem „Workshop“, der die möglichen ersten Schritte im Umgang mit Inkscape beschreibt. Es folgen später noch zwei weitere Workshops in anderen Kapiteln, in denen jeweils ein Schneemann erstellt und erweitert wird und dabei das Gelernte angewendet wird.

Es folgt das Kapitel „Inkscape-Grundlagen“. Hier werden im Wesentlichen nur die Werkzeuge beschrieben, sogar fein nach Rechteck, Ellipse und Kreis, Polygon und Stern sowie Spirale aufgeteilt. Dieses Kapitel bietet viele Hintergrundinformationen, zum Beispiel auch zum zugrundeliegenden XML-Quellcode.

Nach einem kurzen Abschnitt über „Farbmanagement“, in dem nicht nur verschiedene Farbraummodelle dargestellt werden, sondern interessanterweise auch Farbverläufe, Muster und Konturmuster, folgt eine ausführliche Darstellung der „Pfade und Bearbeitungswerkzeuge“ und der „Textbearbeitung“. Beide Kapitel sind sehr ausführlich und beinhalten auch kurze Schritt-für-Schritt-Anleitungen.

Mit dem Abschnitt „Ausschneidungen, Masken, Ebenen und Tipps“ wird der allgemeine Praxisbereich abgeschlossen. Hier finden sich zunächst drei kurze Abschnitte, während der Abschnitt zu Tipps und Tricks länger ausfällt.

Nun folgt nach einem Kapitel „Webseiten mit Inkscape erstellen“ ein Kapitel „Filter“. Dieses listet nach einer kurzen Einführung in diese Thematik im Wesentlichen nur Filter mit einer äußerst knappen Anleitung auf. Ein kurzes Kapitel widmet sich der „Benutzerkonfiguration“ – interessanterweise werden hier die Anwendungseinstellungen und Dokumenteinstellungen zusammengefasst –, wo-

rauf eine längere Auflistung von „Erweiterungen“ folgt. Der Abschnitt „Der Umgang mit Dateien“, gefolgt vom Anhang mit Glossar, Tastenkombinationen und einer weiteren Auflistung von Filtereffekten schließen das Buch ab.

Wie ist es geschrieben?

Die einzelnen Kapitel sind sehr unterschiedlich gestaltet. Einige sind ausführlich und motivieren schon beim Lesen, die Dinge auszuprobieren. Andere sind eher trocken und haben eine enorme Informationsdichte.

An wen richtet es sich?

Laut eigener Aussage richtet sich dieses Buch an Anfänger. Ein Blick auf das Inhaltsverzeichnis lässt allerdings stutzen: Themen wie Webseiten-erstellung haben in einem Anfängerbuch dieses Umfangs sicherlich wenig Sinn – und leider merkt man das diesen Kapiteln auch anhand der Qualität an.

Für professionelle Anwender und insbesondere Anwender, die nicht bei Null anfangen, sondern schon erste Erfahrungen gesammelt haben, bietet das Buch wohl zu wenig Neues. Weiterführende Informationen zu bestimmten Themen finden sich in übersichtlicher Form frei verfügbar im Internet.

Da das Niveau im Buch selber stark schwankt, ist es nur für solche Anfänger geeignet, die bereit sind, ein paar Seiten zu überschlagen, bevor für

sie wieder relevante Informationen kommen. Da aber alle grundlegenden Bereiche dargestellt werden und darüber hinaus noch weitere Kenntnisse vermittelt werden, sollte ein Anfänger nach der Lektüre gut mit Inkscape umgehen können.

Fazit

Insgesamt macht das Buch vom Äußeren her einen guten Eindruck. Dennoch gibt es einige typographische Schwächen, so sind insbesondere Kapitel mit kurzen Textabschnitten und vielen Screenshots sehr unruhig gesetzt und schwer zu lesen. Auch sind die Markierungen in Screenshots – per se ja eine gute Idee – mit gelben Farben im Stil eines Textmarkers sehr irritierend.

Die pädagogischen Schwächen fallen allerdings weit mehr ins Gewicht. Ist der erste Workshop noch statt einer Anleitung zum Selbermachen eher eine Beschreibung des ersten Selbermachens, so nimmt die Qualität im Folgenden eher zu. Das systematische Erstellen eines Schneemanns ist sicherlich ein nettes Beispiel.

Nicht nur das Niveau, sondern auch die Systematik, Lesbarkeit und die Ausführlichkeit schwanken stark. Es fällt u. a. auf, dass schon im Kapitel „Grundlagen“ der Anfänger – und an ihn richtet sich dieses Werk ja – mit umfangreichem XML-Quellcode ohne größere Erläuterungen beworfen wird. Besonders positiv fällt hingegen das Kapitel 6 „Pfade und Bearbeitungswerkzeuge“ auf, denn es erklärt die Sachverhalte genau und ausführlich und bietet darüberhinaus auch kurze Anleitungen. Als Kontrast kann man das Kapitel 9 betrachten: Dort wird in wenigen Seiten nicht nur versucht, HTML und CSS zu erklären, der Autor versucht hier auch sein eher eigenwilliges Konzept von guter Webseitengestaltung kompakt zu erklären.

Es ist dem Autor aber an jeder Stelle des Buches sein guter Wille anzumerken und bei aller Kritik findet dieses Buch sicherlich seine Leser. Denn es ist mit Abstrichen eine gelungene Einführung für Anfänger in die Anwendung Inkscape.

Buchinformationen

Titel	Inkscape – Der Weg zur professionellen Vektorgrafik [1]
Autor	Uwe Schöler
Verlag	Carl Hanser Verlag, 2014
Umfang	326 Seiten
ISBN	978-3-446-43865-1
Preis	29,99 EUR (Druck), 23,99 (PDF)

LINKS

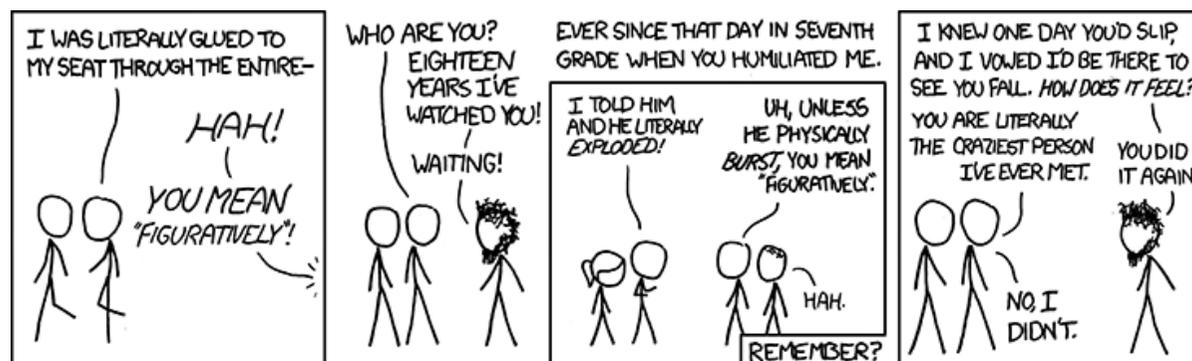
[1] <http://www.hanser-fachbuch.de/buch/Inkscape/9783446438651>

Autoreninformation

Jens Dörpinghaus arbeitet ausschließlich mit freier Software, auch im kreativen Bereich. Deshalb kamen ihm einige Informationen aus dem Buch gerade recht.

 [Teilen](#)

 [Kommentieren](#)



“Literally” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/725/>

Leserbriefe

Für Leserbriefe steht unsere E-Mailadresse redaktion@freiesMagazin.de zur Verfügung – wir freuen uns über Lob, Kritik und Anregungen zum Magazin.

An dieser Stelle möchten wir alle Leser ausdrücklich ermuntern, uns auch zu schreiben, was nicht so gut gefällt. Wir bekommen sehr viel Lob (was uns natürlich freut), aber vor allem durch Kritik und neue Ideen können wir uns verbessern.

Leserbriefe und Anmerkungen

Rezension: Vim in der Praxis

❖ Kommt zum rechten Zeitpunkt. ;) Danke für das Review, ich habe mir nun das EPUB geholt, allerdings in Englisch. Wenn es geht, empfiehlt es sich, sich Bücher immer in der Originalsprache zu Gemüte zu führen.

blizzz

Äquivalente Windows-Programme unter Linux - Teil 5

❖ Vielen vielen Dank für den Hinweis auf QuiteRSS! Ich habe mich dumm und dämlich gesucht nach einer adäquaten Alternative zum Feed Reader von Opera. Überhaupt tolle Artikelserie, weiter so!

Thomas McWork (Kommentar)

GPS: Tracks und Routen erstellen mit QLandkarte GT

❖ Was soll der ganze Wine-Kram in dem Artikel? Sowohl QLandkarte GT, als auch unzählige freie Renderings für Karten, freie Kachel-

server, kommerzielle Renderings/Kachelserver und „Offline-Karten“ für alle möglichen Themengebiete aus allen möglichen Quellen basierend auf allen möglichen Daten gibt es komplett ohne Windows-Zwang (und die daraus resultierende Wine-Verwendung).

Im Artikel wird übrigens fast durchgängig von QLandkarte gesprochen. QLandkarte ist ein völlig anderes Programm als QLandkarte GT.

Und die mehrfach so dargestellte 50-Punkte-Begrenzung bei Routen ist übrigens nicht universell gültig, sondern ist Router-Abhängig. Ich konnte bisher jedenfalls keine Einschränkungen feststellen.

Das über mehrere Spalten hinweg beschriebene, recht umständliche Löschen von Trackteilen geht übrigens wesentlich einfacher, indem man die ungewünschten Punkte einfach markiert und entfernt. QLandkarte GT verbindet den Punkt davor und danach automatisch einfach wieder – ohne, dass man Tracks teilen oder zusammenfügen muss.

Dirk (Kommentar)

❖ *Erstmal Danke für Dein Interesse und die Rückmeldung. Ich will kurz auf die von Dir angesprochenen Punkte eingehen:*

Leider ist es mir nicht möglich nachzuvollziehen, ob „QLandkarte GT“ ein völlig anderes Programm

ist als QLandkarte (ohne den „GT-Zusatz“). Im Netz erfährt man nur, dass QLandkarte GT eine Weiterentwicklung darstellt, die alte Version gibt es nicht mehr. Sicherlich ist die Oberfläche ein bisschen modifiziert worden, natürlich kommen im Laufe der Zeit auch neue Funktionen dazu. Dass ich den „GT-Zusatz“ im Verlauf des Artikels einige male „unterschlagen“ habe, tut mir leid. Ich bin wohl davon ausgegangen, dass dies allein schon durch die Überschrift klar wäre.

Im Artikel steht eigentlich nicht, dass man dazu gezwungen wäre, „Wine“ zu nutzen. Aber jeder (Mountainbiker), welcher nach Karten im Netz sucht wird früher oder später auf die Seite openmtbmap.org [1] stoßen. Und dort sind die Karten nun mal in einer exe-Datei eingebettet. Und ja: Theoretisch kann man die Karten auch via „Zip“ entpacken. So steht es im Internet. Aber bei mir kam es dabei zu einer Fehlermeldung, dagegen funktionierte es mit „wine“. Deshalb habe ich den Weg auch vorgeschlagen.

Über die 50 Punkte Begrenzung bei Routen habe ich im Netz gelesen. Ich habe mir dann selbst eine Route erstellt und bin dann auch selbst auf die genannte Einschränkung gestoßen.

Das Löschen von Trackteilen bei langen Tracks ist umständlich, wenn der Track recht lang ist und viele Punkte zu löschen sind. Deshalb hat wohl der Programmierer auch die Möglichkeit geschaffen,

Tracks zu schneiden und wieder zusammenzufügen. Aber in meinem Artikel bin ich ja auf beide Möglichkeiten eingegangen und habe dies auch jeweils begründet. Natürlich kann man die Punkte einzeln löschen und ja, die Punkte verbinden sich dann auch wieder von selbst. Aber eben auch als Luftlinie quer durch Häuser etc. Dann muss man noch Hand anlegen und bestimmte Wegpunkte wieder auf die Straße legen. Letztlich muss jeder für sich entscheiden, was schneller von der Hand geht, auf jeden Fall ist es gut, dass Du es noch mal herausgestellt hast.

Andreas

WLAN

❖ Ich habe nur herausgefunden, dass es zwei Treiber für diesen Chipsatz gibt. Nur welcher besser ist, weiß ich nicht. Bei einem muss noch eine Firmware geladen werden, die extra installiert werden muss. Wenn WLAN per USB angeschlossen wird, ist es noch einmal ein Extrathema. So ähnlich wie bei einem UMTS-Modem.

Eine andere Möglichkeit ist, dass der Kernel erneuert wurde. Dann kann man aber auch die alten Treiber wieder installieren. Aber bei einem so un-

klaren Fehlerbericht kann ich auch nicht mehr tun.

Matthias Kühmsted (Kommentar)

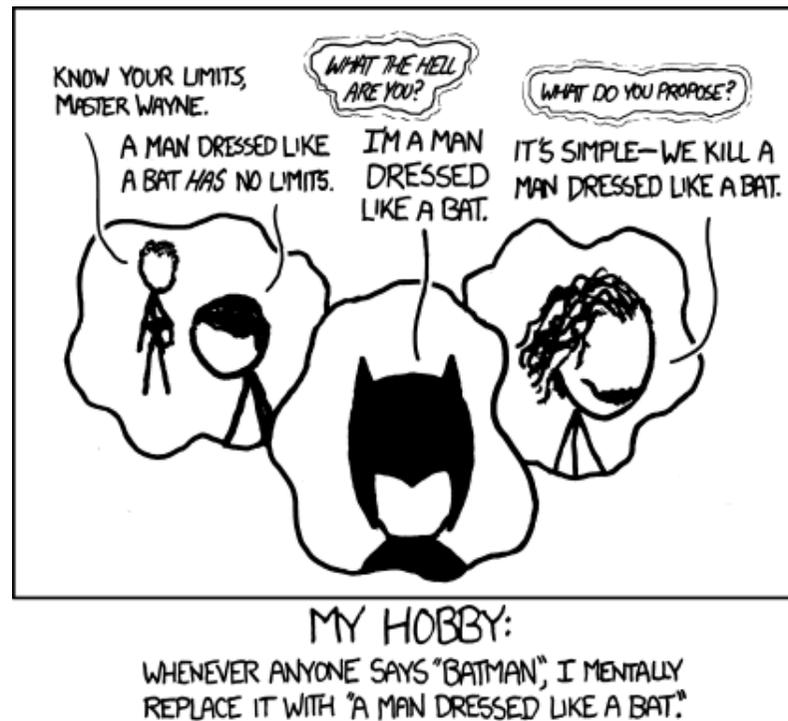
LINKS

[1] <http://www.openmtbmap.org> 

Die Redaktion behält sich vor, Leserbriefe gegebenenfalls zu kürzen. Redaktionelle Ergänzungen finden sich in eckigen Klammern.

 [Teilen](#)

 [Kommentieren](#)



“Batman” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/1004/>

Veranstaltungskalender

Messen

Veranstaltung	Ort	Datum	Eintritt	Link
Maker Faire	Hannover	05.07.–06.07.2014	10 EUR	http://makerfairehannover.com/
LHC V.	Hamburg	18.07.–20.07.2014	frei	https://wiki.attraktor.org/Large_Hackerspace_Convention_5
Linuxbierwanderung	Talybont-on-Usk	20.07.–26.07.2014	frei	http://lbw2014.\T1\th\T1\th.be/
GNU Hackers Meeting	München	15.08.–17.08.2014	frei	http://www.gnu.org/ghm/2014/munich/
FrOSCon	Sankt Augustin	23.08.–24.08.2014	–	http://www.froscon.de/
MRMCD14	Darmstadt	05.09.–07.09.2014	42 EUR	https://mrmcd.net/
Datenspuren	Dresden	13.09.–14.09.2014	frei	http://www.datenspuren.de/
Kieler Linuextage	Kiel	19.09.–20.09.2014	frei	http://www.kieler-linuextage.de/
Software Freedom Day	Weltweit	20.09.2014	frei	http://softwarefreedomday.org/
GUUG FFG2014	Bochum	23.09.–29.09.2014	–	https://www.guug.de/veranstaltungen/ffg2014/
Ubucon 2014	Katlenburg	17.-19.10.2013	–	http://ubucon.de/2014/

(Alle Angaben ohne Gewähr!)

Sie kennen eine Linux-Messe, welche noch nicht auf der Liste zu finden ist? Dann schreiben Sie eine E-Mail mit den Informationen zu Datum und Ort an redaktion@freiesMagazin.de.

Konventionen

An einigen Stellen benutzen wir Sonderzeichen mit einer bestimmten Bedeutung. Diese sind hier zusammengefasst:

- \$: Shell-Prompt
- #: Prompt einer Root-Shell – Ubuntu-Nutzer können hier auch einfach in einer normalen Shell ein **sudo** vor die Befehle setzen.
- ↵: Kennzeichnet einen aus satztechnischen Gründen eingefügten Zeilenumbruch, der nicht eingegeben werden soll.
- ~: Abkürzung für das eigene Benutzerverzeichnis **/home/BENUTZERNAME**
- : Kennzeichnet einen Link, der auf eine englischsprachige Seite führt.
- : Öffnet eine höher aufgelöste Version der Abbildung in einem Browserfenster.

Impressum

freiesMagazin erscheint als PDF, EPUB und HTML einmal monatlich.

Kontakt

E-Mail redaktion@freiesMagazin.de
Postanschrift **freiesMagazin**
c/o Dominik Wagenführ
Beethovenstr. 9/1
71277 Rutesheim
Webpräsenz <http://www.freiesmagazin.de/>

Autoren dieser Ausgabe

Hans-Joachim Baader **S. 32**
Marc Volker Dickmann **S. 28**
Jens Dörpinghaus **S. 36**
Tobias Kaulfuß **S. 5**
Dr. Diether Knof **S. 9**
Mathias Menzer **S. 3**
Jochen Schnelle **S. 34**
Sujeevan Vijayakumaran **S. 22**
Dominik Wagenführ **S. 15, S. 26**

ISSN 1867-7991

Erscheinungsdatum: 6. Juli 2014

Erstelldatum: 11. Juli 2014

Redaktion

Christian Schnell Matthias Sitte
Dominik Wagenführ (Verantwortlicher Redakteur)

Satz und Layout

Jonas Dabelow Moritz Kiefer
Christoph Lehmann Kai Welke

Korrektur

Daniel Braun Frank Brungräber
Vicki Ebeling Stefan Fangmeier
Mathias Menzer Christian Schnell
Karsten Schuldt Toni Zimmer

Veranstaltungen

Ronny Fischer

Logo-Design

Arne Weinberg (CC-BY-SA 4.0 Unported)

Dieses Magazin wurde mit \LaTeX erstellt. Mit vollem Namen gekennzeichnete Beiträge geben nicht notwendigerweise die Meinung der Redaktion wieder. Wenn Sie freiesMagazin ausdrucken möchten, dann denken Sie bitte an die Umwelt und drucken Sie nur im Notfall. Die Bäume werden es Ihnen danken. ;-)

Soweit nicht anders angegeben, stehen alle Artikel, Beiträge und Bilder in freiesMagazin unter der Creative-Commons-Lizenz CC-BY-SA 4.0 International. Das Copyright liegt beim jeweiligen Autor. Die Kommentar- und Empfehlen-Icons wurden von Maren Hachmann erstellt und unterliegen ebenfalls der Creative-Commons-Lizenz CC-BY-SA 4.0 International. freiesMagazin unterliegt als Gesamtwerk der Creative-Commons-Lizenz CC-BY-SA 4.0 Unported mit Ausnahme der Inhalte, die unter einer anderen Lizenz hierin veröffentlicht werden. Das Copyright liegt bei Dominik Wagenführ. Es wird erlaubt, das Werk/die Werke unter den Bestimmungen der Creative-Commons-Lizenz zu kopieren, zu verteilen und/oder zu modifizieren. Die xkcd-Comics stehen separat unter der Creative-Commons-Lizenz CC-BY-NC 2.5 Generic. Das Copyright liegt bei Randall Munroe.