

Topthemen dieser Ausgabe

Administration von Debian & Co im Textmodus – Teil III

Seite 5

Im dritten und damit letzten Teil des insgesamt dreiteiligen Workshops für alle, die sich grundlegende Kenntnisse in der Administration von Debian-basierten Systemen aneignen müssen oder wollen, geht es nun darum, wie man selbst Pakete baut, welche System-Werkzeuge man für den Notfall in der Schreibtischschublade haben sollte, wie man Fehlern auf die Spur kommt, was man tun kann, wenn der Bootmanager nicht mehr will, und wie man verlorene Passwörter neu setzen kann. ([weiterlesen](#))

Arduino

Seite 20

Der Artikel richtet sich an Anfänger, die Mikrocontroller und Arduino unter Linux verwenden wollen. Es werden die Grundlagen von Mikrocontrollern erläutert und die Hard- und Software der Arduino-Plattform vorgestellt. Zwei Beispiele zeigen die praktische Verwendung des Arduino. ([weiterlesen](#))



Siebter freiesMagazin-Programmierwettbewerb: Tron

Seite 40

Der letzte freiesMagazin-Programmierwettbewerb ist bereits zwei Jahre her. Eine Abstimmung im September 2013 zeigte, dass die Leser und potentiellen Wettbewerbsteilnehmer das meiste Interesse daran hatten, den Film „Tron“ bzw. das Lightcycle-Rennen darin umzusetzen. 2013 ergab es sich leider nicht, aber dieses Jahr ist es wieder so weit: Der siebte freiesMagazin-Programmierwettbewerb kann beginnen. ([weiterlesen](#))



Editorial

Siebter freiesMagazin-Programmierwettbewerb

Wie vor zwei Monaten im Editorial angekündigt, haben wir es geschafft, dieses Jahr endlich wieder einen Programmierwettbewerb zu starten. Die Idee für den Wettbewerb basiert auf dem Lightcycle-Race aus dem Film „Tron“ [1]. Sie wurde im September 2013 bei einer Umfrage von den freiesMagazin-Lesern als Aufgabe gewählt [2].

In einer Arena treten zwei Bots gegeneinander an. Jede Runde können sie sich entscheiden, nichts zu tun oder sich um 90 Grad nach rechts oder nach links zu drehen. Danach bewegen sie sich automatisch ein Feld vorwärts. Hierbei besetzt jeder Bot das Feld, das er gerade verlässt, welches dadurch für jeden unpassierbar bleibt (auch für den Bot selbst). Wer bei einer Bewegung gegen eine Wand bzw. gegen ein besetztes Feld läuft/fährt, scheidet aus.

Das einfache Konzept macht es hoffentlich auch vielen Programmieranfängern leicht, am Wettbewerb teilzunehmen. Alles weitere erfährt man im zugehörigen *Artikel* auf [Seite 40](#).

Kurzgeschichten-Experiment

Letzten Monat starteten wir im Magazin ein kleines Experiment. Ein Autor schickte uns eine Kurzgeschichte zu und fragte, ob wir diese veröffentlichen wollen, auch wenn so etwas nicht ganz in das Konzept von freiesMagazin passt. Wir ent-

schieden uns, das Experiment zu wagen und gleichzeitig auch die Leser zu fragen, was sie davon halten.

An der Umfrage [3] haben sich 147 Leser beteiligt. 60% (89 Stimmen) waren für eine Fortsetzung des Kurzgeschichten-Experiments. 22% (33 Stimmen) waren dagegen. Dem Rest war es egal. Damit wollen wir dem Autor Stefan Wichmann auch in dieser Ausgabe die Möglichkeit geben, etwas zu präsentieren. Insgesamt waren die Leserbriefe zu dem Thema positiv, wenn natürlich auch Kritik geäußert wurde.

Zusätzlich geht der Aufruf auch an alle Hobby-Schriftsteller da draußen: Wenn Sie eine interessante Kurzgeschichte haben, die sich irgendwie mit Linux, Open Source oder Technik beschäftigt, schicken Sie uns diese doch unter redaktion@freiesMagazin.de zu.

Wichtig ist aber: freiesMagazin hat seine Wurzeln in Artikeln aus den Bereichen Linux, Open Source und Freier Software – und dies wollen wir auch beibehalten.

Ihre freiesMagazin-Redaktion

LINKS

- [1] [https://de.wikipedia.org/wiki/Tron_\(Film\)](https://de.wikipedia.org/wiki/Tron_(Film))
- [2] <http://www.freiesmagazin.de/20130909-ergebnisse-der-abstimmung-fuer-programmierwettbewerb>
- [3] <http://www.freiesmagazin.de/20140907-moechten-sie-weitere-kurzgeschichten-in-freiesmagazin-lesen>

[Teilen](#)[Kommentieren](#)

Inhalt

Linux allgemein

[Der September im Kernelrückblick](#) S. 3

Anleitungen

[Administration von Debian & Co – Teil III](#) S. 5

[Reprepro](#) S. 10

Hardware

[Arduino](#) S. 20

Community

[Bericht: Herbsttagung DANTE e.V. 2014](#) S. 25

[Rezension: Seven Web Frameworks in Seven Weeks](#) S. 27

[Rezension: Spieleprogrammierung mit Android Studio](#) S. 29

[Fo Rensiker – sofort!](#) S. 31

Magazin

[Editorial](#) S. 2

[7. freiesMagazin-Programmierwettbewerb](#) S. 40

[Leserbriefe](#) S. 44

[Veranstaltungen](#) S. 46

[Vorschau](#) S. 46

[Konventionen](#) S. 46

[Impressum](#) S. 47

Der September 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 3.17

Zur Veröffentlichung der finalen Version von Linux 3.17 kam es im September nicht mehr. Stattdessen lief die Entwicklung ihren gewohnten Gang. An der vierten Entwicklerversion [1] war das Bemerkenswerteste, dass sie sehr kompakt ausfiel. Knapp über 200 Änderungen verteilten sich über etwa 150 Dateien. Die umfangreichste einzelne Änderung dürfte dabei ein neuer Treiber für Elantech-Trackpoints sein, ansonsten wurde das Flash-Dateisystem f2fs [2] mit einem größeren Satz an Fehlerkorrekturen bedacht.

Linux 3.17-rc5 [3] machte jedoch jede Illusion einer frühzeitigen Beruhigung der Entwicklung zunichte. Unter den 330 Änderungen findet sich unter anderem eine Sammlung von Patches für die USB-Treiber, die alleine fast ein Zehntel der bearbeiteten Quelltextzeilen ausmachen. Die Behebung von Problemen an f2fs wurden fortgeführt – wenngleich auch in erheblich geringerem Umfang verglichen zum -rc4. Auch Korrekturen am Intel-Grafiktreiber, wo unter anderem ein Deadlock [4] beseitigt wurde, fielen ziemlich umfangreich aus.

Die sechste Entwicklerversion [5] fiel dann so klein aus, dass man schon auf ein baldiges Ende der Entwicklung hoffen konnte. Die dicksten Änderungen in -rc6 finden sich im ARM-Bereich, wo die Entwickler ein Datenleck stopften, das Anwendungen das Auslesen von Speicherstellen anderer Prozesse erlaubte. Fast ebenso umfangreich war eine Sammlung von Patches für die PCI-Treiber [6], die unter anderem die Erkennung von Video-Geräten auf Apple-Systemen mit zwei Grafik-Kernen verbessert.

Die folgende Woche zerschlug jedoch jegliche Hoffnung auf eine kommende Final-Version und so legte Torvalds am Sonntag dann auch einen -rc7 [7] vor. Er war etwas größer als sein Vorgänger und enthielt gerade genug Änderungen, damit sich Torvalds etwas mehr Reifezeit für den Code gönnen konnte. Dabei sind die meisten Änderungen recht unspektakulär. Knapp 360 Zeilen an Änderungen entfallen auf Verschiebungen von Quellcode im Umfeld des i2c-Busses [8]. Ein letzter Satz an Patches für die ARM-Architektur kam ebenfalls hinzu, der hauptsächlich kleine Korrekturen enthielt. Die Rücknahme einer vorhergehenden Änderung und Korrekturen einiger mit Linux 3.14 eingeschlichener Fehler an der Energieverwaltung ACPI [9], trugen in Summe etwas bei.

Fit für die Zukunft

Einen Ausblick auf die Zukunft der Massenspeicher gibt ein Satz von Patches, die Matt Wilcox

vorgelegt hatte [10]. Konkret geht es um die Nutzung von NVRAM [11], sozusagen Arbeitsspeicher, der seine Daten auch ohne Stromzufuhr behält. Hier hofft man darauf, dass damit Speichermedien im gleichen Umfang wie heutige Festplatten zur Verfügung stehen werden, die jedoch sehr viel schneller gelesen und beschrieben werden können.

Würden NVRAM-basierte Speicher nun wie herkömmliche blockorientierte Geräte behandelt, so müssten alle Daten beim Zugriff zuerst in den Page Cache im nur wenig schnelleren Arbeitsspeicher kopiert werden, bevor damit gearbeitet werden kann. Effektiver wäre es, wenn man direkt auf die Daten im Massenspeicher zugreifen würde. Für das etwas ältere Dateisystem ext2 existiert bereits eine Methode, um Dateien direkt dem Userspace zur Verfügung zu stellen. Diese Methode, Execute-In-Place (XIP), nutzte Wilcox nun, um darauf aufbauend ein neues Subsystem namens DAX (Direct access) für ext4 zu entwickeln.

Noch ist DAX nicht für die Aufnahme in den Kernel bereit, die Änderungen wurden bislang noch nicht einmal für den Staging-Zweig vorgeschlagen. Dennoch hat diese Diskussion in der Entwicklergemeinde bereits zu Verbesserungen am Code geführt. Nun fehlen den Dateisystem-Entwicklern nun noch einige schnelle und große NVRAM-Speicher, an denen sie DAX testen und ihm den Feinschliff geben können.

NVIDIA und freie Treiber

NVIDIA war in der Vergangenheit nicht unbedingt für die bedingungslose Unterstützung freier Treiber für ihre Grafik-Chips bekannt, bis sie vor etwa einem Jahr die Dokumentationen ihrer Grafikprozessoren teilweise für Entwickler des freien Treibers Nouveau geöffnet hatten [12]. Eine Neuerung der Chip-Generation „Maxwell“ könnte jedoch wieder zur Entfremdung führen [13]. So sollen diese Chips künftig die gerade laufende Firmware überprüfen, ob sie von NVIDIA zertifiziert ist. Ist dies nicht der Fall, wird der Zugriff auf bestimmte Funktionen und Speicherstellen unterbunden.

Der freie Grafik-Treiber Nouveau [14] benötigt nun ebenfalls Firmware, die beim Start in die Grafikeinheit geladen wird. Diese wurde als freie Software mittels Reverse Engineering und den im letzten September freigegebenen Dokumentationen entwickelt und enthält keine gültige Signatur von NVIDIA. Damit dürfte mit diesem Treiber die Maxwell-Generation nur eingeschränkte Funktionen bieten. NVIDIA plant zwar, ihre proprietäre

Firmware eigenständig zum Herunterladen zur Verfügung zu stellen, doch deren Nutzung widerspricht dem Gedanken der freien und offenen Software, da es sich um einen fertig kompilierten binären Blob [15] handelt, dessen Quellcode nicht eingesehen werden kann.

Andy Ritger äußerte bereits, dass dies nicht die beliebteste Entscheidung sein wird und dürfte damit richtig liegen, wenn auch die Reaktionen auf der Nouveau-Mailingliste bislang eher verhalten ausfielen. Allerdings hat er den Umgang mit dieser Neuerung als Diskussionspunkt für die im Oktober stattfindende X.Org Developers Conference in Aussicht gestellt.

- [6] https://de.wikipedia.org/wiki/Peripheral_Component_Interconnect
- [7] <https://lkml.org/lkml/2014/9/28/161> 
- [8] <https://de.wikipedia.org/wiki/1%C2%B2C>
- [9] https://de.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface
- [10] <http://www.pro-linux.de/-0h2153e7>
- [11] <https://de.wikipedia.org/wiki/NVRAM>
- [12] <http://www.pro-linux.de/-0h214f3e>
- [13] <http://www.pro-linux.de/-0h21543c>
- [14] <https://de.wikipedia.org/wiki/Nouveau>
- [15] https://de.wikipedia.org/wiki/Binary_Large_Object

LINKS

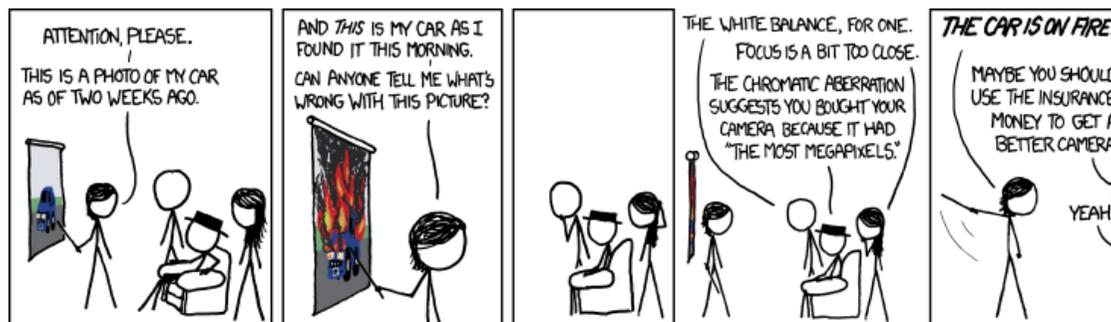
- [1] <https://lkml.org/lkml/2014/9/7/127> 
- [2] <https://de.wikipedia.org/wiki/F2FS>
- [3] <https://lkml.org/lkml/2014/9/14/233> 
- [4] [https://de.wikipedia.org/wiki/Deadlock_\(Informatik\)](https://de.wikipedia.org/wiki/Deadlock_(Informatik))
- [5] <https://lkml.org/lkml/2014/9/21/138> 

Autoreninformation

Mathias Menzer ([Webseite](#)) behält die Entwicklung des Linux-Kernels im Blick, um über kommende Funktionen von Linux auf dem Laufenden zu bleiben.

 [Teilen](#)

 [Kommentieren](#)



“Car Problems” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/1014/>

Administration von Debian & Co im Textmodus – Teil III von Hauke Goos-Habermann und Maren Hachmann

Im letzten Teil des dreiteiligen Workshops für alle, die sich grundlegende Kenntnisse in der Administration von Debian-basierten Systemen aneignen müssen oder wollen, geht es nun darum, wie man selbst Pakete baut, welche System-Werkzeuge man für den Notfall in der Schreibtischschublade haben sollte, wie man Fehlern auf die Spur kommt, was man tun kann, wenn der Bootmanager nicht mehr will und wie man verlorene Passwörter neu setzen kann.

Um richtig mitmachen und alles selbst ausprobieren zu können, benötigt man zwei virtuelle Maschinen. Der Link zum Herunterladen der VM und eine Anleitung zum Einrichten sind im ersten Teil der Serie im [freiesMagazin 08/2014 \[1\]](#) zu finden. Im zweiten Teil der Serie in [freiesMagazin 09/2014 \[2\]](#) wurden die Netzwerkeinstellungen der Workshop-VM verändert. Quereinsteiger, die sich nur für den Themenbereich dieses dritten Workshopteils interessieren, sollten also zumindest den Anfang vom ersten Teil und die Einstellung der Netzwerkparameter aus dem zweiten Teil nachholen, um an der Workshop-VM mitmachen zu können.

Weiterhin ist es für jeden, der möglichst viel von den Inhalten mitnehmen möchte, empfehlenswert, die hier aufgeführten Kommandos selbst auszuprobieren, auch wenn nicht ausdrücklich dazu aufgefordert wird.

Probleme, die beim Ausprobieren auftreten und die entsprechenden Problemlösungsstrategien lassen sich so am einfachsten finden, erproben und im Gedächtnis verankern.

Auf dem virtuellen Workshop-Computer kann ja zum Glück – im Gegensatz zum Server „im echten Leben“ – nichts kaputt gehen, was durch ein erneutes Importieren der virtuellen Maschine nicht wiederherzustellen wäre.

Pakete bauen

Im letzten Workshop-Teil ging es unter anderem um die Benutzung der Debian-Paketverwaltung. Es wurden Paketquellen eingerichtet, Pakete gesucht, installiert, aktualisiert und auch wieder gelöscht.

Aber immer nur vorgefertigte Pakete zu benutzen wäre doch langweilig, oder? Also die Systemwerkzeuge angeworfen und flugs ein eigenes Paket gebastelt!

Pakete aus Debian-Quellen

Am einfachsten bekommt man den Quelltext eines Debian-Paketes, indem man

```
# apt-get source <Paketname>
```

in einem neuen Verzeichnis ausführt. APT [3] lädt daraufhin den Quelltext herunter und entpackt ihn auch gleich. Damit dies funktioniert, müssen in

der Paketquellenliste (siehe „Quellen einrichten“ in Teil II der Serie) aber nicht nur die Binärquellen, sondern auch die Quelltextquellen eingetragen sein. Ein Aufruf von

```
# apt-get build-dep <Paketname>
```

versorgt das System zudem mit allen Entwicklerpaketen, die benötigt werden, um das Paket zu bauen. Nachdem alles an Ort und Stelle ist, wechselt man in das Quell-Verzeichnis des Paketes

```
$ cd <Quellverzeichnis>
```

und startet den Kompilierungs- und Paketierungsvorgang mit

```
# dpkg-buildpackage -us -uc
```

Nach einiger Zeit sollten sich die Debian-Pakete (Endung **.deb**) im Verzeichnis darüber befinden und installieren lassen. Die Parameter **-us** und **-uc** erlauben das Bauen eines Paketes, ohne es signieren zu müssen. Bei einem Paket „für den Eigengebrauch“ kann man das machen – für offizielle, selbstgebaute Pakete sollte man natürlich eine entsprechende Schlüsselinfrastruktur vorbereiten.

Wofür muss man diesen Aufwand mit dem „Selberbauen“ nun eigentlich betreiben, die Binärpakete gibt es bei Debian doch schon – und Langwei-

le hat man doch auch nicht wirklich? Manchmal möchte man die Funktionalität eines Paketes verbessern (z. B. Bugs beheben), und nachdem man einen Patch auf den Quelltext angewandt oder anderweitige Veränderungen vorgenommen hat, muss man das Paket dann leider selbst aus den veränderten Quelldateien erstellen.

Pakete aus Quelltexten

Häufiger dürfte der Fall sein, dass es die gewünschte Software nicht oder nur in einer älteren Version in der Paketquelle der Distribution gibt. Wenn man nun diese Software aber dennoch benötigt, geht man meistens (ggf. mit kleinen Abwandlungen) wie folgt vor:

- eventuelle Abhängigkeiten für die Kompilierung herunterladen und installieren (Informationen hierzu sind ggf. in der beigefügten Dokumentation zu finden);
- Quelltext herunterladen und in einem neuen Verzeichnis entpacken;
- in das Quelltextverzeichnis wechseln;
- `./configure --prefix=/usr` aufrufen;
- und die Quellen mit `make` kompilieren.

Nachdem die Software erfolgreich kompiliert wurde, kann diese nun (anstelle von `make install`, denn dieses würde die Software unter Umgehung der Paketverwaltung installieren) als root-Benutzer mit dem Befehl `./checkinstall -D` in ein Debian-Paket verwandelt werden. Die Installation der `.deb`-Datei und das Entfernen ist im Kapitel „Paketverwaltung“ beschrieben.

Notfallkoffer

Jeder Administrator sollte – für den Fall des Falles – einen „Notfallkoffer“ mit wichtigen Systemwerkzeugen darin bereithalten. Doch was sollte enthalten sein? Empfehlenswert sind die folgenden Bestandteile:

- Eine halbwegs aktuelle Knoppix-CD(-RW) [4], sowie ein Knoppix, das mittels UNetbootin [5] auf einen USB-Stick geschrieben wurde – denn es soll ja bekanntlich auch Rechner ohne optische Laufwerke geben. . .
- Genauso sollten auch eine Clonezilla-CD [6] sowie ein entsprechend bestückter USB-Stick nicht fehlen, um Festplatten und Partitionen zu kopieren, zu sichern und wiederherzustellen. Je nach Geschmack gibt es natürlich auch noch eine Reihe weiterer Live-Linuxe mit verschiedenen Schwerpunkten.
- Ein Administrator sollte sich außerdem darüber Gedanken machen, ob – falls ihm etwas zu stößt oder er aus anderen Gründen verhindert ist – es anderen Personen möglich ist, seine Aufgaben zu übernehmen. Für diesen Fall sollte er – je nach Wichtigkeit der administrierten Rechner – einen Notfall-Umschlag mit Passwörtern oder eine verschlüsselte Datei an einem sicheren Ort deponieren, der den möglichen Vertretern jedoch bekannt sein sollte. Bei einigen Accounts sind hierbei natürlich die AGB des jeweiligen Anbieters zu beachten, die eine Passwortweitergabe untersagen. Ein Ausweg kann hier das Anlegen zusätzlicher Administrator-Accounts sein.

Fehler identifizieren (und beheben)

Fehler sind lästig, aber bei komplexen Systemen letztlich kaum zu vermeiden. Bei den meisten Fehlern ist man zum Glück nicht auf sich allein gestellt und das Internet und die Suchmaschine des Vertrauens (bzw. des geringsten Misstrauens) bieten für fast jeden Fehler eine passende Lösung. Um diese zu finden, sind gute Suchbegriffe nötig. Neben der Fehlersuche und Testreihen geht es in diesem Kapitel um möglichst gute Suchbegriffe.

Suchmaschinen füttern

Als Faustregel gilt, dass die Suchmaschine mit exakten Fehlermeldungen (ggf. bereinigt von persönlichen Informationen wie Rechner- oder Benutzername, etc.) gefüttert werden sollte. Zu der Fehlermeldung selbst sollte der Name des fehlerproduzierenden Programms hinzugenommen werden. Sollten die Suchergebnisse zu umfangreich oder unpassend ausfallen, so ist es sinnvoll, die Suchanfrage um den Namen der Distribution (zum Beispiel Debian, Ubuntu) und gegebenenfalls den Releasenamen (z. B. Wheezy, Quantal) oder ganz allgemein um „Linux“ zu erweitern.

„Programmgesprächigkeit“ erhöhen

Viele (Kommandozeilen-)Programme bieten die Möglichkeit, sie „gesprächiger“ zu machen und so weitergehende Informationen über den Programmablauf (inklusive genauerer Fehlermeldungen) zu gewinnen. Bei vielen Programmen verbirgt sich diese Funktionalität hinter einer „verbose“-Option, die oft mit `-v` oder `-V` aktiviert wird.

Alternativ kann diese auch unter den Begriff „debug“ oder „log“ fallen. Der genaue Name der Option sowie weitere Parameter stehen entweder in der Manpage (**man**) oder können über die eingebaute Hilfefunktion des Programms abgerufen werden.

Systemfehler finden

Eine wichtige Anlaufstelle für Systemfehler sind die Ausgaben von

```
$ dmesg
```

und die Dateien unter **/var/log** und in den (ggf. vorhandenen) Unterverzeichnissen. Hier befinden sich Logdateien diverser Programme. Bei dieser Vielzahl an Logdateien kann aber schon mal die Übersicht verlorengehen. Hier hilft die Eingabe der folgenden Befehlskombination

```
$ ls -lt /var/log/ | head
```

der die Dateien nach der Änderungszeit sortiert, wobei die neuesten ganz oben stehen. Dies sollte man möglichst schnell nach dem Auftreten eines Fehlers machen, da ansonsten andere Programme neuere Logdateien schreiben könnten und somit die interessante Datei aus der Auflistung „herauswandert“. Schließlich beschränkt

```
$ head
```

die Ausgabe auf 10 Zeilen [7]. Die so gefundene Datei sollte nun ausgiebig mit dem Befehl **less**

<Dateiname> (verlassen mit **Q**) oder Nano begutachtet werden, um Fehlermeldungen und andere Hinweise auf das Problem zu finden.

Was läuft?

Auf einem Linuxrechner laufen eine Vielzahl von Programmen gleichzeitig. Welche das sind, verrät

```
$ ps -A
```

Sollte sich ein (eigentlich) laufendes Programm nicht darunter befinden, so sollte man diesem seine besondere Aufmerksamkeit schenken. Außerdem liefert dieser Befehl auch die Prozessnummern, die man zum „gewaltsamen“ Beenden eines Programms benötigt, das auf andere Aufforderungen zum Beenden nicht mehr reagiert.

Möchte man ein Programm (bzw. einen Prozess) beenden, so geschieht dies durch [8]

```
# kill <Prozessnummer>
```

beziehungsweise

```
# killall <Prozessname>
```

Sollte sich der Prozess nicht zum Beenden bewegen lassen, so hilft der zusätzliche Parameter **-9** (oder **-KILL**). Also:

```
# kill -9 <Prozessnummer>
```

beziehungsweise

```
# killall -9 <Prozessname>
```

Wer sendet?

Genauso interessant wie die Frage nach laufenden Programmen kann die Frage danach sein, wer wohin sendet bzw. wer auf welchen Ports lauscht. Die Frage beantwortet **netstat** [9]:

```
# netstat -a
```

Programme belauschen

Das Programm **strace** belauscht andere Programme und erstellt ein detailliertes Protokoll. Da **strace** eine Menge Informationen sammelt, sollten diese von der Fehlerausgabe direkt in eine Datei umgeleitet werden. Ein Aufruf könnte folgendermaßen aussehen:

```
# strace -f -o /tmp/logdateiname <~>
  Programmname> <Programmparameter>
```

Dieser Aufruf startet das angegebene Programm und schreibt alle Informationen (und mit dem Parameter **-t** auch die der sog. Kindprozesse) in die Datei **/tmp/logdateiname**. Diese Datei sollte man nun genau begutachten. Geht es z. B. um Fehler mit nicht gefundenen Dateien, so kann man diese in der Logdatei leicht mit den Suchbegriffen „ENOENT“ und „open“ finden.

Bootmanager und Live-Linux

Der Bootmanager (meist kommt GRUB [10] zum Einsatz) sorgt dafür, dass ein oder auch mehrere Betriebssysteme starten können. Aus verschie-

denen (glücklicherweise eher seltenen) Gründen kann es dazu kommen, dass der Bootmanager beschädigt wird und somit kein Booten mehr möglich ist. Eine Neuinstallation des Betriebssystems ist normalerweise nicht notwendig. Mit ein bisschen „Admin-Magie“ lässt sich der GRUB-Bootmanager rasch „wiederbeleben“ – allerdings nur dann, wenn die Festplatte nicht verschlüsselt wurde (alles andere würde den Umfang dieses Workshops sprengen).

Live-Linux booten

In diesem Abschnitt geht es um das Booten eines Live-Linux-Systems und den Wechsel auf das System der Festplatte. Dieses Vorgehen ist nicht nur bei GRUB-Problemen nützlich, sondern auch bei anderen Arten von Fehlern, bei denen das installierte System nicht mehr startet.

Zuerst bootet man das Live-Linux (hier z. B. Knoppix [4]) und gibt bei Knoppix direkt beim Startbildschirm (Textmodus) den Befehl

```
knoppix 2
```

ein. Dies bootet Knoppix im Textmodus. Bei anderen Live-Linuxen muss man eine root-Shell starten. Nun ermittelt man die richtige Festplatte mit der Partition des Betriebssystems:

```
# parted /dev/sda print
```

Wenn dies nicht die richtige Festplatte sein sollte, probiert man weiter mit sdb, sdc und so weiter.

Nun muss man die Partition des Betriebssystems (z. B. an Größe und dem Dateisystem wie z. B. ext3 oder ext4 zu erkennen) finden. Der komplette Gerätenamen der Partition setzt sich dann aus der Festplatte und Partitionsnummer (`/dev/<Festplatte><Partitionsnummer>`) zusammen, also z. B. `/dev/sda1`.

Im Live-Linux legt man dann ein Verzeichnis zum Einhängen an:

```
# mkdir /mnt/linux
```

Nun kann man die Partition einhängen:

```
# mount /dev/sda1 /mnt/linux
```

Danach wechselt man in das zu reparierende System [11] und startet eine Bash-Shell:

```
# chroot /mnt/linux
# bash
```

Nun sollte man sicherstellen, dass das (neue) root-Dateisystem wirklich beschrieben werden kann. Dazu lässt man das Programm `mount` [12] mit den Optionen **remount** („neu mounten“) und **rw** („im Lese-Schreib-Modus“) das Wurzelverzeichnis neu einhängen:

```
# mount -o remount,rw /
```

Zum Schluss muss man noch die sys- und proc-Pseudodateisysteme einhängen:

```
# mount /proc
# mount /sys
```

GRUB reparieren

Die folgenden Tricks helfen bei Problemen mit GRUB, falls z. B. `/dev` leer sein sollte:

```
# cd /dev
# MAKEDEV generic
```

GRUB kann man auch neu auf der Systemplatte (hier `/dev/sda`) installieren:

```
# grub-install -f /dev/sda
```

Zum Schluss sollte man die GRUB-Einstellungen aktualisieren:

```
# update-grub
```

Aufräumarbeiten

Am Schluss einer jeden Live-Linux-Sitzung sollte man zunächst sicherstellen, dass die Daten auf die Festplatte geschrieben wurden:

```
# sync
```

Danach kann man den Rechner neu starten:

```
# reboot -f
```

Passwort vergessen

Es sollte natürlich nicht vorkommen (vor allem nicht beim Administrator), aber in der Praxis ge-

schieht es angeblich öfter, dass Passwörter vergessen werden...

Wenn es sich um die Passwörter von lokalen Benutzern (auch **root** gehört dazu) handelt, ist es nicht schwer, ein neues Passwort zu setzen.

Wenn man das Root-Passwort vergessen hat (und die Festplatte nicht verschlüsselt wurde, denn in diesem Fall hat man schlechte Karten), ist es notwendig, zunächst das Live-Linux zu booten und in das System, dessen Passwort man vergessen hat, „hinein zu chrooten“ – wie oben im Kapitel „Bootmanager und Live-Linux“ beschrieben.

Um dann das root-Passwort zu ändern, genügt

```
# passwd
```

Hat hingegen ein Benutzer sein Passwort vergessen, so ist es ausreichend, den folgenden Befehl aufzurufen (ohne Live-Linux, einfach als root aus der Shell):

```
# passwd <Benutzername>
```

Geschafft!

Wer nun erfolgreich das Ende des Workshops erreicht hat, kann darauf stolz sein, einen großen Teil des grundlegenden Debian-Administrator-Handwerkszeugs ausprobiert und gelernt zu haben. Sicherlich deckt das neu erworbene Wissen nicht alles ab, was einem in der Praxis begegnen kann, doch nun hat man auf jeden Fall schon einmal einen „roten Faden“, dem man im Problemfall folgen kann. Allen „Workshop-Absolventen“ sei an dieser Stelle herzlich gratuliert – einem sicheren Durchkommen durch den „Debian-Dschungel“ steht nun nichts mehr im Wege!

LINKS

- [1] <http://www.freiesmagazin.de/freiesMagazin-2014-08>
- [2] <http://www.freiesmagazin.de/freiesMagazin-2014-09>
- [3] http://de.wikipedia.org/wiki/Advanced_Packaging_Tool

- [4] <http://www.knoppix.org/>
- [5] <http://unetbootin.sourceforge.net/>
- [6] <http://clonezilla.org/>
- [7] [http://de.wikipedia.org/wiki/Head_\(Unix\)](http://de.wikipedia.org/wiki/Head_(Unix))
- [8] [http://de.wikipedia.org/wiki/Kill_\(Unix\)](http://de.wikipedia.org/wiki/Kill_(Unix))
- [9] <http://de.wikipedia.org/wiki/Netstat>
- [10] http://de.wikipedia.org/wiki/Grand_Unified_Bootloader
- [11] <http://de.wikipedia.org/wiki/Chroot>
- [12] http://de.wikipedia.org/wiki/Mounten#Das_Unix-Kommando_mount

Autoreninformation

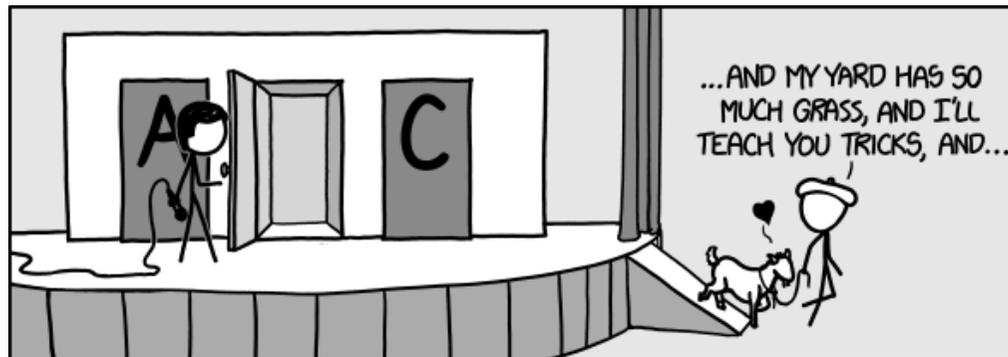
Hauke Goos-Habermann und Maren Hachmann ([Webseite](#)) sind beide langjährige Linuxnutzer und Autoren der Softwareverteilung m23.



Teilen



Kommentieren



“Monty Hall” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/1282/>

Reprepro – Debian-Systeme mit einem selbst aufgesetzten Paket-Repo versorgen von Valentin Höbel

Wer mehrere Debian-Systeme verwaltet kommt irgendwann einmal um das Erstellen eigener Debian-Pakete nicht mehr herum. Wer die selbst erstellten Pakete nicht mittels `rsync & Co.` auf den eigenen Servern ablegen möchte oder nach einer professionelleren Lösung strebt, setzt dafür besser ein eigenes Debian-Repository auf. Dieser Artikel erläutert die einzelnen Schritte zum Aufsetzen eines eigenen Repositories.

Einführung

Das Aufsetzen eines eigenen Servers für die Speicherung und Organisation von Debian-Paketen mag einen gewissen Initialaufwand mit sich bringen, als Resultat erhalten Systemadministratoren jedoch ein mächtiges Werkzeug mit einer ganzen Reihe an Vorteilen.

Die Vorteile eines eigenen Debian-Paket-Repos

Neben eigenen Paketen kann ein selbst aufgesetztes Paket-Repo beispielsweise auch den gesamten Debian-Paket-Pool spiegeln. Das ist besonders sinnvoll, wenn im Rechenzentrum Traffic eingespart werden soll, da nur noch ein Server die aktuellen Pakete von einem Debian-Mirror bezieht und die restlichen Systeme am eigenen Repo angebunden sind.

Wer darüber hinaus auch seine eigenen Pakete in den selbst aufgesetzten Paket-Pool schiebt, kann

diese in verschiedenen Distributionen organisieren und ein Staging für den Entwicklungs-, Pilot- und Live-Betrieb einführen.

Je nachdem, in welchem Bereich die eigenen Debian-Server dann betrieben werden sollen, können diese die Pakete aus den jeweiligen Staging-Distributionen beziehen. So lassen sich mit einfachen Mitteln professionelle Arbeitsweisen für die eigene Server-Landschaft etablieren.

Die Qual der Wahl

Wer sich einmal für die Einführung eines eigenen Paket-Repos entschieden hat, begibt sich womöglich mit Hilfe von Google auf die Suche nach einer geeigneten Lösung. Besonders häufig finden sich Berichte und Anleitungen zu `aptly`, `dak`, `debpool`, `freight` und `reprepro` im Netz. Eine Übersicht von verbreiteten Tools findet sich darüber hinaus im Debian-Wiki [1].

In diesem Artikel wird das Aufsetzen eines eigenen Debian-Repositorys inklusive Debian-Mirror mittels `reprepro` [2] beschrieben, da sich dieses Werkzeug in den offiziellen Paket-Quellen finden lässt, einfach zu bedienen ist und ohne externe Datenbank auskommt.

Die Testumgebung

Die Testumgebung wurde unter Debian Wheezy aufgesetzt. Während der erste Server **repo.xenuser.org** als Un-

tersatz für den Paket-Pool dienen soll, wird im späteren Verlauf des Artikels das System **client.xenuser.org** an das frisch installierte Repo angebunden. Beide Systeme laufen als virtuelle Systeme bei verschiedenen Hosting-Anbietern.

Ob ein Debian-Repo virtualisiert oder auf einer physikalischen Maschine betrieben werden sollte, lässt sich pauschal nicht festlegen. In der Regel genügt jedoch eine performante virtuelle Maschine.

Für beide Server kann eine Standard-Partitionierung vorgenommen werden. Wichtig ist nur, dass auf dem Repo-Server eine Partition mit mindestens 65 GB Speicherplatz zur Verfügung steht. Im Rahmen dieses Artikels wird dafür das Verzeichnis `/var/www` verwendet.

In diesem Fall sind die Systeme „repo“ und „client“ identisch, da beide VMs dem günstigsten Tarif des verwendeten Hosters entsprechen. Für **client.xenuser.org** hätten hier auch 1 vCPU sowie 512 MB RAM mit insg. 8 GB Speicherplatz ausgereicht.

Das Test-Setup im Überblick.

Hostname	OS	vCPUs	vRAM	Speicher
repo.xenuser.org	Deb7 64 Bit	3	2 GB	75 GB
client.xenuser.org	Deb7 64 Bit	3	2 GB	75 GB

Umsetzung

Installation von reprepro unter Debian Wheezy

Ein Paket-Repository besteht meist aus mehreren Komponenten: Zum einen wird ein Tool benötigt, das die Pakete organisiert ablegt und daraus Paketlisten generiert, zum anderen sollte ein Webserver diese Dateien dann über Port 80 (oder bei Bedarf auch via HTTPS) zur Verfügung stellen.

Dementsprechend gilt es, gleich mehrere Pakete und Dienste auf dem Repo-Server zu installieren:

```
# apt-get update
# apt-get dist-upgrade
# apt-get install reprepro apache2
```

Im Anschluss kann die Konfiguration des Webserver erfolgen:

```
# cd /var/www
# htpasswd -c /etc/apache2/.htpasswd ~
repo
New password: <zu Demozwecken: „repo“>
Re-type new password: <zu Demozwecken: „~
repo“>
Adding password for user repo
```

Anschließend wird die Datei `/var/www/.htaccess` erzeugt und mit Inhalt befüllt:

```
AuthType Basic
AuthUserFile /etc/apache2/.htpasswd
AuthName "Login required"
Require valid-user
```

Listing 1: .htaccess

Der Einfachheit halber wird die Standard-VHost-Konfiguration von Apache 2 übernommen und nur minimal angepasst. Hierfür ist das Editieren der Datei `/etc/apache2/sites-available/default` erforderlich:

```
# Der Block für die Definition von /var/www
www sollte wie unten aussehen
<Directory /var/www/>
    Options Indexes ~
    FollowSymLinks MultiViews
    AllowOverride All #<<-- ~
    Hier wird absichtlich vom ~
    Standard abgewichen
    Order allow,deny
    allow from all
</Directory>
```

Listing 2: default

Um sicher zu stellen, dass alle Änderungen übernommen werden, sollte Apache2 an dieser Stelle neu gestartet werden. Für den Test der Authentifizierung ist es zudem sinnvoll, das System über den Browser anzusteuern, z. B. <http://93.180.154.95/>, wobei hier selbstverständlich die IP-Adresse bzw. der Hostname des eigenen Repo-Servers angegeben wird.

Wenn die vorherigen Schritte richtig durchgeführt wurden, erscheint nun eine Login-Abfrage.

Da der Webserver erwartungsgemäß funktioniert, kann nun reprepro konfiguriert werden. Bevor dieser Schritt angegangen wird, sollte man sich je-

doch ausreichend Gedanken über die Struktur des eigenen Paket-Servers machen. Es gilt zu klären, welche Distributionen zur Verfügung gestellt werden sollen.

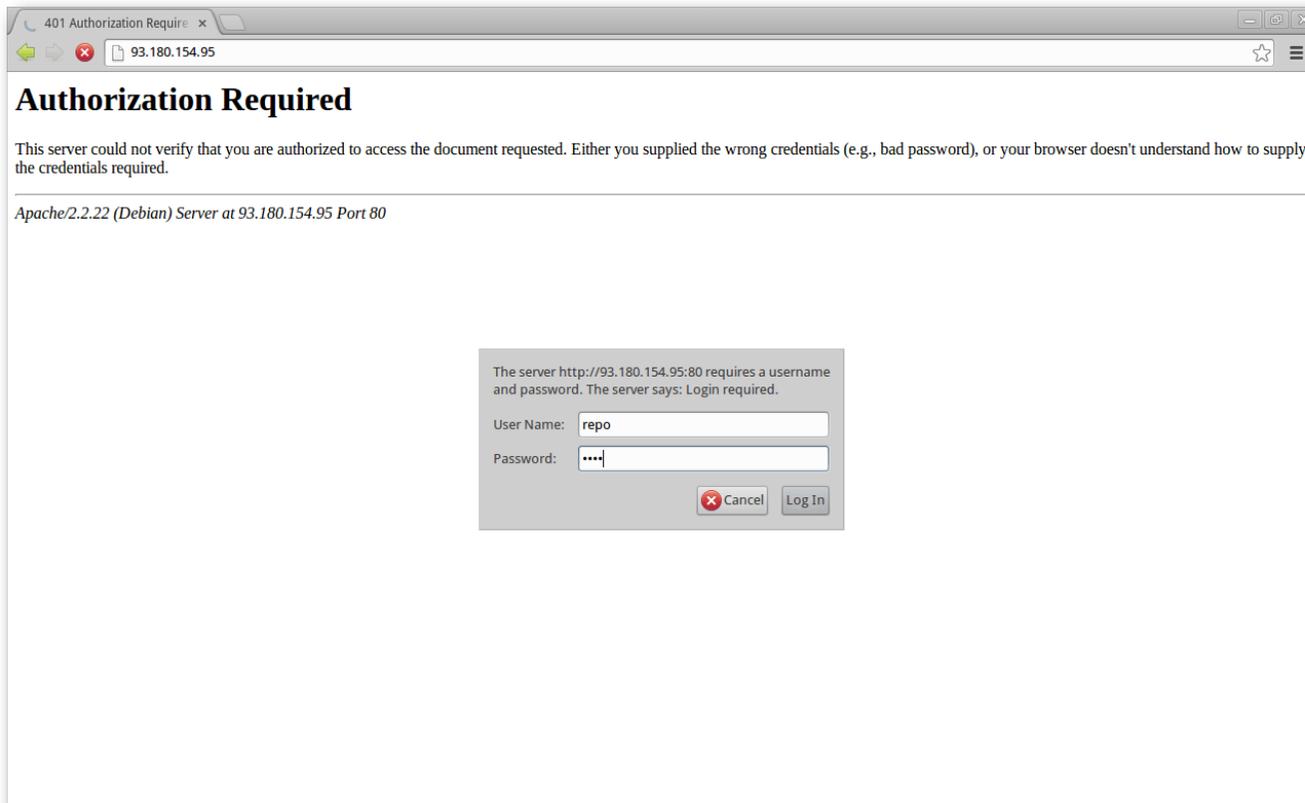
In diesem Test-Setup ist etwa denkbar, dass sowohl der offizielle Debian-Pool gespiegelt als auch eigene Distributionen angeboten werden sollen.

Grundsätzlich werden drei Distributionen angelegt, wobei „wheezy“ sowohl den normalen Paket-Pool von Debian Wheezy als auch (Security)-Updates beinhalten soll. Die Distribution „xenuser“ enthält nur selbst gebaute Pakete, während „external“ Pakete von externen Quellen umfasst (die sich nicht im offiziellen Debian-Pool befinden).

In diesem stark vereinfachten Setup existiert noch keine Form von Staging (z. B. Trennung von Entwicklungs-, Test- und Live-Betrieb). Prinzipiell existieren gleich mehrere Wege, mit denen ein Staging unter reprepro umgesetzt werden kann. Denkbar ist etwa eine Distribution in mehreren Ausführungen anzulegen, beispielsweise:

- xenuser-devel (Entwicklung)
- xenuser-testing (Testing+Pilotierung)
- xenuser-prod (Live-Betrieb)

Diese Form des Staging erlaubt eine größtmögliche Trennung und ein hohes Maß an Flexibilität, da einzelne Distributionen „eingefroren“ werden können. Der Nachteil liegt klar auf der Hand: Mehr Distributionen benötigen mehr Pflegeaufwand.



Login-Abfrage des Webservers auf dem Testsystem. 🔍

Eine Alternative zu dieser Variante des Staging stellt die Verwendung von Sections dar (z. B. „main“, „devel“ usw.). Hier besteht jedoch die Gefahr, dass sich die einzelnen Paketgruppen zu stark miteinander vermischen oder gar Konflikte bei gleichen Paketnamen entstehen.

Im Rahmen dieses Artikels wird daher ein Staging durch Trennung der Distributionen vorgenommen. Womöglich existieren noch andere und bessere

Wege, um ein Staging sauber umzusetzen; im Alltag hat sich jedoch diese Variante bewährt:

Distributionsübersicht inkl. Staging.		
„wheezy“ (Debian-Mirror)	„xenuser“ (nur eigene Pakete)	„external“
-	xenuser-devel	-
-	xenuser-testing	external-testing
wheezy	xenuser-prod	external-prod

In dieser erweiterten Form der Distributionsplanung ist kein Staging für den offiziellen Wheezy-Pool vorgesehen, die selbst gebauten Pakete werden hingegen in den Formen „devel“, „testing“ und „prod“ behandelt. Für die Pakete aus Drittquellen ist nur ein zweistufiges Staging vorgesehen, da diese Pakete in der Regel nicht von einem selbst entwickelt werden, aber zumindest vor dem Produktiveinsatz getestet werden sollten.

Mit diesen Plan im Hinterkopf kann die Einrichtung von reprepro vorgenommen werden:

```
# mkdir /var/www/log
# mkdir /var/www/conf
# ln -s /var/www /var/www/debian # ↪
Wichtig für Erreichbarkeit durch „apt-↪
get“ von Clients
```

Die Distributionen werden innerhalb der selbst zu erstellenden Datei `/var/www/conf/distributions` angelegt:

```
# Configuration for our Wheezy mirror
Origin: Debian
Codename: wheezy
Suite: stable
Description: Official Debian Wheezy mirror
Architectures: i386 amd64
Components: main contrib non-free
UDEbComponents: main
Contents: .gz
Update: - debian-wheezy debian-wheezy-↪
updates debian-wheezy-security
Log: /var/www/log/wheezy.log
ReadOnly: No
```

```
# This dummy entry is required for receiving updates for the Debian mirror
Origin: Debian
Codename: wheezy-updates
Description: Just a dummy entry, not in use.
Architectures: i386 amd64
Components: main contrib non-free
UDEbComponents: main
Contents: .gz
ReadOnly: No

# This dummy entry is required for receiving updates for the Debian mirror
Origin: Debian
Codename: wheezy-security
Description: Just a dummy entry, not in use.
Architectures: i386 amd64
Components: main contrib non-free
UDEbComponents: main
Contents: .gz
ReadOnly: No

# Configuration for xenuser-devel
Origin: custom
Label: xenuser Debian Devel Repository
Codename: xenuser-devel
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian devel packages.
Log: /var/www/log/xenuser-devel.log

# Configuration for xenuser-testing
Origin: custom
Label: xenuser Debian Testing Repository
Codename: xenuser-testing
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian testing packages.
Log: /var/www/log/xenuser-testing.log
```

```
# Configuration for xenuser-prod
Origin: custom
Label: xenuser Debian Production Repository
Codename: xenuser-prod
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian production packages.
Log: /var/www/log/xenuser-prod.log

# Configuration for external-testing
Origin: custom
Label: External tools Debian Testing Repository
Codename: external-testing
Architectures: amd64 i386
Components: main
Description: This repo contains external Debian packages (testing).
Log: /var/www/log/external-testing.log

# Configuration for external-prod
Origin: custom
Label: External tools Debian Production Repository
Codename: external-prod
Architectures: amd64 i386
Components: main
Description: This repo contains external Debian packages (production).
Log: /var/www/log/external-prod.log
```

Listing 3: distributions

Im Anschluss wird die Datei `/var/www/conf/updates` erstellt und mit Inhalt befüllt:

```
Name: debian-wheezy
Method: http://ftp.debian.de/debian
```

```
Name: debian-wheezy-updates
Method: http://ftp.debian.de/debian
Suite: wheezy-updates

Name: debian-wheezy-security
Method: http://security.debian.org/debian-security
Suite: wheezy/updates
```

Listing 4: updates

Während die Datei **distributions** die benötigten Distributionen beschreibt, legt die Datei **updates** fest, von welchen Quellen Pakete bezogen werden können. Weiterführende Informationen zu den möglichen Konfigurationsparametern in den beschriebenen Dateien finden sich unter [3].

Den Debian-Mirror spiegeln

Das selbst gebaute Debian-Repo ist nun fertig konfiguriert. Bevor weitere Schritte erfolgen, bietet es sich an, `reprepro` die Pakete von den offiziellen Debian-Servern holen zu lassen. Dieser Befehl kann nun – je nach Server-Ausstattung – bis zu vier Stunden lang arbeiten und dabei rund 65 GB an Daten ziehen. Es ist daher sinnvoll, ihn im Screen zu starten und nur dann laufen zu lassen, wenn das Netzwerk den zusätzlichen Traffic vertragen kann:

```
# reprepro -b /var/www -V update
```

Die hier verwendete Option `-b` beschreibt unter `reprepro` das Basisverzeichnis **basedir**, in dem das Repo auf dem Dateisystem anzutreffen ist.

```

Terminal - valentin@xt: ~
File Edit View Terminal Go Help
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-ldap_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-mcrypt_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-mysql_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-mysqldb_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-odbc_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-pgsql_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-pspell_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-recode_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-snmp_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-sqlite_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-sybase_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-tidy_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-xmlrpc_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/php5/php5-xs1_5.4.4-14+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-dbg_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-doc_1.1.7-4+deb7u14_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-sane_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-sane-dbg_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-tk_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/p/python-imaging/python-imaging-tk-dbg_1.1.7-4+deb7u14_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/r/reportbug/python-reportbug_6.4.4+deb7u14_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/r/reportbug/reportbug_6.4.4+deb7u14_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/tshark_1.8.2-5wheezy11_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/wireshark_1.8.2-5wheezy11_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/wireshark-common_1.8.2-5wheezy11_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/wireshark-dbg_1.8.2-5wheezy11_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/wireshark-dev_1.8.2-5wheezy11_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wireshark/wireshark-doc_1.8.2-5wheezy11_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wordpress/wordpress_3.6.1+dfsg-1+deb7u4_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/w/wordpress/wordpress-l10n_3.6.1+dfsg-1+deb7u4_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xen-docs-4.1.4-1.4-3+deb7u2_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xen-hypervisor-4.1-amd64_4.1.4-3+deb7u2_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xen-system-amd64_4.1.4-3+deb7u2_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xen-utils-4.1.4-1.4-3+deb7u2_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xen-utils-common_4.1.4-3+deb7u2_all.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/x/xen/xenstore-utils_4.1.4-3+deb7u2_amd64.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/o/openssl/libcrypto1.0.0-udeb_1.0.1e-2+deb7u12_i386.udeb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/a/acpi-support/acpi-fakekey_0.140-5+deb7u2_i386.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/liba/libav/ffmpeg_0.8.15-1_i386.deb'
aptmethod got 'http://security.debian.org/debian-security/pool/updates/main/k/kde4libs/kdelibs-bin_4.8.4-4+deb7u1_i386.deb'

```

reprepro beim Download von offiziellen Debian-Paketen. 🔍

Der Schalter **-V** sorgt für mehr Ausgabe, was besonders bei den ersten Schritten der Repo-Verwaltung sinnvoll ist. Hier auftretende Warnungen zu fehlenden Signaturen oder Distributionen, die keine Update-Konfiguration besitzen, können ignoriert werden.

reprepro zieht hier die drei externen Zweige „wheezy“, „wheezy-updates“ und „wheezy-security“ in der lokal angelegten Distribution „wheezy“

zusammen. Wer diese aktuell halten möchte, kann den Befehl nächtlich per CronJob laufen lassen.

reprepro wird übrigens bei jedem Update alte Pakete löschen und „vergessen“, wenn neue Versionen davon herein kommen.

Nach der Ausführung sieht das Repo-Verzeichnis wie folgt aus:

```

total 40K
drwxr-xr-x 8 root root 4.0K Aug 10 20:14 ~
.
drwxr-xr-x 12 root root 4.0K Aug 10  ~
20:14 ..
drwxr-xr-x 2 root root 4.0K Aug 3 17:37 ~
conf
drwxr-xr-x 2 root root 4.0K Aug 10 19:38 ~
db
lrwxrwxrwx 1 root root 8 Aug 10 20:06 ~
debian -> /var/www
drwxr-xr-x 4 root root 4.0K Aug 10 19:36 ~
dists
-rw-r--r-- 1 root root 96 Aug 1 16:42 . ~
htaccess
-rw-r--r-- 1 root root 177 Aug 1 16:36 ~
index.html
drwxr-xr-x 2 root root 4.0K Aug 3 20:15 ~
lists
drwxr-xr-x 2 root root 4.0K Aug 10 19:36 ~
log
drwxr-xr-x 6 root root 4.0K Aug 10 19:36 ~
pool

```

Arbeiten mit dem Repo

Pakete in das eigene Repo aufnehmen

Wer ein eigenes Debian-Repo betreibt, möchte sicherlich auch eigene Pakete hinzufügen, löschen, innerhalb von Distributionen verschieben oder wo möglich auch einfach nur einsehen können, welche Pakete sich derzeit im Repo befinden.

reprepro bringt dafür gleich eine ganze Reihe von Befehlen mit. Der Einfachheit halber stellt der Autor dieses Artikels ein Testpaket bereit, welches

selbst gebaut und in das eigene Repo verschoben werden kann.

Hierfür empfiehlt es sich auf dem Repo-Server die Pakete **git** und **devscripts** über die Paketverwaltung nachzuinstallieren. Im Anschluss kann das Paket von GitHub geklont und gebaut werden:

```
# mkdir /root/devel
# cd /root/devel
# git clone https://github.com/xenuser/
test-deb-package.git
# cd test-deb-package
# dpkg-buildpackage -tc -us -uc -\
rfakeroot
# cd ..
# ls -lah
total 64K
drwxr-xr-x 3 root root 4.0K Aug 3 18:07 \
.
drwx----- 5 root root 4.0K Aug 3 18:02 \
..
drwxr-xr-x 4 root root 4.0K Aug 3 18:04 \
test-deb-package
-rw-r--r-- 1 root root 1.3K Aug 3 18:07 \
test-deb-package_0.4_amd64.changes
-rw-r--r-- 1 root root 1.2K Aug 3 18:07 \
test-deb-package_0.4_amd64.deb
-rw-r--r-- 1 root root 540 Aug 3 18:07 \
test-deb-package_0.4.dsc
-rw-r--r-- 1 root root 39K Aug 3 18:07 \
test-deb-package_0.4.tar.gz
```

Warnungen können ignoriert werden, so lange als Resultat das frisch gebackene Debian-Paket **test-deb-package_0.4_amd64.deb** vorliegt.

```
Terminal - valentin@xt:~
File Edit View Terminal Go Help
root@repo: /var/www/log# ls -lah
total 5.2M
drwxr-xr-x 2 root root 4.0K Aug 10 21:40 .
drwxr-xr-x 8 root root 4.0K Aug 10 20:14 ..
-rw-r--r-- 1 root root 5.1M Aug 21 22:14 wheezy.log
-rw-r--r-- 1 root root 155 Aug 10 22:02 xenuser-devel.log
-rw-r--r-- 1 root root 78 Aug 10 21:40 xenuser-testing.log
root@repo: /var/www/log# cat xenuser-*
2014-08-10 19:36:48 add xenuser-devel deb custom amd64 test-deb-package 0.4
2014-08-10 22:02:33 remove xenuser-devel deb custom amd64 test-deb-package 0.4
2014-08-10 21:40:54 add xenuser-testing deb custom amd64 test-deb-package 0.4
root@repo: /var/www/log#
root@repo: /var/www/log# tail -f wheezy.log
2014-08-21 22:14:10 replace wheezy deb main i386 wordpress 3.6.1+dfsg-1-deb7u4 3.6.1+dfsg-1-deb7u3
2014-08-21 22:14:10 replace wheezy deb main i386 wordpress-l10n 3.6.1+dfsg-1-deb7u4 3.6.1+dfsg-1-deb7u3
2014-08-21 22:14:10 replace wheezy deb main i386 xen-docs-4.1 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-hypervisor-4.1-amd64 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-hypervisor-4.1-i386 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-system-amd64 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-system-i386 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-utils-4.1 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xen-utils-common 4.1.4-3+deb7u2 4.1.4-3+deb7u1
2014-08-21 22:14:10 replace wheezy deb main i386 xenstore-utils 4.1.4-3+deb7u2 4.1.4-3+deb7u1
```

Blick in die von reprepro erzeugten Log-Dateien der Distributionen.

Nun kann das erzeugte Debian-Paket in die Distribution der eigenen Wahl importiert werden:

```
# cd /root/devel
# reprepro -b /var/www/ includedeb \
xenuser-devel test-deb-package_0.4_amd64\
.deb
Exporting indices...
```

Das eigene Paket befindet sich nun im Pool, was sich leicht verifizieren lässt:

```
# reprepro -b /var/www/ list xenuser-\
devel
xenuser-devel|custom|amd64: test-deb-\
package 0.4
```

Durch das Kommando **list** listet reprepro alle Pakete aus der Distribution „xenuser-devel“ auf.

Alternativ ist auch denkbar, dass man den Paketnamen kennt und wissen möchte, in welcher Distribution sich die zugehörige Datei befindet:

```
# reprepro -b /var/www/ ls test-deb-\
package
test-deb-package | 0.4 | xenuser-devel | \
amd64
```

Ein **find** nach **/var/www/pool** zeigt, an welcher Stelle sich nun das Paket im Dateisystem befindet:

```
# find /var/www/pool/custom/
/var/www/pool/custom/
/var/www/pool/custom/t
/var/www/pool/custom/t/test-deb-package
/var/www/pool/custom/t/test-deb-package/\
test-deb-package_0.4_amd64.deb
```

Alle Pakete werden im Ordner **pool** und in einem Unterordner organisiert, der sich nach der Bezeichnung der „Component“ richtet. In unserem Fall wurde **xenuser-devel** mit der Konfiguration **Components: custom** angelegt, womit alle neu importierten Pakete automatisch zu diesem Bereich zählen. Andere übliche Bezeichner für „Component“ lauten **main**, **contrib** und **non-free**.

Die „Component“ wird auch in den Quellen für apt angegeben, damit das entsprechende Debian-System an der richtigen Stelle nach den Paketen für den Download sucht.

Fans von Log-Dateien kommen übrigens auch bei reprepro auf ihre Kosten – unter **/var/www/log/** ist jeweils pro konfigurierter Distribution eine eigene Datei vorhanden, die u. a. auch die Aufnahme eines neuen Pakets meldet.

Nutzung des eigenen Debian-Repos durch Clients

Eigene Pakete im eigenen Debian-Pool wären sinnlos, wenn diese nicht von anderen Systemen genutzt würden. Daher wird testweise das System **client.xenuser.org** an das Repo angebunden.

Die Datei **/etc/apt/sources.list** wurde dafür mit folgendem Inhalt ausgestattet:

```
deb http://repo:repo@93.180.154.95/
debian wheezy main contrib non-free
deb http://repo:repo@93.180.154.95/
debian xenuser-devel custom
```

```
deb http://repo:repo@93.180.154.95/
debian external-testing custom

# Alter Zustand, deaktiviert
#deb http://ftp.de.debian.org/debian
wheezy main
#deb http://security.debian.org/ wheezy/
updates main
```

Listing 5: *sources.list*

Mit **apt-get update** werden die entsprechenden Paketlisten vom Client heruntergeladen:

```
Ign http://93.180.154.95 xenuser-devel
Release.gpg
Ign http://93.180.154.95 external-
testing Release.gpg
Hit http://93.180.154.95 wheezy Release
Hit http://93.180.154.95 xenuser-devel
Release
Ign http://93.180.154.95 external-
testing Release
Ign http://93.180.154.95 wheezy/main
amd64 Packages/DiffIndex
Ign http://93.180.154.95 wheezy/contrib
amd64 Packages/DiffIndex
Ign http://93.180.154.95 wheezy/non-free
amd64 Packages/DiffIndex
Ign http://93.180.154.95 xenuser-devel/
custom amd64 Packages/DiffIndex
Hit http://93.180.154.95 wheezy/main
amd64 Packages
Hit http://93.180.154.95 wheezy/contrib
amd64 Packages
Hit http://93.180.154.95 wheezy/non-free
amd64 Packages
```

```
Ign http://93.180.154.95 wheezy/contrib
Translation-en_US
Ign http://93.180.154.95 wheezy/contrib
Translation-en
Ign http://93.180.154.95 wheezy/main
Translation-en_US
Ign http://93.180.154.95 wheezy/main
Translation-en
Ign http://93.180.154.95 wheezy/non-free
Translation-en_US
Ign http://93.180.154.95 wheezy/non-free
Translation-en
Hit http://93.180.154.95 xenuser-devel/
custom amd64 Packages
Ign http://93.180.154.95 xenuser-devel/
custom Translation-en_US
Ign http://93.180.154.95 xenuser-devel/
custom Translation-en
Err http://93.180.154.95 external-
testing/custom amd64 Packages
404 Not Found
Ign http://93.180.154.95 external-
testing/custom Translation-en_US
Ign http://93.180.154.95 external-
testing/custom Translation-en

W: Failed to fetch http://93.180.154.95/
debian/dists/external-testing/custom/
binary-amd64/Packages 404 Not Found

E: Some index files failed to download.
They have been ignored, or old ones used
instead.
```

Erwartungsgemäß mündet die Ausgabe des Kommandos in einem Fehler, da reprepro für die Dis-

tribution „external-testing“ keine Paketliste generiert hat. Diese Distribution beinhaltet noch keine Pakete, womit sie für reprepro zunächst noch uninteressant ist. Für das weitere Vorgehen kann dieser Umstand ignoriert werden.

Auf client.xenuser.org kann das Testpaket nun durch die apt-Tools gefunden werden:

```
# apt-cache search test-deb-package
test-deb-package - Just another test package

# apt-get install test-deb-package
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 test-deb-package
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 1,190 B of archives.
After this operation, 29.7 kB of additional disk space will be used.
WARNING: The following packages cannot be authenticated!
 test-deb-package
Install these packages without verification [y/N]? y
Get:1 http://93.180.154.95/debian/xenuser-devel/custom test-deb-package amd64 0.4 [1,190 B]
Fetched 1,190 B in 0s (0 B/s)
Selecting previously unselected package test-deb-package.
```

```
(Reading database ... 27249 files and directories currently installed.)
Unpacking test-deb-package (from .../test-deb-package_0.4_amd64.deb) ...
Setting up test-deb-package (0.4) ...
```

Nach Bestätigung der Nachfrage von apt wird das Paket ordnungsgemäß installiert – das selbst aufgesetzte Repo funktioniert.

apt fragt an dieser Stelle übrigens nach, da das selbst gebaute Paket nicht signiert wurde und dem Client damit auch kein Schlüssel bekannt sein kann. Ob selbst gebaute Pakete signiert werden sollten, muss jeder Systembetreuer selbst entscheiden – für dieses Testszenario war es jedenfalls nicht erforderlich.

Wer an dieser Stelle noch weitere Informationen zum Paket und seinen Ablageort auf dem Repo-Server erfahren möchte, kann auf dem Client mit **apt-cache show test-deb-package** und **apt-cache policy test-deb-package** nachsehen.

Repo-Verwaltung ganz unkompliziert

Für die Ablage und Organisation von Paketen bietet reprepro mehr Möglichkeiten als den bloßen Paket-Import. Im Sinne des Stagings ist beispielsweise interessant, wenn ein Paket die Entwicklungsphase verlässt und dies auch im Debian-Repo widergespiegelt werden soll. In so einem Fall ist das Kopieren eines Pakets von der Devel- in die Testing-Distribution eine nützliche Funktion:

```
# reprepro -b /var/www copy xenuser-devel test-deb-package
Exporting indices...
```

reprepro erwartet hier zunächst die Ziel- und erst dann die Quell-Distribution, was für Neulinge erst einmal gewöhnungsbedürftig sein kann.

Das Paket ist nun sowohl in „xenuser-devel“, als auch in „xenuser-testing“ verfügbar. Wenn das Paket aus der Devel-Distribution verschwinden soll, kann es mit einem Kommando entfernt werden:

```
# reprepro -b /var/www remove xenuser-devel test-deb-package
Exporting indices...
```

Ein prüfender Blick auf „xenuser-devel“ mit **reprepro -b /var/www list xenuser-devel** zeigt keine Pakete mehr an – damit war das Entfernen erfolgreich.

Distribution unter reprepro einfrieren

Wer von Staging spricht, muss auch die Möglichkeit haben, einen Zustand im Repo einzufrieren. In der vorliegenden Distributionskonstellation ist dies recht simpel durch die Veränderung der Distributionskonfiguration möglich.

In diesem Beispiel wird davon ausgegangen, dass ein Systembetreuer einen Freeze für die Distribution „xenuser-testing“ ausgesprochen hat: Es sollen keine weiteren Pakete mehr aufgenommen oder gelöscht werden.

Für die Umsetzung dieser Anforderung wird die Konfigurationsdatei `/var/www/conf/distributions` geöffnet und die Distribution „xenuser-testing“ umkonfiguriert:

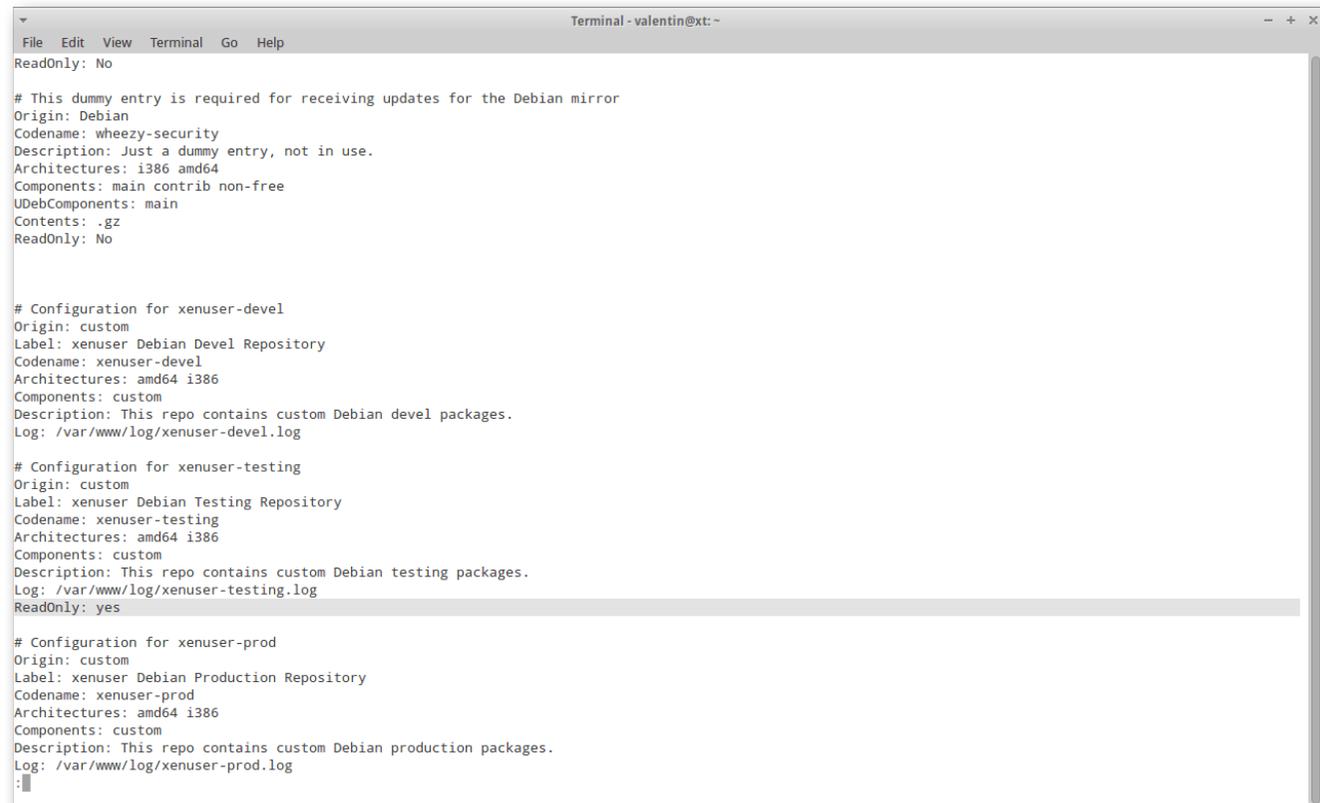
```
Origin: custom
Label: xenuser Debian Testing Repository
Codename: xenuser-testing
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian testing packages.
Log: /var/www/log/xenuser-testing.log
ReadOnly: yes # <- Neu
```

Das Freezing wird durch das Setzen von **ReadOnly: yes** herbei geführt. Beim Versuch ein Testpaket aus „xenuser-testing“ zu löschen erhält man nun erwartungsgemäß eine Fehlermeldung:

```
# reprepro -b /var/www remove xenuser-testing test-deb-package
Cannot remove packages from read-only distribution 'xenuser-testing'
There have been errors!
```

Tipps für den Alltag

Alleine die Tatsache, dass schon mit wenigen Handgriffen ein funktionsfähiges Debian-Repo mit Hilfe von reprepro erstellt werden kann, zeigt, wie mächtig dieses Werkzeug ist. Trotz der ausgereiften Funktionalität und der einfachen Handhabung kann es allerdings passieren, dass bspw. durch gestorbene Prozesse oder zu viele verschiedene Administratoren ein inkonsistenter Zustand im



```
Terminal - valentin@xt: -
File Edit View Terminal Go Help
ReadOnly: No

# This dummy entry is required for receiving updates for the Debian mirror
Origin: Debian
Codename: wheezy-security
Description: Just a dummy entry, not in use.
Architectures: i386 amd64
Components: main contrib non-free
UDebComponents: main
Contents: .gz
ReadOnly: No

# Configuration for xenuser-devel
Origin: custom
Label: xenuser Debian Devel Repository
Codename: xenuser-devel
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian devel packages.
Log: /var/www/log/xenuser-devel.log

# Configuration for xenuser-testing
Origin: custom
Label: xenuser Debian Testing Repository
Codename: xenuser-testing
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian testing packages.
Log: /var/www/log/xenuser-testing.log
ReadOnly: yes

# Configuration for xenuser-prod
Origin: custom
Label: xenuser Debian Production Repository
Codename: xenuser-prod
Architectures: amd64 i386
Components: custom
Description: This repo contains custom Debian production packages.
Log: /var/www/log/xenuser-prod.log
:
```

Blick in die Distributionskonfiguration von reprepro unter `/var/www/conf/distributions`. 🔍

Repo entsteht. Für solche Fälle helfen **reprepro -b /var/www clearvanished--delete** (löscht alle Paketdatenbanken und Pakete, die nicht mehr in `conf/distributions` referenziert werden) sowie **reprepro -b /var/www export <codename/distname>** (exportiert die Index-Dateien neu). Zudem gibt es mit **reprepro -b /var/www checkpool** die Möglichkeit, den Pool auf „verloren gegangene“ bzw. versehentlich gelöschten Pakete prüfen zu lassen.

Wer viele Pakete mit reprepro verwaltet, will womöglich Paketaktionen (wie Löschen) auf mehrere Pakete gleichzeitig anwenden. Für diesen Zweck bringt reprepro Selektionsfilter mit, die prinzipiell getestet werden können.

In diesem Beispiel beinhaltet eine eigene unter reprepro angelegte Distribution viele Pakete, die mit der Zeichenkette „freesw“ beginnen. Ziel ist es, diese Pakete aus dem Repo zu entfernen. Eine

Möglichkeit wäre, sich alle Pakete mit „freesw“ im Namen ausgeben und einzeln löschen zu lassen – effektiver ist jedoch die Anwendung von Filtern.

Sicherheitshalber wird dieser zunächst getestet:

```
# reprepro -b /var/www listfilter ↻
xenuser-prod 'Package (% freesw*)'
```

Ist die Auflistung erfolgreich, kann der Löschvorgang eingeleitet werden:

```
# reprepro -b /var/www removefilter ↻
xenuser-prod 'Package (% freesw*)'
```

Mehr zu den Selektionsfiltern findet sich in der Man-Page zu reprepro.

Neben dem Auflisten und Löschen von vielen Paketen mit einem Kommando deckt reprepro auch in anderen Bereichen die Anforderung ab, schnell und mit wenig Zeitaufwand viele Pakete verwalten zu müssen.

So kann beispielsweise reprepro mit **reprepro processincoming default** als CronJob automatisch dazu gebracht werden, Pakete aus einem bestimmten Verzeichnis zu importieren. Damit entfällt in vielen Fällen das manuelle Hinzufügen von Debian-Paketen mit **includedeb**. Wer dennoch gerne auf dieses Kommando zurück greift, kann übrigens auch Wildcards (wie **reprepro -b /var/wwwinclude deb freesw*.deb**) verwenden.

Die Eingabe des Basisverzeichnisses mit **-b /var/www** kann auf Dauer nerven – Abhilfe schafft der Export der Umgebungsvariable **REPREPRO_BASE_DIR**.

Fazit

reprepro ist ein mächtiges und einsteigerfreundliches Werkzeug für die Organisation eines eigenen Debian-Repos. Die Möglichkeit, mit wenigen Handgriffen auch den offiziellen Debian-Mirror zu spiegeln, macht reprepro auch für größere Server-Landschaften interessant, in denen Traffic und Bandbreite eingespart werden sollen.

Im Alltag mag man sich zwar ein bisschen mehr Dokumentation und HowTos auf der offiziellen Webseite wünschen, aber mit etwas Aufwand liefern Ausprobieren und Suchmaschinen genug Erkenntnisse, um schnell selbstständig mit reprepro arbeiten zu können.

reprepro muss allerdings nicht zwingend die Lösung der Wahl sein. Andere Programme bieten einen ähnlichen Funktionsumfang, sind aber nicht immer Teil des offiziellen Debian-Repos und benötigen damit etwas mehr Pflegeaufwand.

Leser, die sich generell mehr für Debian-Repositorys und deren Struktur interessieren, finden übrigens unter [4] geeignetes Lesematerial.

LINKS

[1] <https://wiki.debian.org/HowToSetupADebianRepository> 

[2] <http://mirrorer.alioth.debian.org/> 

[3] <http://mirrorer.alioth.debian.org/reprepro.1.html> 

[4] <https://wiki.debian.org/RepositoryFormat> 

Autoreninformation

Valentin Höbel ([Webseite](#)) setzt seit 2007 Debian in Unternehmen ein. In seiner Freizeit widmet er sich seinem Blog oder schreibt für verschiedene Print-Medien.

 [Teilen](#)

 [Kommentieren](#)



MY HOBBY: TAKING ADVANTAGE OF THE FACT THAT SOME SPECIFIC DREAMS ARE WEIRDLY COMMON, BUT NOT EVERYONE WHO HAS THEM REALIZES THIS.

“Brain Worms” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/719/>

Arduino von Franz-Josef Lücke

Der Artikel richtet sich an Anfänger, die Mikrocontroller und Arduino unter Linux verwenden wollen. Es werden die Grundlagen von Mikrocontrollern erläutert und die Hard- und Software der Arduino-Plattform vorgestellt. Zwei Beispiele zeigen die praktische Verwendung des Arduino.

Mikrocontroller

Ein Mikrocontroller ist „vollständiger“ Rechner in einem einzigen Chip. Er besteht aus einer CPU, einem flüchtigen Speicher, digitalen und analogen Ein- und Ausgängen sowie einer Schnittstelle zur externen Kommunikation.

In der Regel wird ein Mikrocontroller ohne ein Betriebssystem verwendet. So müssen auch die Standard-Aufgaben wie z. B. Speichern und Lesen von Dateien, Ausgabe von Texten auf einem Display oder das Laden von Treibern von dem Programm des Mikrocontrollers erledigt werden.

Wird ein Betriebssystem benötigt, oder sind aufwendige Interaktionen mit Benutzern notwendig, dann ist der Raspberry Pi [1] eine Alternative. Beim Raspberry Pi handelt es sich um einen vollwertigen kleinen Computer mit USB-Schnittstellen, Audio- und Display-Ausgängen und mit einem Linux-Betriebssystem (siehe auch freiesMagazin 06/2013 [2]). Darüber kann auf die digitalen Aus- und Eingänge zugegriffen werden. Er wird wie ein „richtiger“ Computer programmiert.

Zur Speicherung von Programmen verfügen viele Mikrocontroller über einen eingebauten Programmspeicher, der das Programm auch nach dem Abschalten der Versorgungsspannung behält. Dies wird in der Regel durch einen Flash-Speicher realisiert. Nach dem Anlegen der Versorgungsspannung wird das dort hinterlegte Programm automatisch ausgeführt. Das RAM wird vom Programm zur Ablage der Variablen und der Funktionsaufrufe verwendet. Er ist häufig klein im Vergleich zum Programmspeicher.

Über die digitalen Eingänge können binäre Informationen, z. B. von Schaltern, ausgelesen werden. Mit dem Ansteuern von digitalen Ausgängen können Vorgänge geschaltet werden. Durch Auslesen der analogen Eingänge können die Werte von Spannungen eingelesen werden, die z. B. von Temperatursensoren, Lichtsensoren ausgegeben werden. Über Ausgänge mit Pulsweitenmodulation (PWM) kann man digitale Impulse einer definierten Dauer ausgeben. Damit kann man einen Motor ansteuern oder es kann die Ausgabe eines analogen Signals realisiert werden. Dafür ist allerdings noch eine zusätzliche elektrische Beschaltung notwendig, die hier aber nicht weiter erläutert wird.

Zur Kommunikation mit der Umwelt, z. B. einem PC, wird häufig die serielle Schnittstelle eingesetzt. Die Erstellung und Übersetzung von Programmen erfolgt in der Regel auf einem PC. Die

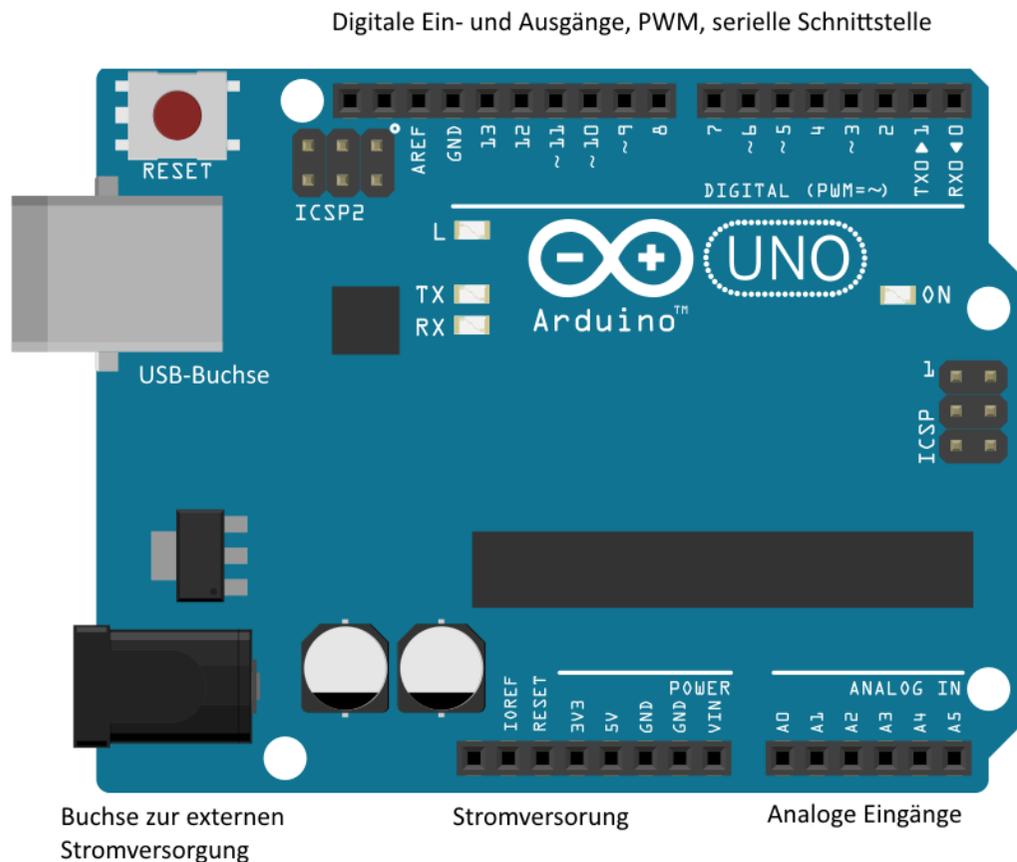
Übertragung eines Programms vom PC zum Mikrocontroller erfordert meistens über einen speziellen Programmieradapter, z. B. zur „In-System-Programmierung“ (ISP).

Solche Mikrocontroller und Programmieradapter können mit wenigen Bauteilen aufgebaut werden [3] [4]. Das Arbeiten mit solchen Mikrocontrollern erfordert dann allerdings Kenntnisse im Löten von Elektronik-Bauteilen. Zusätzlich werden auch fertige Module mit Mikrocontrollern angeboten, die allerdings auf den professionellen Einsatz zielen. Wegen der notwendigen Unterstützung übersteigen sie häufig den Preis, den Hobby-Nutzer bereit sind zu zahlen.

Arduino

Die Arduino-Plattform wurde im Jahre 2005 von Massimo Banzi und David Cuartelle als einfach zu verwendende und preisgünstige Plattform entworfen. Sie besteht aus einer Mikrocontroller-Platine und einer zugehörigen Entwicklungsumgebung. Die Hardware und die Software sind unter einer Open Source-Lizenz verfügbar [5] [6].

Ein wichtiges Merkmal der Arduino-Plattform ist die standardisierte und freie Hardware. Zur Kommunikation mit dem Rechner wird häufig die gebräuchliche USB-Schnittstelle verwendet. Dadurch wird zur Programmierung nur ein Rechner mit einer USB-Schnittstelle benötigt. Auf dem Rechner läuft die Entwicklungsumgebung, die



Arduino UNO. 

neben Linux auch für einige andere gebräuchliche Betriebssysteme verfügbar ist. Die Programmierung des Arduino erfolgt in der Programmiersprache C++ unter Verwendung von Bibliotheken. Das erleichtert die Programmierung und ermöglicht die schnelle Umsetzung von Ideen.

Als wichtiger Vorteil der Standardisierung sind unterschiedlich leistungsfähige Arduino verfügbar

neben Linux auch für einige andere gebräuchliche Betriebssysteme verfügbar ist. Die Programmierung des Arduino erfolgt in der Programmiersprache C++ unter Verwendung von Bibliotheken. Das erleichtert die Programmierung und ermöglicht die schnelle Umsetzung von Ideen.

Für die ersten Experimente sind einige Leuchtdioden, Widerstände, Kabel und ein Steckbrett sinnvoll.

(siehe Tabelle „Eigenschaften ausgewählter Arduino“).

Ein weiteres wichtiges Merkmal ist die standardisierte Schnittstelle für die Ein- und Ausgabe, die auch durch Aufsteckplatinen genutzt werden kann. Diese werden Shields genannt (s. Tabelle „Ausgewählte Erweiterungskarten“). Durch die Verfügbarkeit der passenden Bibliotheken ist die Verwendung der Aufsteckplatinen in den eigenen Programmen einfach möglich.

Ausrüstung

Für die ersten Projek-

Auf dem Linux-Rechner wird über die Paketverwaltung das Paket Arduino installiert. Dieses enthält auch die Entwicklungsumgebung.

Einrichten

Vor der ersten Verwendung muss der Arduino gegebenenfalls auf die verwendete Spannungsversorgung eingestellt werden. Er sollte über die USB-Schnittstelle mit 5 V Spannung versorgt werden. Die genaue Einstellung beschreibt die Dokumentation des Arduino.

Programmierung

Einstellungen

Über den Befehl **Arduino** im Terminal oder über das Menü wird die Entwicklungsumgebung gestartet. Ihre wichtigsten Funktionen sind über die Schaltflächen erreichbar. Nach Betätigung der Schaltfläche „Überprüfen“ wird die Syntax des Programms überprüft und das Programm übersetzt. Syntaxfehler werden in einem Fenster der Entwicklungsumgebung angezeigt. Durch Betätigung der Schaltfläche „Upload“ wird das übersetzte Programm vom Rechner in den Speicher des Arduino übertragen.

In der Entwicklungsumgebung muss zunächst der verwendete Arduino über das Menü „Tools → Board“ ausgewählt werden. Anschließend wird der Arduino per USB-Kabel mit dem PC verbunden. In der Entwicklungsumgebung wird über das Menü „Tools → Serieller Port“ der verwendete USB-Anschluss selektiert, z. B. `/dev/tty/USB0`. Kann er nicht identifiziert werden, dann sollte das

Eigenschaften ausgewählter Arduino

Name	Prozessor	RAM	Programmspeicher	Besonderheit	Analoge Ein-/Ausgänge	Digitale Ein-/Ausgänge PWM
Uno	8 Bit	2 KByte	32 KByte	Für Einsteiger	6/0	14/6
Leonardo	8 Bit	2 KByte	32 KByte	Mehr Ein- und Ausgänge	12/0	20/7
Micro	8 Bit	2,5 KByte	32 KByte	Kleine Abmessungen	12/0	20/7
Ethernet	8 Bit	2 KByte	32 KByte	Mit Ethernet-Anschluss	6/0	14/4
Due	32 Bit	96 KByte	512 KByte	Schnell	12/2	54/12

Wird das Programm auf dem Arduino ausgeführt und leuchten die Leuchtdioden nicht, dann sollten die Anschlüsse der Leuchtdiode getauscht werden.

USB-Kabel abgezogen werden. Anschließend wird das Menü erneut geöffnet. Danach wird das USB-Kabel wieder eingesteckt und das Menü wieder geöffnet. Der „neu“ hinzugekommene Eintrag ist der gesuchte Anschluss. Dieser Eintrag muss ausgewählt werden.

Programmaufbau

Ein Arduino-Programm wird auch als Sketch bezeichnet. Es besteht zunächst aus den Funktionen `void setup()` und `void loop()`. Die Funktion `void setup()` wird einmal nach dem Programmstart durchlaufen. Dort werden alle Initialisierungen durchgeführt, z. B. das Festlegen der Ein- und Ausgänge. Anschließend wird vom Arduino-Framework die Funktion `void loop()` zyklisch in einer Endlosschleife aufgerufen. Dort werden die zyklischen Abläufe programmiert. Selbstverständlich können aus diesen beiden Funktionen auch eigene Funktionen oder auch Funktionen der Arduino-Bibliothek aufgerufen werden.

Anwenden

Die Entwicklungsumgebung liefert sehr viele Beispiele mit. Sie können über „Datei → Beispiele“

ausgewählt werden. Die notwendige Beschaltung ist im Internet auf den Arduino-Seiten unter „Examples“ zu finden [7].

Leuchtdiode blinken lassen

Das vorliegende Programm basiert auf dem Beispiel Blink. In diesem Beispiel werden zwei Leuchtdioden im regelmäßigen Rhythmus ein- und ausgeschaltet.

Verdrahtung

Ein Ende eines Widerstands mit 220 Ohm wird mit dem Anschluss 13 des Arduino verbunden. Das andere Ende des Widerstandes wird mit dem angewinkelten Anschluss der roten Leuchtdiode (= Anode) verbunden. Das freie Ende der roten Leuchtdiode wird mit dem Anschluss GND des Arduino verbunden.

Ein Ende eines weiteren Widerstands mit 220 Ohm wird an dem Anschluss 12 des Arduino angeschlossen. Das andere Ende dieses Widerstandes wird mit dem angewinkelten Anschluss der grünen Leuchtdiode (= Anode) verbunden. Das freie Ende der grünen Leuchtdiode wird ebenfalls mit dem Anschluss GND des Arduino verbunden.

Programm

Eine Leuchtdiode blinkt

Zur Programmierung wird die Arduino-Entwicklungsumgebung gestartet. Über die Menüs wird das Beispielprogramm Blink („Datei → Beispiele → 01.Basics → Blink“) ausgewählt. Danach wird der Code in der Entwicklungsumgebung angezeigt. Nach dem Übersetzen und dem Übertragen wird der Code auf dem Arduino ausgeführt. Die rote Leuchtdiode blinkt im Takt einer Sekunde.

Zwei Leuchtdioden blinken

Nun wird ein neues Programm durch „Datei → Neu“ angelegt. In das neue Fenster der Entwicklungsumgebung wird der folgende Code eingefügt:

```

/*
  Blinken zweier Leuchtdioden
  Schaltet beide Leuchtdioden für eine
  Sekunde ein und dann für eine Sekunde
  aus.
  Das Ganze wird endlos wiederholt.

  Der Code ist public domain.
*/

```

```

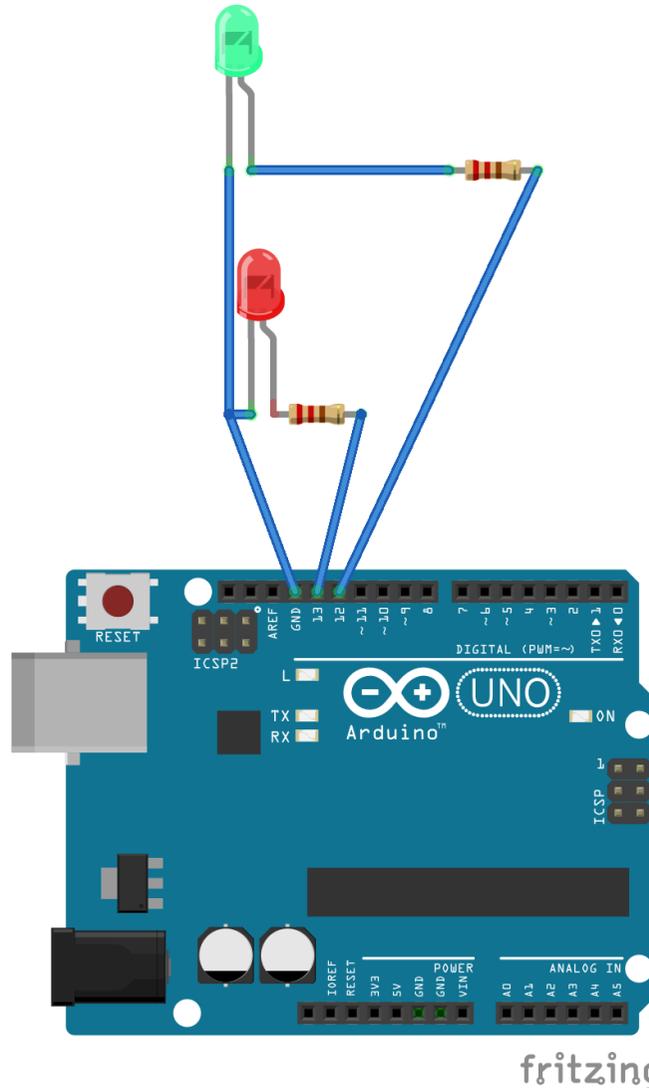
// An Pin 13 (digitaler Ausgang) ist die
// rote Leuchtdiode angeschlossen
int outLedRed = 13;

// An Pin 12 (digitaler Ausgang) ist die
// grüne Leuchtdiode angeschlossen
int outLedGreen = 12;

// Die Funktion setup() wird einmal nach
// dem Programmstart ausgeführt
void setup() {
  // Konfiguration der digitalen Ports
  // als Ausgänge
  pinMode(outLedRed, OUTPUT);
  pinMode(outLedGreen, OUTPUT);
}

// Die Funktion loop() wird zyklisch und
// endlos ausgeführt
void loop() {
  // Schaltet die rote Leuchtdiode ein
  digitalWrite(outLedRed, HIGH);
  // Schaltet die grüne Leuchtdiode ein
  digitalWrite(outLedGreen, HIGH);
  // Wartet eine Sekunde
  delay(1000);
  // Schaltet die rote Leuchtdiode aus
  digitalWrite(outLedRed, LOW);
  // Schaltet die grüne Leuchtdiode aus
  digitalWrite(outLedGreen, LOW);
  // Wartet eine Sekunde
  delay(1000);
}

```

Listing 1: *blinken_v2*Anschluss der Leuchtdioden an einen Arduino Uno. 

Anschließend wird der Code übersetzt und zum Arduino übertragen. Danach blinken beide Leuchtdioden im Takt einer Sekunde.

Nun soll das Beispiel modifiziert werden: Die rote Leuchtdiode soll leuchten und die grüne Leuchtdiode soll nicht leuchten. Nach einer Sekunde soll die rote Leuchtdiode nicht leuchten und die grüne Leuchtdiode soll leuchten. Dazu muss die Funktion `loop()` modifiziert werden.

```

// Die Funktion loop() wird zyklisch und
// endlos ausgeführt
void loop() {
  // Schaltet die rote Leuchtdiode ein
  digitalWrite(outLedRed, HIGH);
  // Schaltet die grüne Leuchtdiode aus
  digitalWrite(outLedGreen, LOW);
  // Wartet eine Sekunde
  delay(1000);
  // Schaltet die rote Leuchtdiode aus
  digitalWrite(outLedRed, LOW);
  // Schaltet die grüne Leuchtdiode ein
  digitalWrite(outLedGreen, HIGH);
  // Wartet eine Sekunde
  delay(1000);
}

```

Listing 2: *Funktion loop()*

Nach dem Übersetzen und dem Übertragen zum Arduino leuchten die beiden Leuchtdioden wie gewünscht.

Weiter

In den nächsten Schritten kann das Beispiel weiter geändert und getestet werden. Auch die mitgelieferten Beispiele der Entwicklungsumgebung können natürlich ausprobiert und modifiziert werden.

Ausgewählte Erweiterungskarten

Erweiterungskarte	Funktion
Arduino GSM Shield	Empfang und senden von SMS per Mobilfunk (GSM), benötigt zusätzlich eine SIM-Karte
Arduino Ethernet Shield	Erweiterung um eine Ethernet-Schnittstelle
Arduino WiFi Shield	Erweiterung um eine drahtlose (W-LAN) Schnittstelle
Arduino Motor Shield	Ansteuerung eines zweier Gleichstrommotoren oder eines Schrittmotors
Arduino Proto Shield	Platine zum Aufbau eigener Erweiterungen

Im Internet sind viele Anwendungen für Arduino verfügbar, z. B. eine Audio-Pegelanzeige, die automatische Fütterung von Fischen oder ein Mail-Server [6] [3] [4].

Zusammenfassung

Ein Arduino ist eine standardisierte Mikrocontroller-Plattform für den einfachen und schnellen Einstieg in die Welt der Mikrocontroller. Durch die angebotenen

Erweiterungen kann unkompliziert eine Vielzahl von Anwendungen realisiert werden. Im Internet werden sehr viele Projekte beschrieben.

LINKS

- [1] https://de.wikipedia.org/wiki/Raspberry_Pi
- [2] http://www.freiesmagazin.de/mobil/freiesMagazin-2013-06.html#fm_13_06_RaspberryPi
- [3] <http://www.mikrocontroller.net/>

- [4] <http://www.roboternetz.de/>
- [5] <https://de.wikipedia.org/wiki/Arduino-Plattform>
- [6] <http://Arduino.cc/> 
- [7] <http://http://arduino.cc/en/Tutorial/Blink> 

Autoreninformation 

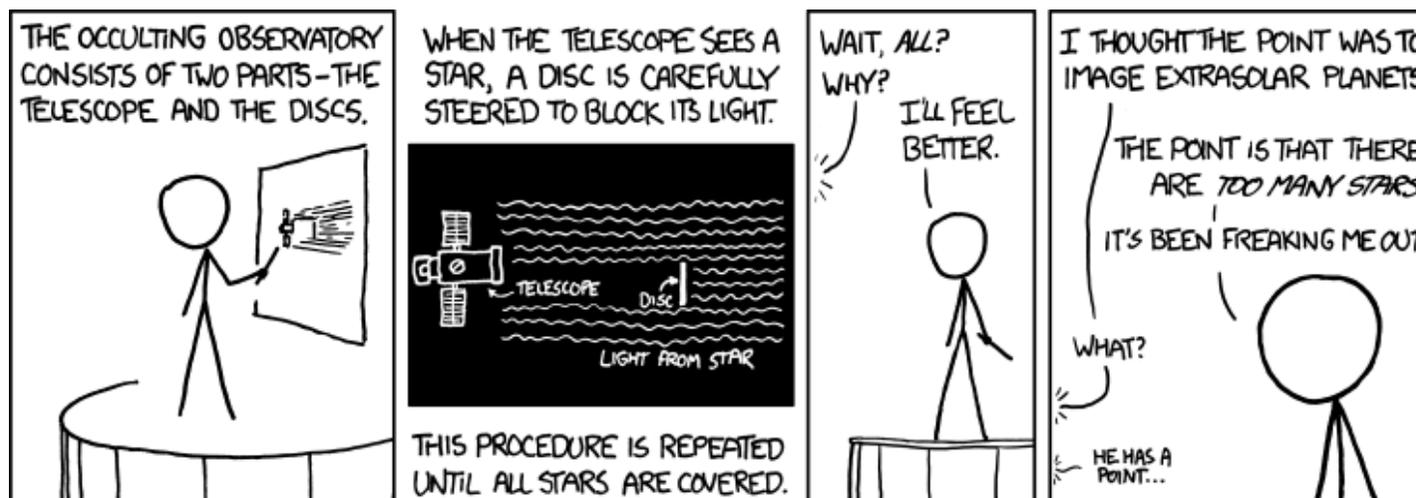
Franz-Josef Lücke ist Software-Entwickler für Embedded-Systeme, d. h. er freut sich immer, etwas zu bewegen. Linux setzt er seit dem Kernel 2.0 immer wieder ein und betreibt damit vor allem ältere Notebooks, die dadurch ein zweites Leben gewinnen.



Teilen



Kommentieren



“Occulting Telescope” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/975/>

Bericht: Herbsttagung DANTE e.V. 2014 in Karlsruhe von Dominik Wagenführ

Jedes halbe Jahr trifft sich die Deutschsprachige Anwendervereinigung T_EX e.V. (kurz DANTE [1]), um die Mitgliederversammlung abzuhalten und dabei gleichzeitig auch noch interessante Vorträge rund um das Thema T_EX anzubieten. Dieses Jahr fand die Herbsttagung in Karlsruhe statt.

Vorabendtreff

Bevor es zum vergnüglichen Teil ging, traf sich der Vorstand des Dante e.V. zur halbjährlichen Sitzung. Da ich das erste Mal als Beisitzer den Diskussionen beiwohnen konnte, war es sehr interessant, etwas mehr Einblick in die internen Vorgänge und Abläufe zu erhalten. Vor allem einige kontroverse Themen wurden sehr hitzig und spannend diskutiert.

Nach dreieinhalb Stunden waren die Getränke alle und wir erschöpft, sodass wir uns nach einer kurzen Pause gemeinsam zum Abendtreff aufmachen konnten. Im Vogelbräu in der Südstadt trafen sich ca. 25 Teilnehmer der Tagung, was auf eine rege Teilnahme am Folgetag hoffen ließ.

Sehr schön war, dass man viele neue und vor allem junge Gesichter sehen konnte, die sich dann hoffentlich auch aktiv bei DANTE e.V. einbringen werden, sodass die älteren Semester irgendwann einmal in den verdienten Ruhestand gehen und das Zepter an eine neue Generation übergeben können.

Tagungstag

Nach einer kleinen Jogging-Runde durch den Zoo bzw. am Zoo vorbei und einem ausgiebigen Frühstück machten wir uns als kleine Gruppe vom Hotel zur Tagungsstätte auf. Kurz nach 9 Uhr konnte der Vorsitzende des DANTE e.V., Martin Sievers, die ca. 30 Zuhörer im Architekturgebäude des Karlsruhe Institut für Technologie (KIT) begrüßen.

Danach hielt Prof. Dr. Lohner vom Steinbuch Centre for Computing (SCC) das Grußwort und stellte die Universität und vor allem das SCC und dessen Aufgaben ausführlich vor. Die Menge an Daten, die verarbeitet werden, bzw. die vorgehaltene Rechnerleistung war dabei sehr beeindruckend.

Nach der kurzen Ansprache eröffnete Martin Sievers den offiziellen Teil der Tagung mit der 51. Mitgliederversammlung. Von den Mitgliedern wurden verschiedene Eindrücke vergangener Tagungen geteilt und auch auf zukünftige T_EX-Tagungen aufmerksam gemacht. An dieser Stelle sei vor allem auf die kommende Frühjahrstagung des DANTE e.V. im wunderschönen Stralsund an der Ostsee hingewiesen, die Mitte April 2015 stattfindet. Daneben wurden auch die abgeschlossenen und laufenden Projekte vorgestellt. Genauere Details kann man der aktuellen DTK entnehmen.

Den ersten Vortrag des Tages hielt Markus Kohm zum Thema „Tipps und Tricks mit KOMA-Script“. Er stellte in dem sehr interessanten Vortrag das

Paket `scrlayer` zum Definieren von Ebenen vor, auf denen man beispielsweise Logos, Versionsinformationen oder Ähnliches im Dokument auf allen oder manchen Seiten anzeigen lassen kann. Daneben beschrieb er noch, wie man mithilfe von `\addchaptertocentry` eigens definierte Kapitelbeschreibungen vor dem Inhaltsverzeichnis definieren kann.

Danach unterhielt Heiner Richter vom Centrum für bessere Übergänge und Studienbedingungen (kurz CÜS) der Fachhochschule Stralsund [2] die Zuhörer und zeigte einmal, wie das Bild auf der reinen Anwenderseite von T_EX aussieht. Sein Wunsch war es, zwei getrennte Texte vertikal parallel zueinander setzen zu können. Daneben standen auch noch doppelseitige Bilder, farbige Informationsboxen und QR-Codes auf seiner Wunschliste. Bei allen Problemen probierte er viel aus und fragte die Profis, wenn er nicht weiter wusste. Seine Ausführungen zeigten, dass man mit etwas Durchhaltevermögen, Neugier und – wie er es nannte – Frechheit auch als normaler Anwender schwierigere Probleme mit T_EX lösen kann.

Das schöne Wetter in Karlsruhe am Samstag lud vor und nach dem Mittagessen zu einem kleinen Spaziergang am Schloss entlang ein. Danach zeigte Uwe Ziegenhagen auf unterhaltsame Weise, welche neuen oder interessanten Pakete es auf CTAN gibt, die für die Allgemeinheit ggf. interessant sein könnten. Darunter war dann auch

das **ctable**-Paket für den Satz von Tabellen, das bereits in der DTK 1/2014 („Explizite Positionierung in L^AT_EX“) erwähnte Paket **grid-system** zum Setzen von mehrspaltigen Textblöcken oder die Erstellung von Flussdiagrammen mit **flowchart**. Solche Diagramme kann man auch mit Graphviz [3] über das Paket **gaphviz** direkt aus L^AT_EX heraus erzeugen.

Gleich danach stellte Herbert Voss ConT_EXt [4] vor, wobei er sich vor allem auf die lokale Installation außerhalb von T_EX Live bezog und zeigte, wie dies prinzipiell möglich ist. Das gestartete ConT_EXt-Buch von Herbert Voss wird es nach seinen Aussagen erst einmal nicht geben, weil die ConT_EXt-Entwicklung noch sehr aktiv betrieben wird und so nach drei Monaten die verwendeten Beispiele nicht mehr korrekt durchlaufen.

Nach einer kurzen Pause, in der die Tagungsteilnehmer von den Veranstaltern sehr großzügig mit diversen Kuchen eingedeckt wurden – die das Abendessen schon fast hinfällig machten – stellte Benjamin Berg das Open-Source-Projekt SDAPS [5] vor. Es handelt sich dabei um ein Optical Mark Recognition Programm (OMR [6]), welches u. a. mit der Hilfe von L^AT_EX und der eigens definierten Dokumentenklasse **sdaps** ein Umfrageformular erstellt, welches nach dem Ausdrucken, manuellen Ausfüllen und Einscannen die angekreuzten Antworten erkennt und auswertet. Die Möglichkeiten, die SDAPS bietet und die von Benjamin Berg live vorgeführt wurden, sahen sehr interessant aus.

Zum Abschluss grub Wolfram Schwenzer etwas in der Historienkiste von T_EX und holte die Programmiersprache WEB [7] heraus, die 1983 von Donald Knuth entwickelt wurde. Die Nachfrage beim Publikum, wer diese wohl kennt, zeigte bei der Meldung von nur drei Alteingesessenen, dass es sich hier um ein Urgestein des Computerzeitalters handelte. Mit Noweb [8] gab es 1989 eine Weiterentwicklung von Norman Ramsey. Hiermit war es möglich, Programmdokumentation direkt im Code zu verfassen. Diese Dokumentation kann in HTML oder eben auch T_EX stattfinden, wodurch man auch komplexe, mathematische Algorithmen gut beschreiben kann.

Der Abendtreff fand im Pizzahaus statt, welches sehr leckere italienische Gerichte anbot und allen einen guten Ort für Gespräche bot.

Touristisches Rahmenprogramm

Am Sonntag fand noch ein Stadtrundgang unter dem Motto „Architektur & Stadtplanung“ statt. Das Motto war ganz passend, da in der Südstadt von Karlsruhe an sehr vielen Stellen gebaut wird, so dass beispielsweise der Marktplatz aktuell nicht mehr als solcher zu erkennen ist. Ich habe das Programm aber ausgelassen und mich bereits am Morgen auf den Heimweg gemacht.

Insgesamt war es wieder eine sehr interessante Veranstaltung mit vielen alten (nicht zwingend aufs Alter bezogen) und auch ein paar neuen Gesichtern, die hoffentlich auch bei späteren Veranstaltungen wieder zu sehen sein werden. Vor

allem die Vorträge zu SDAPS und **scrlayer** waren für mich sehr interessant, auch wenn ich sie voraussichtlich nie wirklich nutzen werde.

Wie immer gilt: Wer T_EX (in jeglicher Form) nutzt und Interesse daran hat, kann auch als Anfänger bei den DANTE-Veranstaltungen immer etwas mitnehmen. Die nächsten Tagungen werden in Stralsund (DANTE-Frühjahrstagung 2015), Darmstadt (TUG 2015) und Graz (DANTE-Herbsttagung 2015) stattfinden und sicherlich trifft man sich irgendwo dort wieder.

LINKS

- [1] <http://www.dante.de/>
- [2] <https://cues.fh-stralsund.de/>
- [3] <http://graphviz.org/> 
- [4] <http://wiki.contextgarden.net/> 
- [5] <http://sdaps.org/> 
- [6] http://de.wikipedia.org/wiki/Optical_Mark_Recognition
- [7] [http://de.wikipedia.org/wiki/Web_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Web_(Programmiersprache))
- [8] <http://de.wikipedia.org/wiki/Noweb>

Autoreninformation

Dominik Wagenführ ([Webseite](#)) ist Chefredakteur von **freiesMagazin**, das mit L^AT_EX gesetzt wird. Daneben ist er im Vorstand des DANTE e.V.



Teilen



Kommentieren

Rezension: Seven Web Frameworks in Seven Weeks von Jochen Schnelle

In der Webentwicklung gehört heutzutage der Einsatz eines Webframeworks (also einer Web-Bibliothek) zum Standard, sowohl serverseitig als auch clientseitig. Das vorliegende Buch „Seven Web Frameworks in Seven Weeks“ gibt dazu einen Überblick über sieben verschiedene Frameworks mit unterschiedlichen Stärken und Einsatzgebieten.

Das Buch ist das dritte im Bunde der „... in Seven Weeks“-Serie. Das erste behandelt Programmiersprachen, das zweite Datenbanken (Buchbesprechungen siehe freiesMagazin 7/2011 [1] und freiesMagazin 9/2012 [2]). Das Konzept des Buchs ist dabei beibehalten worden: Jedes Webframework wird in drei Abschnitten anhand eines Praxisbeispiels vorgestellt, das Konzepte und Stärken vermittelt. Am Ende jedes Abschnitts gibt es dann noch Tipps für eigene „Experimente“ und weiterführende Dokumentation.

Wie Sand am Meer

Nun ist die Ausgangssituation hier etwas schwieriger als bei Programmiersprachen und Datenbanken, da es Webframeworks gibt wie Sand am Meer. Von daher ist es auch gar nicht der Anspruch der Autoren, eine repräsentative Übersicht zu schaffen, sondern Frameworks für spezielle Probleme bzw. Aufgaben vorzustellen.

Aus diesem Grund findet man in diesem Buch auch keine Kapitel über „Mainstream“-Frame-

works wie z. B. Ruby on Rails, Django oder Node.js, sondern eher über spezialisiertere Bibliotheken.

Über die Autoren

Das Buch wurde von Jack Moffit, Senior Research Engineer bei Mozilla, und Fred Daoud, Webentwickler und Autor von diversen anderen Büchern zum Thema Webentwicklung, geschrieben. Gute Voraussetzungen also für ein fundiertes Buch über das Thema Frameworks.

Die Webframeworks

Wie der Titel schon sagt, werden in dem Buch sieben Frameworks vorgestellt, fünf für die server- und zwei für die clientseitige Programmierung.

Letztere sind die beiden JavaScript-Frameworks CanJS und AngularJS, die aber völlig unterschiedliche Ansätze verwenden. Während CanJS eher den klassischen Ansatz mit Model-View-Controller verfolgt, hat AngularJS einen deklarativen Ansatz. Im Rahmen des Buchs wird die gleiche Applikation jeweils mit beiden Bibliotheken entwickelt, so dass die individuellen Ansätze und Stärken gut sichtbar werden.

Die im Buch behandelten serverseitigen Frameworks sind das Ruby-basierte Sinatra, das in Erlang geschriebene Webmachine, das Haskell-Framework Yesod sowie die Clojure-basierten Webbibliotheken Ring und Immutable.

Wie bereits erwähnt werden dabei die jeweils individuellen Stärken herausgestellt: mit Sinatra wird serverseitig eine REST-API realisiert, Ring dient als Basis für die meisten anderen Clojure-basierten Webframeworks, mit Webmachine lassen sich sehr einfach HTTP-basierte APIs entwickeln, Yesod nutzt die statische Typisierung und den Compiler von Haskell, um (Programmier-) Fehler schon während der Entwicklung zu finden und zu eliminieren und Immutable bietet sehr einfach verteilte Caches, Message Queues und Scheduling und das ganze auch über mehrere Server hinweg.

Beispiele und Code

Im Buch wird mit jedem Framework eine Applikation entwickelt, die die jeweiligen Stärken der Bibliothek gut zeigt. Für die fünf serverseitigen Webframeworks sind dies somit (völlig) verschiedene Applikationen.

Alle relevanten Codeteile sind im Buch vollständig abgedruckt, sodass man das Buch auch problemlos „offline“ lesen kann. Den kompletten Code kann man aber auch im Internet herunterladen.

Wie gut man selber als Leser die Codebeispiele versteht, hängt dann in erster Linie von den eigenen Kenntnissen der jeweiligen Programmiersprachen ab, da diese – Ruby, Clojure, Erlang, Haskell und JavaScript – sich schon erheblich unterscheiden. Polyglotte Programmierer sind hier

Buchinformationen

Titel	Seven Web Frameworks in Seven Weeks [3]
Autor	Jack Moffit, Fred Daoud
Verlag	O'Reilly, 2013
Umfang	302 Seiten
ISBN	978-1937785635
Preis	ca. 22 Euro (broschiert)

also im Vorteil. Seitens der Autoren sind alle Beispiele gut, verständlich und kompakt erklärt, so dass von dieser Seite aus eigentlich keine Fragen offen bleiben.

Sprache und Verständlichkeit

Das Buch ist insgesamt sehr verständlich geschrieben, es bleiben, wie oben bereits erwähnt, keine Fragen offen. Das Buch liest sich recht interessant und kurzweilig, zumal die Theorie minimiert und alles sehr praxisbezogen ist.

Die Sprache des Buchs ist Englisch. Die eigenen Englischkenntnisse sollten dabei doch etwas über das normale Schulenglisch hinaus gehen, damit man den Inhalt gut nachvollziehen und verstehen kann.

Fazit

Das vorliegende Buch „Seven Web Frameworks in Seven Weeks“ ist kein „klassisches“ IT-Buch, in dem Wissen vermittelt/gelehrt wird. Vielmehr möchte das Buch den Leser dazu animieren, auch über den Tellerrand zu schauen und zu realisieren, dass sich mit dem richtigen Webframework bestimmte Aufgabenstellungen schneller, besser oder auch effizienter realisieren lassen.

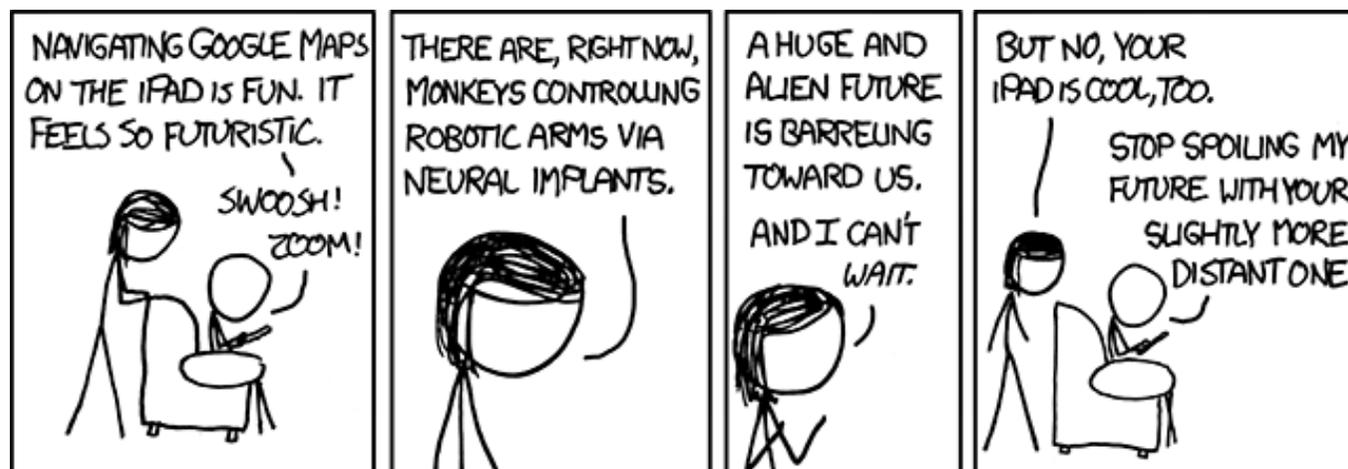
Wer als Webentwickler arbeitet oder sich für Webentwicklung interessiert, dem kann das Buch jedenfalls uneingeschränkt empfohlen werden.

LINKS

- [1] <http://www.freiesmagazin.de/20110703-juliausgabe-erschienen>
- [2] <http://www.freiesmagazin.de/20120902-septemberausgabe-erschienen>
- [3] <http://shop.oreilly.com/product/9781937785635.do>

Autoreninformation

Jochen Schnelle ([Webseite](#)) entwickelt zwar bevorzugt mit Python (und Python Webframeworks), blickt aber auch gerne über den Tellerrand auf andere Programmiersprachen und Frameworks.

 [Teilen](#) [Kommentieren](#)

“iPad” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/728/>

Rezension: Spieleprogrammierung mit Android Studio von Sujeevan Vijayakumaran

Das Buch „Spieleprogrammierung mit Android Studio“ vom Autor Uwe Post erschien 2014 in der ersten Auflage. Auf fast 400 Seiten erlernt man sowohl den Umgang mit Android Studio als auch die Spieleprogrammierung unter Android.

Redaktioneller Hinweis: Wir danken Galileo Computing für die Bereitstellung eines Rezensionsexemplares.

Was steht drin?

Das Buch ist in elf Kapitel unterteilt und umfasst 381 Seiten. Der Autor ist Chefentwickler in einer Firma für Smartphone-Spiele. Veröffentlicht wurde das Buch vom Verlag Galileo Computing.

Das erste Kapitel ist die „Einführung“, in der ein kurzer Überblick über Spiele-Klassiker gegeben wird. Im darauf folgenden zweiten Kapitel erfolgt die erste Einführung in Android Studio, in der nicht nur die Installation kurz erläutert wird, sondern auch einige Tipps zur Arbeitserleichterung gegeben werden. Danach folgt auch schon das erste kleine Spiel, welches im dritten Kapitel erstellt und erklärt wird. Das aufgeführte Spiel wird dabei Schritt für Schritt aufgebaut. Der Fokus liegt in diesem Kapitel auf den Grundlagen eines Android-Projekts unter Android Studio.

Im vierten Kapitel folgt erstmals der Einsatz von Bildbearbeitungsprogrammen. In diesem Buch

wird dabei grundsätzlich Gimp genutzt. Weiterhin werden Animationen behandelt, sodass man sinnvolle Animationen in Spielen einbauen kann.

Smartphones besitzen grundsätzlich sehr viele Sensoren, die man sehr gut für Spiele nutzen kann. Das fünfte Kapitel behandelt daher die Nutzung des Lagesensors. Im sechsten Kapitel folgt dann die Nutzung von Audio und Musik-Stücken.

Eine relativ neue Art von Spielen sind solche, die auf Geolokalisation beruhen. Im siebten Kapitel wird ein kleines Spiel programmiert, in dem man sich frei herumbewegen muss. Die Lokalisation erfolgt mittels GPS.

Die letzten Kapitel des Buchs vertiefen die Art und Weise der Spiele sowie deren Komplexität. Im achten Kapitel wird die Bibliothek libgdx verwendet, um ein schnelles 2-D-Spiel zu erzeugen, während es im neunten Kapitel an die Königsdisziplin geht: 3-D-Spiele. Auch für 3-D-Spiele wird die Open-Source-Bibliothek libgdx verwendet. Die Beschreibungen und Erläuterungen gehen relativ weit, sodass man nicht nur ein Szenenbild aufbaut, sondern auch ein kleines 3-D-Labyrinth erstellt.

Die letzten beiden Kapitel beziehen sich nicht mehr direkt auf die Spiele-Programmierung selbst. Das zehnte Kapitel beschäftigt sich mit Social-Gaming, also wie man den Spielstand seines

Spiels in Google Play Games Services einbindet oder auch wie man einen Facebook-Login realisiert. Das letzte Kapitel behandelt eher den wirtschaftlichen Aspekt von Spielen, nämlich die Monetarisierung. Der Autor gibt einige Tipps, welche Arten von Monetarisierung es gibt, und wie man sie sinnvoll einsetzen kann, um im Idealfall gutes Geld zu verdienen.

Wie liest es sich?

Das Buch liest sich durchweg sehr gut. Der Autor entwickelt in jedem Kapitel ein eigenes Spiel, welches zum Thema passt. Die Erläuterungen sind meistens nicht zu kurz, aber auch nicht sonderlich lang. Nach dem einen oder anderen Code-Schnipsel folgt meist eine prägnante Erklärung, was genau getan wird.

Für dieses Buch sind nicht nur gute Java-Kenntnisse Pflicht; auch ein grundsätzliches Verständnis der Programmierung von Android-Apps ist hier sinnvoll. Installationsanleitungen sind kurz gehalten, die erste Installation von Android Studio wird nicht im Detail erklärt, da dies Leser des Buches sowieso können sollten. Praktisch ist vor allem, dass alle Anwendungen im Buch Open-Source-Anwendungen sind. Neben Gimp wird auch Inkscape kurz thematisiert. Der Autor selbst nutzt nach eigenen Angaben Ubuntu, das Buch ist allerdings nicht auf Ubuntu fokussiert, sondern behandelt die drei Betriebssysteme Linux, Windows und Mac OS X zu gleichen Anteilen.

Buchinformationen

Titel	Spielprogrammierung mit Android Studio [1]
Autor	Uwe Post
Verlag	Galileo Computing, 2014
Umfang	381 Seiten
ISBN	978-3-8362-2760-5
Preis	34,90 € (gebunden), 29,90 € (E-Book)

Kritik

Das Buch bietet einen guten Einstieg in die Programmierung von Spielen für Android. Die Erklärungen sind durchgängig gut zu verstehen und vieles lässt sich einfach nachvollziehen. Man merkt, dass der Autor viel Erfahrung in der Spieleprogrammierung besitzt. Es kommen durchaus viele Hinweise, wenn einige Tools oder Verfahren umständlich oder kompliziert sind. Die Schreibweise von Uwe Post ist zudem angenehm zu lesen.

Für knapp 35 € bekommt man daher ein gutes Buch über die Android-Spielprogrammierung. Der Leser gewinnt dabei einen sehr guten Einblick in die Welt der Spiele, womit man dann schon fast sein erstes Spiel programmieren und veröffentlichen kann. Das Buch lässt sich insbesondere für diejenigen empfehlen, welche gute Erfahrung in der Android-Entwicklung besitzen und in die Spiele-Programmierung einsteigen wollen.

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

„Android-Studio wird in Zukunft die Standard-IDE für die Android-Entwicklung sein, doch welches IDE wird es ablösen?“

Die Antwort kann bis zum **12. Oktober 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 beantworten konnten, verlost.

LINKS

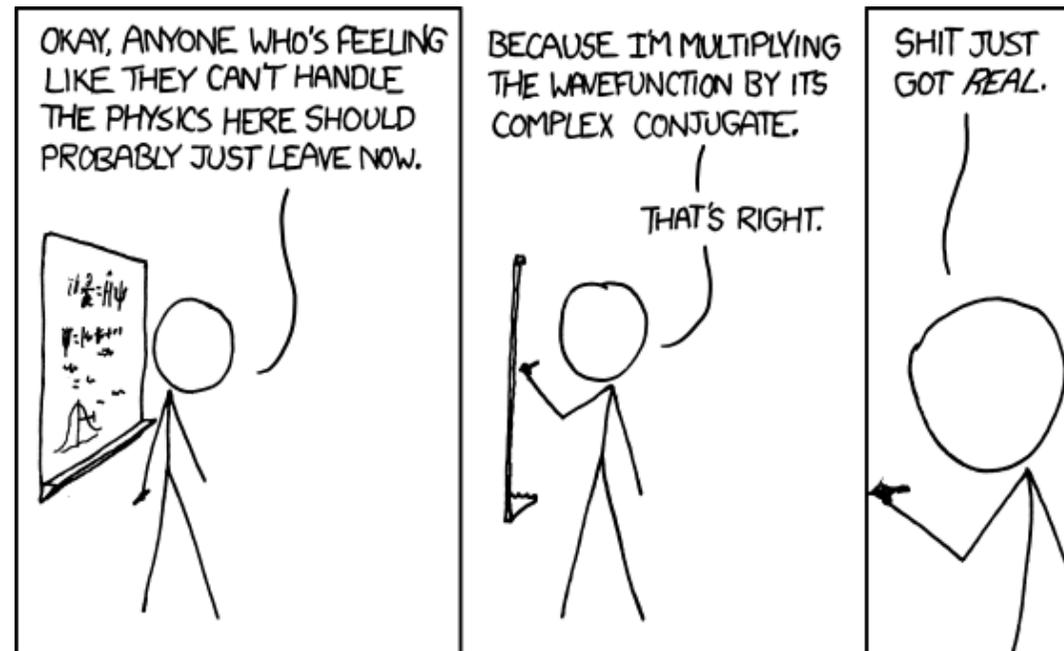
[1] https://www.galileo-press.de/spielprogrammierung-mit-android-studio_3537/

Autoreninformation

Sujeevan Vijayakumaran ([Webseite](#)) gewann mit dem Buch einen ersten Einblick in die Spieleprogrammierung unter Android.

 [Teilen](#)

 [Kommentieren](#)



“Complex Conjugate” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/849/>

Fo Rensiker – sofort! von Stefan Wichmann

„Wir brauchen Sie! Hier und sofort!“

Obwohl es in meinem Beruf gängige Praxis war, dass Kunden meine Dienste immer unverzüglich benötigten, irritierte mich der Ton. Nein, er brachte mich in Rage. Ich konnte befehlende und derart resolute Äußerungen genauso wenig leiden, wie den immer häufigeren Einsatz von Robotern mit menschlichen Zügen. „Sofort“, wiederholte ich trotzig.

Eigentlich wollte ich nicht bei einem Arbeitgeber tätig sein, der ungehörig seine Macht demonstriert, offensichtlich Mitarbeiter wie Leibeigene behandelt und dann womöglich ausspeit, wenn diese nicht einen erhofften Umsatz erwirtschaften. Genau aus diesem Grund gab ich vor gut einem halben Jahr meinen Job in einem Systemhaus auf, besuchte Seminare zum Thema kriminaltechnische Untersuchungen/Forensik und versuchte mein Glück in der Selbstständigkeit. Ich schwankte zwischen meinem entfachten Trotz und dem benötigten Geld.

„Was ist Ihnen dieses ‚sofort‘ wert?“, fragte ich entsprechend aufgebracht. Bereits in dem Moment, in dem ich es aussprach, ärgerte ich mich. Schließlich wollte ich einen Kundenstamm aufbauen. Ärger hin oder her!

Mein zweites Standbein der PC-Nothilfe bei Virenbefall mit Linux-Mitteln griff nicht. „Linux ver-

schreckt die Kunden“, hatte mir jemand gesteckt, „weil damit kein Windows-Normalo etwas anfangen kann.“ Außerdem hatte ich festgestellt, dass sich fast jeder, der einen Virus einfiel, einen Bekannten rief. Kam der nicht weiter, holte er sich Informationen im Internet.

All dies waren keine guten Voraussetzungen für mich als frischgebackenen Forensiker. Als Ziel hatte ich mir gesetzt, für die Polizei oder für die Staatsanwaltschaft tätig zu sein. Doch um dort Aufträge zu ergattern, musste man Erfolge und einen untadeligen Ruf nachweisen. An Firmen kam ich ohne Beziehungen auch schlecht heran, schon gar nicht, da ich so unbekannt war wie ein Krümel in einem Sandkuchen.

„Es wird Ihr Schaden nicht sein!“, bellte er und legte auf.

Mit klopfendem Herzen zerrte ich am Mikrofon meines Headsets herum. Er nannte weder Namen noch Adresse! Zum Glück hatte ich kürzlich meine FRITZ!Box gepatcht, um VoIP-Gespräche automatisch in der Anrufbeantworterfunktion mitzuschneiden. Ich hörte den Anruf erneut ab. Nebenbei öffnete ich das Log meiner FRITZ!Box und suchte nach der zuletzt notierten Telefonnummer. Gegen meine natürliche Neugier kam ich nicht an. Sobald diese geweckt war, gab es kein Halten mehr für mich. Über den Dienst Klicktel führte ich eine Rückwärtssuche zur Telefonnummer durch

und erfuhr Namen und Adresse meines Auftraggebers.

„Ein Headhunter“, jubelte ich. Eilig wählte ich die Nummer. Wenn ich schnell zurückrief und ihn auf sein Versäumnis mir keine Adresse genannt zu haben aufmerksam machte, konnte ich vielleicht punkten. Meine Güte, ein Headhunter, ein Jobvermittler! Der hatte natürlich Verbindungen!

„KI einschalten“, rief ich in das Mikro meines Headsets. Diese wunderbare Verquickung von künstlicher Intelligenz und Hardware konnte mir helfen, schneller an Informationen zu kommen.

„Headhuntingbüro Peters und Partner“, meldete er sich. Mein Headset-Mikrofon reagierte prompt auf die Nachricht, sendete die Audionachricht über das Spracherkennungsprogramm Palaver an meinen KDE-Browser. Der wiederum übernahm mittels Sprachsteuerung die Adresse im Suchfenster. Dies ermöglichte mir, mich vollständig auf meinen Gesprächspartner zu konzentrieren und gleichzeitig die Webseite zu studieren. Headhunting PP stand dort. PP wie Pinkelpause? Statt zu lächeln, hätte ich mich darüber wundern oder besser noch gleich auflegen sollen. Das konnte ja nicht seriös sein. Er war selbst am Telefon, also leitete er nur ein kleines Büro. Auch dies als Warnung zu nehmen wäre hilfreich gewesen. Doch bevor ein Hauch von Vorsicht oder Enttäuschung meine Euphorie dämpfen konnte, sprach er schon weiter.

„Herr Rensiker, sie haben eine Minute und 13 Sekunden gebraucht. Das ist gar nicht schlecht! Wenn Sie jetzt nicht auflegen, sind sie automatisch einverstanden, dass das Gespräch aufgezeichnet wird.“

Mir blieb meine eiligst überlegte Ansprache im Halse stecken. Für einen winzigen Augenblick fühlte ich mich wie bei einem dieser Werbetelefonanrufe, doch mir blieb keine Zeit zum Nachdenken.

„Verzeihen sie“, hörte ich seine Stimme. „Normalerweise gehen wir die Auswahl möglicher Arbeitnehmer anders an, aber unser Kunde legte das Vorgehen so fest, um einen Arbeitnehmer zu ermitteln, dem das Wasser bis zum Hals steht.“

Ein Klos bildete sich in meinem Hals. Nur mühsam gelang es mir, ihn wegzuschlucken. Ungeschickt, dachte ich noch, aber die Erklärung fand ich bald. Herr Peters sprach weiter, ohne eine Reaktion von mir abzuwarten.

„Der Kunde möchte unerkannt bleiben und den ersten Forensiker einstellen, der es nötig hat und in der Lage ist innerhalb von zwei Minuten zurückzurufen. Also, herzlichen Glückwunsch, sie führen mit Abstand.“

Seine Stimme kam mir etwas gelangweilt vor, trotzdem umkrampfte ich die Kopfhörer meines Headsets, als hätte ich soeben den Jackpot geknackt.

„Wenn sie einen Kurzjob mit absoluter Verschwiegenheitsklausel und doppeltem Verdienst annehmen möchten, dann sagen sie jetzt ja.“

Ein Telefonroboter, schoss mir durch den Kopf. Die kleinen Geräte hatten längst die althergebrachten Anrufbeantworter abgelöst und waren für die tollsten Dinge zu gebrauchen. Manchmal setzen sie Aufgaben etwas tölpelhaft um, doch im großen und ganzen. . .

„drei, zwei“, zählte er herunter.

„Ja“, schrie ich.

„Melden Sie sich unter der Adresse 52 57 766 00 und 13 30 072 00.“

Die Verbindung brach ab und ich fühlte mich total überrannt. ‚Komische IP-Adresse‘, dachte ich noch. Über den Dienst whois versuchte ich etwas über meinen Auftraggeber zu erfahren. Nichts. Ich rief Suchmaschinen auf, suchte in aktuellen sozialen Netzwerken wie neuroLingua und in alten wie Facebook, Google+ und Second Life. Nichts. Mein Portscanner NMap meldete auch keinen offenen Port auf der IP-Adresse und Zugänge über rLogin, ssh oder die Direkteinwahl per VoIP führten ins Nirwana. Ich verträdelte meine Zeit mit einem geheimnisvollen Arbeitgeber und hatte nicht einmal einen Arbeitsvertrag in der Hand! Entmutigt ließ ich meine Hände auf die Tastatur sinken. ‚Vielleicht war die Adresse ja noch gar nicht freigeschaltet?‘

Über den Ping-Befehl in der Konsole erhielt ich eine Bestätigung, dass die Adresse weder verfügbar noch erreichbar war. Enttäuscht ging ich zum Kühlschrank, nahm mir ein Bier und trank es im Stehen halbleer. Irgendjemand hatte mir einen blöden Streich gespielt. Nur wer? Ein Aprilscherz im Oktober? In Gedanken ging ich alle meine Bekannten durch.

„Moment mall!“ Ich stellte das Bier auf dem Tisch ab. Erneut rief ich die Telefonnummer an. Vielleicht hatte ich mich ja verhört oder die Adresse falsch mitgeschrieben. „Die Nummer ist nicht vergeben“, tönte es aus dem Hörer. Erneut rief ich die Rückwärtssuche zu der Telefonnummer auf. Der Eintrag stand noch in der Datenbank. Über Google Maps schaute ich mir ein Satellitenfoto der Adresse an. Nichts, das heißt, an der Adresse lag ein Park. Zögernd tippte ich die mir genannte Adresse in einen Internetdienst zur GPS-Lokalisierung ein.

„Bingo“, flüstere ich, nahm meine Jacke vom Garderobenhaken und machte mich auf den Weg. Die Adressangabe 52 57 766 00 und 13 30 072 00 stimmten von den Koordinaten her mit der Ortsangabe zur Telefonnummer überein!

Mein Sunstepper stand wie immer direkt neben der Tür. Tagsüber speiste die Solarzellenoberfläche des Rollers die Elektronik und den Motor. Da bereits die Nacht hereinbrach, übernahm automatisch der Akku die Stromversorgung. Der Einfachheit halber tippte ich die GPS Koordinaten

ein. Während der Fahrt grübelte ich, warum ich weitere Zeit mit diesem „Auftrag“ vergeudete. Ein Forensiker sichert Spuren. Er hackt sich nicht in Systeme ein, sondern erkennt ein unsicheres System und bereitet getätigte Angriffsversuche für die Staatsanwaltschaft auf. Er ist auch kein Detektiv, also was sollte das hier alles?

Ein Summton riss mich aus meinen Gedanken. Ich erreichte mein Ziel und stand direkt vor einem Baum. Bewaffnet mit meinem linuxbasierten iCom-Phone umrundete ich diesen, dann rief ich die Taschenlampen-App auf und leuchtete den Stamm ab. An einer Stelle ragte ein USB-Anschluss heraus. Ohne zu zögern, steckte ich meinen iCom an den Anschluss. Auch dies war ein leichtsinniger Fehler. Zu dieser Zeit vertraute ich noch darauf, dass es für ein Linuxbetriebssystem kaum Viren und Malware gab. Das Display schaltete sich aus. Verwundert tippte ich auf darauf herum. ‚Vielleicht war ja nur der Bildschirm schöner angegangen‘, hoffte ich.

Automatisch hielt ich die Luft an, stöpselte mein Gerät ab und schaltete es aus. Versuchte es auszuschalten. Die grüne LED blinkte. Die LED für Funkverkehr. Hektisch öffnete ich die Rückseite meines iCom.

„Mist!“, schrie ich unbeherrscht. Alte Handys hatten noch eine Mikro-Simkarte, dies hier war ein iCom. Ein lebenslang an eine feste IP-Adresse gebundenes Gerät mit tausend Funktionen, jedoch ohne Ausschalter. In meinem Hirn purzelten

die Werbeslogans durcheinander: Handy, Mail, Social Network, Fernsehen, Notizblock, sämtlicher Schriftverkehr, Scanner, Drucker... Das Ding sendete wahrscheinlich mein ganzes Leben, sprich meine Kontodaten an einen Hacker und womöglich meinen Lebenslauf an einen Personalausweisfälscher! Mein Magen rebellierte. Weder konnte ich die Datenübertragung beenden, noch das Gerät ausschalten.

Wie konnte ich den Funkverkehr am ehesten unterbinden? Die Antenne abbrechen, indes die war im Gerät fest integriert, genauso wie der Akku. Außerdem wollte ich das 2000 Credits teure Gerät nicht zerstören. Meine Augen suchten hektisch das Gelände ab. Hier gab es aber auch nichts, was einen Funkverkehr lahmlegen würde. Ein Faradayscher Käfig konnte das Übertragungssignal abschirmen! Ich rannte von Grab zu Grab und suchte nach Metallkreuzen, doch selbst wenn ich eines fände, wäre es zu fest verankert, um es herauszureißen. Wahrscheinlich hätte ich so ein Kreuz eh nicht um mein Gerät biegen können.

Völlig außer Atem kam ich am Müllcontainer vorbei. Er war aus Metall und ich warf mein Gerät hinein, klappte den Containerdeckel drüber und hoffte so den Funkverkehr zu unterbinden. Doch was nun? Der Container war viel zu groß, um ihn mit nach Hause zu nehmen. Ich beschloss meinen iCom hierzulassen und nach Hause zu fahren. Dort würde ich per Fernwartung kontrollieren, was hier geschehen war. Der Sunstepper brummte so laut er konnte, während ich mit voller

Energie nach Hause düste. Mir blieb fast das Herz stehen, als ich ankam. Meine Wohnungstür stand weit offen. Die Diebe hatten alle PCs inklusive meiner nagelneuen Forensikerausrüstung gestohlen. Selbst das halbvolle Bier hatten sie ausgetrunken. Sie mussten also auch die Position meines iComs ausgelesen haben und räumten meine Bude aus, solange ich nicht da war. Wütend klingelte ich bei meinem Nachbarn Sturm.

„Was soll denn das?“, schnappte dieser aufgebracht.

„Rufen sie die Polizei!“

„Wer sind sie denn überhaupt?“

Argwöhnisch starrten wir uns an. Wie kannten einander nicht. Keiner kannte mehr seine Nachbarn. Wir alle waren weltweit vernetzt und kannten niemanden mehr persönlich. Offensichtlich schossen ihm die gleichen Gedanken durch den Kopf. Er wählte auf seinem Smartphone und reichte mir das Gerät, während er sich gleichzeitig zwischen mich und die Haustür schob. Er war misstrauisch, wie ich selbst im realen Leben und bei Onlineaktivitäten auch. Nur beim Anschließen von Geräten an irgendwelche Stecker ging ich offensichtlich äußerst unvorsichtig vor. Im Geiste trampelte ich vor Wut auf der Stelle.

„Kommen sie zur Direktion 13, Abteilung Schwerverbrechen, sie sind nicht der Erste, der aufgenommen wurde“, antwortete eine Frauenstimme.

„Danke“, rief ich meinem Nachbarn zu, der ohne Regung gewartet hatte. Jetzt galt es, schnell dorthin zu kommen.

Ich drängte mich am Nachbarn vorbei, obwohl der wie angewurzelt im Wege stand und rannte, so schnell ich konnte zum Sunstepper. Gerade wollte ich losbrausen, als seine Faust mich traf.

„So einfach klaust du mir mein Smartphone nicht“, schnarrte er.

Während mein Auge zuschwoll, suchte ich den Weg zur Direktion 13. Dort angekommen lotste mich ein iRobot in die Abteilung für Körperverletzung.

„Ich wurde ausgeraubt“, rief ich. „Man hat mich in die Abteilung Schwerverbrechen bestellt.“

Einer dieser iRobots gluckste verhalten: „Schwerverbrechen, die kleine Beule, hahaha.“

Wenn ich sonst schon keine Roboter mit menschlichen Zügen mochte, heute stand mir der Sinn erst recht nicht danach. Ich suchte krampfhaft Blickkontakt zu dem Blechkameraden und sprach wie mit einem störrischen Kind. „Hör zu!“ Er erstarrte und gehorchte.

Endlich wies er mir den Weg in die Direktion 13, Abteilung Schwerverbrechen. Sie schraubten das Schild gerade ab und ersetzten es gegen ein Neues. „Wir heißen jetzt Rookie Vitro Pish!“, lachte ein

Mitarbeiter verhalten. Er verstummte, als ein stämmiger Mann ihn missbilligend ansah. Wahrscheinlich sein Chef. Ich blieb stirnrunzelnd stehen und überlegte: Rookie, ja, wie ein Anfänger fühlte ich mich. Vitro könnte bedeuten ‚im Glas sitzen‘ und Pish ist der, der wie ein Vogel zwitschert.

„Wollen sie die Opfer auch noch beleidigen?“, fragte ich scharf.

Eine Frau stand vom Schreibtisch auf und zeigte auf ein bereits angebrachtes Schild: „Rootkit, Virus, Trojaner und Phishing, kurz RoKiViTroPhish. Hier beleidigt niemand jemanden. Wir sind innerhalb der Behörde eine externe und private Sonderermittlung mit Spezialbefugnissen.“ Das letzte Wort zog sie in die Länge. „Sie sind Herr Rensiker?“

Ich nickte.

„Kommen sie, es können ja nicht alle Forensiker sein und derlei Betrügereien im Vorfeld erkennen.“

Nervös bohrte ich mit der Zunge in meinem Mund herum.

„Kann ich ihre Daten haben?“

Erneut nickte ich. Bei der Frage nach meinem Beruf murmelte ich: „Forensische Analysen.“

Sie hielt beim Tippen inne. „Ach eine dieser Budden, die versuchen, unbedarften Lamern den Weg

zu weisen?“ Mühsam unterdrückte sie ein Lachen, hob eine Augenbraue und schaute ab sofort durch mich hindurch.

„Lamer?“, fragte ich verstört.

„Na wie dieser Lamer Exterminator Virus, der auf dem seligen AMIGA im letzten Jahrhundert sein Unwesen trieb. Der hat doch nur unvorsichtige Benutzer heimgesucht, und die haben dann den PC einfach neu aufgesetzt, egal wie viel Arbeit und Datenverluste dies verursachte.“ Sie sah kurz in meine Richtung, fixierte jedoch erneut irgendeinen Ort hinter meiner Nasenwurzel. „Lamer halt.“

Ich riss mich zusammen, in der Hoffnung Informationen und Hilfe zu erhalten. Gleichzeitig wusste ich, dass sie nachher ebenso über mich herziehen würde, wie über alle anderen auch.

„Kommen sie mal zu mir herum“, sagte sie fast mütterlich. Ich interpretierte dies als Entschuldigung.

Beide starrten wir auf den Bildschirm. „Es ist mir so peinlich auf den Betrug hereingefallen zu sein“, stammelte ich. Sie ignorierte meine Bemerkung.

Als sie endlich sprach, hörte ich einen besserwisserischen Ton heraus. Es ärgerte mich. „Sie wissen, was eine KI ist?“

Ich nickte. Künstliche Intelligenz, na und. Die gibt es in abgespeckter Form sogar in Kühlschränken.

„Die Betrüger setzen KI-Roboter ein, die all Ihre Daten aus dem Internet zusammentragen. Diese sammeln sie in Big Data Containern. Anhand der Analyseprogramme werden alle passenden Daten miteinander verknüpft.“ Kurz hielt sie inne, um die Wirkung ihres Wissens auf mich zu prüfen. Jetzt erhob sie auch noch ihre Stimme. „Immer, wenn sie über ihre E-Mail-Adresse im Internet Nachrichten versenden“, belehrte sie mich, „unterschreiben sie ihren Text üblicherweise zumindest mit ihrem Vornamen. Irgendwann senden sie irgendjemandem z. B. ihr Geburtsdatum oder ihren Beruf.“

Ich räusperte mich in meinem Ärger. Am liebsten hätte ich sie zur Rede gestellt, warum sie mich so niedermachte mit ihrer Stimme, ihrem arroganten Gehabe, mit ihrer ganzen Art. Offensichtlich merkte sie meinen Unmut. Sie zögerte kurz, trällerte dann jedoch doch etwas keck: „Forensiker, ja? Ich finde ja, das sollte eine geschützte Berufsbezeichnung sein.“

Ich zuckte innerlich zusammen, während sie es genoss, ihr augenscheinlich kleines Selbstbewusstsein auf meine Kosten aufzuwerten. Ein Kollege schaute zu uns herüber und sie fuhr in ihren Ausführungen fort, als hätte sie mich eben nicht gedist. „Dann trägt der KI-Roboter alle Daten im Internet zusammen und analysiert alles, bis er ein Profil des neuen Opfers erstellt hat. Jede neue Information ermöglicht es den Robotern und deren Analyseprogrammen weitere UND-Verknüpfungen und neuerdings auch eine

unbegrenzte Anzahl von ODER-Verknüpfungen in Suchanweisungen zu verbinden.“

Sie sprach immer schneller, tippte einfach meinen Vornamen, Nachnamen und meinen Beruf in der kostenlosen Personensuche pipl.com ein und warf sich, überlegen, wie sie war, in ihrem gepolsterten Bürostuhl zurück. Ich staunte nicht schlecht über die seitenweisen Ausgaben auf dem Bildschirm.

„Und warum haben die Diebe es bei mir so kompliziert gemacht?“, fragte ich. „Sie hätten doch einfach in meine Wohnung einbrechen können.“

Ihre Augen klimperten überheblich. „Dies ist die neue Variante der Diebe. Heute trägt jeder normalerweise ein Smartphone oder Tablet immer bei sich. Hätten Sie ihr Smartphone unbedarft an einen USB-Stick zweifelhaften Ursprungs gesteckt? Nein. Ihre Neugier hat sie getrieben und ihr Streben, endlich einen Auftrag zu ergattern. Die Diebe nutzen mittlerweile alle Kanäle, die sie auswerten können. Ich nenne sie SoTa, Social-Taktiker. Sie überlegen sich eine Taktik, um mit ihren Opfern zu spielen und dann optimal auszulöschen. Neben allen Daten, die sie, Herr Rensiker“, sie trieb es langsam auf die Spitze, aber ich brauchte ihre Informationen, „im Internet selbst preisgeben, hinterlegen sie in ihrem Handy oder wo auch immer weitere Informationshäppchen.“

„Ja und?“, fragte ich matt.

„Ja, was haben Sie denn in ihrem. . .“

„iCom“, half ich ihr aus.

„. . . ihrem „iCom“ gespeichert, was nicht im Internet verfügbar ist?“

Ich dachte kurz nach. Online sicherte ich meine Fotos, Mails, Adressen, meinen Aufenthaltsort, meine Urlaubsplanungen und Angaben über meine Hobbys. Was sich für zusätzliche Daten auf dem „iCom“ befinden sollten, konnte ich mir nicht erklären.

„Ich helfe ihnen. Sie hinterlegen in ihrem Gerät ihre Passwörter für ihre Mailaccounts, SMS, MMS und vielleicht sogar ihre Verbindungsdaten für das Onlinebanking.“

„Sie rauben mich aus“, stöhnte ich, „und das komplett.“

Sie schüttelte den Kopf. „Kompletter als komplett. Sie haben zeitgleich ihre Wohnung durchstöbert und ihr Leben vom Scheitel bis zur Sohle gestohlen.“

Ich runzelte die Stirn.

„Angenommen sie wollen sich jetzt bewerben, egal um was; Wohnung, Arbeitsplatz oder eine Ehe eingehen. Sie haben keine Papiere, keine Verträge, ja nicht einmal einen Ausbildungsnachweis. Beantragen sie mal eine Versicherungsnummer ohne jegliche Papiere.“ Sie sah mich herausfordernd an.

Zögernd schüttelte ich den Kopf: „Quatsch. Wenn jemandem sein Haus abbrennt, sind seine Daten auch weg.“

Sie nickte. „Während wir so nett plaudern“, sagte sie, „wählen sich die Diebe mit den gestohlenen Passwörtern in ihr Leben ein und verkaufen einem Mafiosi dann nicht nur einen nachgemachten Personalausweis, sondern ein komplettes Leben.“ Sie sah mich mitleidig an. „Sie sterben gerade digital. Sie werden gelöscht, ihr Bild ausgetauscht, sie sind tot, mausetot.“

„Dann tun sie doch etwas“, rief ich aufgebracht.

„Ich weiß ja nicht einmal, wer sie wirklich sind“, grinste sie.

„Was lachen sie denn jetzt so blöd?“, schrie ich.

„Wenn sie mir eine Anzeige unterschreiben, wird diese im EDV-System abgelegt und von den KI-Robots der Gauner automatisch gelöscht“, japste sie. „Oh, das ist genial!“

Wütend stürmte ich aus dem Gebäude. So wie sie wollte ich nicht werden. Am Schreibtisch gefesselt und die einzige Freude darin sehen, sich am Leid anderer zu ergötzen. Wahrscheinlich war sie unglücklich mit sich selbst, aber immerhin gab sie mir wertvolle Informationen. Sie hatte mich angestachelt. Allein, um es ihr zu zeigen, würde ich eine Lösung finden, zu der sie nicht in der Lage war.

„Überlege, überlege, überlege“, trieb ich mich an. Meine Gedanken kreisen. „Ich bin Forensiker, das wäre doch gelacht, wenn ich aus dieser Nummer nicht herauskäme!“ Etwas ratlos stand ich vor meinem Sunstepper. Dann kam mir die Idee. Solange eine Datenübertragung stattfindet, kann ich die übertragenen Daten abfangen und sichern. Mit Schwung warf ich mich auf meinen Sunstepper, fuhr zu meiner Freundin und bat um ihren alten Tablet-PC.

„Der Akku ist bestimmt leer“, rief sie. Besorgt strich sie mir eine Haarsträhne aus dem Gesicht und besah mich. „Was ist mit Deinem Auge?“

Ich hob abwehrend die Hand. „Ich erzähle es dir nachher, versprochen. Im Moment arbeite ich an einem wichtigen Auftrag.“ Jeder Mensch lügt am Tag 200 mal, dies war heute meine erste Lüge, jedenfalls soweit ich mich erinnerte.

Sie runzelte die Stirn, als ich mir aus dem Internet einen Netsniffer herunterlud, das Tablet nahm und aus der Wohnung stürmte. Am Friedhof umrundete ich den Müllcontainer.

„Die LED blinkt noch, also sendet der iCom“, murmelte ich vor mich hin. „Wahrscheinlich, weil der Müllcontainer den Datenverkehr stört und die Datenpakete immer wieder gesendet werden müssen, bis eine erfolgreiche Quittung zurückkommt.“ Hastig schloss ich den Tablet PC an den Host-USB-Port des iCom der für Zusatzgeräte ausgelegt war. Es klappte. Mühelos sicherte ich den

Inhalt des Hauptspeichers und das Flashrom, sowie die interne SSD-Card meines iCom auf den Tablet-PC.

Besorgt schaute ich auf das Akkusymbol des Tablets. „Der Sunstepper hat einen Steckdosenanschluss“, rief ich. „Das sind 230 Volt, die ich heruntertransformieren müsste.“

Entnervt vergrub ich meinen Kopf in meinen Händen. Wenn ich das iCom vom Tablet aus roote, kann ich das Tablet vielleicht über den USB-Port mit Strom versorgen. Vielleicht könnte ich auch die Datenübertragung zu den Gaunern abbrechen! Auf der anderen Seite verändere ich damit den Zustand des Hackvorgangs, überlegte ich laut. Damit wären meine Logmeldungen wertlos! Kein Gericht der Welt würde meine Aufzeichnungen mehr anerkennen. Als mein Blick auf den Tacho des Sunsteppers fiel, wusste ich, was zu tun war. Mit Bordwerkzeug des Sunsteppers löste ich die Schrauben der Verkleidung, legte das Display frei, zog den Stecker ab und isolierte die rote und blaue Leitung mit dem Daumennagel. Vorsichtig steckte ich die Leitungen an die beiden äußeren Kontakte des USB-Ports vom Tablet-PC. Er wurde jetzt vom Akku des Sunsteppers versorgt.

Mittels Paketsniffer schrieb ich die Bits, Bytes und Nibbles mit und durchsuchte die Logmeldungen nach der Ziel-IP-Adresse. Der Internetdienst GeolP half mir bei der Standortbestimmung. „Das macht ein Forensiker!“, rief ich und schlug mei-

ne linke Faust in die rechte Handfläche. „Daten sichern!“

Euphorisch ließ ich die Aufzeichnung weiterlaufen, während ich den Baum mit dem USB-Stick suchte. Mühsam löste ich den Dual-USB-Stick heraus und steckte ihn in meine Tasche, bevor ich zum Müllcontainer zurückrannte.

Über 123recht.de berichtete ich von dem Vorgang in der Hoffnung einen Anwalt für meinen Fall begeistern zu können. Ich bot für Beratung und Aufklärung 5000 Credits und das ist nicht viel. Es meldeten sich genau 13 Anwälte von Betrogenen. Alle wiesen auf ähnliche Fälle hin, bei denen sie aufgrund fehlender Beweise nicht weitergekommen waren.

„Eine Sammelklage“, stammelte ich erleichtert. „Wir machen eine Sammelklage. Ich liefere die Daten, die die Täter ans Messer liefern und sie vertreten mich kostenlos!“ Ich schuf eine Win-Win-Situation.

In dem Moment endete die Datenübertragung kurz zwischen meinem iCom und dem Tablet. Mein Paketsniffer meldete jetzt eingehende Daten. Bevor ich analysieren konnte, was an meinen iCom gesendet wurde, erstarb das Display. Eine Rauchwolke stieg auf. Als ich das Gerät berührte, verbrannte ich mir die Finger. Dann explodierte der Akku und zerfetzte das Gerät. Ein Plastiklitter des Gehäuses traf mich am Kinn. Ich lächelte. Auf dem Tablet-PC war der Angriff mitgelogget.

Die Hacker leisteten ganze Arbeit, wenn sie das kompromittierte Gerät zerstörten.

Natürlich sah ich mir die mitgeschriebenen Logmeldungen an, die Daten waren ja alle auf dem Tablet gelogget. Staunend erkannte ich, dass es möglich war, die im akkubetriebenen Gerät eingesetzte Akkuladezustandssoftware zu manipulieren. Sie hatten dem Akku einfach volle Last abverlangt und dem Betriebssystem gleichzeitig signalisiert, dass er zu wenig Leistung bringt. Das Betriebssystem regelte die Akkuleistung automatisch in einen gigantischen Bereich, der zur Überhitzung im Gerät führt. Die Leitungen des Displays schmolzen, bewirkten einen Kurzschluss, die per Gesetz vorgeschriebene Notabschaltung des Akkus wurde aktiviert und die Elektronen in den Akkuzellen konnten mit dem plötzlichen Abschaltvorgang nicht umgehen. Das war, als machte man mit einem Sunstepper aus vollster Fahrt eine Vollbremsung und erwartet, dass der angeschmorte Sicherheitsgurt das verkraftet.

Gleich am nächsten Tag legte ich meine Auswertung der Staatsanwaltschaft vor.

Zwei Tage später durchsuchte die Kripo die Büroräume, die zu der von mir ermittelten IP-Adresse gehörten. Es handelte sich um einen einzigen Täter. „Gefahr in Verzug“, hatte der Richter geurteilt.

Bei der Gerichtsverhandlung beteuerte der Angeklagte, dass er nichts gemacht hätte: „Ich habe keinem Gewalt angetan, der Kläger hat selbst sein

Gerät an den frei zugänglichen USB-Anschluss angesteckt.“

Wütend beugte ich mich zu meinem Anwalt. „Ich werde behaupten, er hätte mir mein blaues Auge verpasst.“

Er grinste. „Das hat ihr Nachbar längst seinen Freunden geschattet. Das steht längst im Internet, damit kommen sie also nicht durch.“

Dann hörte ich das Unfassbare vom gegnerischen Anwalt: „Herr Rensiker hätte die Daten per Sniffer gar nicht mitschreiben dürfen! Das ist Datenschutz!“

Mir platzte der Kragen. Unbeherrscht sprang ich auf: „Doch, die Daten im Internet werden frei übertragen, jeder kann sie empfangen. Man darf nur nichts Illegales damit anstellen! Meine Mitschriften können jedoch helfen, ein Verbrechen aufzuklären!“

Der Richter sah mich über seine randlose Brille hinweg an: „Haben sie einen Beweis, wann sie an welchem Ort waren?“

Ich holte tief Luft, besann mich aber und presste mühsam hervor: „Beweis? Google Maps, WebCams des Friedhofseingangs und die NSA hat bestimmt auch etwas!“

Er grinste kurz, bevor er sich an den Angeklagten wandte.

„Nachdem der Kläger also sein Gerät an ihren Dual-USB-Stick angeschlossen hatte, war der Besitzer dieses Gerätes gewissermaßen ‚ihr Kunde‘?“

Der Angeklagte sah zu mir herüber und stand vorsichtig auf. „Ja.“

„Sie erkennen also Herrn Rensiker als ihren Kunden an?“

Der Angeklagte schien jetzt bei der Nachfrage zutiefst verunsichert und legte mit offenem Mund seinen Kopf schief. Es sah ein wenig dämlich aus. „Äh, ja, als er. . .“, hilflos suchend sah er zu seinem Verteidiger, der den Satz beendete „. . . als Herr Rensiker sein Gerät ansteckte, nahm er freiwillig die Dienste in Anspruch.“

„Haben Sie den Kunden vor der Übertragung der Daten auf die allgemeinen Geschäftsbeziehungen ihres. . . sagen wir mal. . . ihres Dienstes hingewiesen?“

Stille.

„Sie haben doch eine Firma, oder?“ Der Angeklagte nickte zögernd. „Welche Rechtsform?“

„Rechtsform?“ Irritiert schaute er erneut seinen Verteidiger an. Der flüsterte laut: „GmbH, GBR, OHG, Ltd.“

„Privat.“

„Dann haften sie mit ihrem persönlichen Vermögen!“

„Äh“, machte der Angeklagte. Sein Blick hetzte vom Verteidiger zum Richter und wieder zurück.

„In dem einen meinem Mandanten nachgewiesenen Fall bekennt sich der Angeklagte schuldig keine Datenschutzerklärung und AGB mitgeteilt zu haben“, meldete sich der Verteidiger plötzlich aufgeschreckt zu Wort.

„Das zahle ich aus der Portokasse“, wiegelte der Angeklagte leise ab. Offensichtlich war er sich seiner Sache wieder sicher. Sein Körper straffte sich und er stand gerade und breitbeinig vor seinem Stuhl. Mit erhobenem Kopf stand er da, dann sah er sich um und setzte sich. Ich tat es ihm nach.

Der Richter belehrte ihn: „Nein, das gibt ihre Portokasse nicht her. Herr Rensiker ist Forensiker. Er hat alle ihre ‚Kundenkontakte‘ über den Backup-partner ihres Providers in Erfahrung gebracht.“

„Vorratsdatenspeicherung ist doch verboten!“, rief der Verteidiger.

„Aber ihr Mandant hat doch selbst ein Backup seiner Daten bei seinem Provider beantragt.“ Der Richter blätterte in den Unterlagen. „Ah, das ist eine Voreinstellung des Providers. Sie haben eine eigene MAX DB genutzt, die ihr Provider gemäß ihres Vertrages automatisch sicherte. Im Zuge der von ihnen in Auftrag gegebenen Ermittlungstätig-

keit des Herrn Rensiker haben sie ihn zu dem Ort gelotst, an dem er gemäß ihrer Weisung hinkommen sollte. Es war naheliegend, dass er den USB-Port nutzt, der am Zielort installiert war, im Vertrauen an seinen Auftraggeber, also an sie.“ Der Richter schaute auf. „Zwar haben sie den von ihnen in Auftrag gegebenen Job noch nicht bezahlt, jedoch ist er telefonisch nachgewiesen durch die FRITZ!Box des Herrn Rensikers. Und da sie bereits erklärt haben, dass er ihr Kunde war, wird auch der anhängenden Sammelklage stattgegeben. Folgerichtig sind diese nämlich auch ihre Kunden gewesen.“ Er schaute erneut kurz in seine Aufzeichnungen. „Sie haben als Arbeitgeber eine Verpflichtung dem Auftragnehmer gegenüber!“ Zufrieden schaute der Richter den Angeklagten an. „Laut Aktenlage haben wir insgesamt festgestellt, dass sie und ihre KI gesteuerten Roboter jeweils Personendaten im Internet zu ihren Gunsten gelöscht oder geändert haben, um diese meistbietend weiterzuveräußern. Zugegeben, wir haben Glück gehabt, dass sie ihre eigenen Datensicherungen übersehen haben. Durch das nofollow-Tag auf ihrer Webpräsenz griff die Suchlogik ihrer KI-Routinen nicht auf ihren eigenen Seiten, sodass die ursprünglichen Originaldaten nicht manipuliert wurden und uns jetzt helfen, den Fall aufzuklären. Einzig Herrn Rensiker ist die Aufklärung des Falles zu verdanken.“ Er faltete die Hände auf seinem Richtertisch. „Und das Schönste für mich ist das Folgende: Sie müssen zum einen ihre Beauftragung von Herrn Rensiker zahlen und zwar wie vereinbart zum doppelten Satz und bis zum Abschluss der Arbeiten, also bis heute, als auch den

entstandenen Schaden, zuzüglich der Gerichtskosten dieses und aller anhängenden Verfahren. Der entstandene Schaden ihrer bisher ermittelten ‚Kunden‘ wird in einem Folgeverfahren festgestellt.“

Bleich war gar kein Ausdruck für das Gesicht des Verurteilten. Ich hingegen konnte mein Glück kaum fassen.

„Herr Rensiker!“ Die Hand eines stämmigen Mannes schob sich mir entgegen. Ich erkannte ihn als

Mitarbeiter von RoKiViTroPhish. Freudig schüttelte ich ihm die Hand. Sein Jobangebot jedoch lehnte ich ab und ich muss sagen, ich fühlte mich sehr wohl dabei, vor allem, als ich ihm Grüße an seine Kollegin ausrichtete und er den Kopf schüttelte.

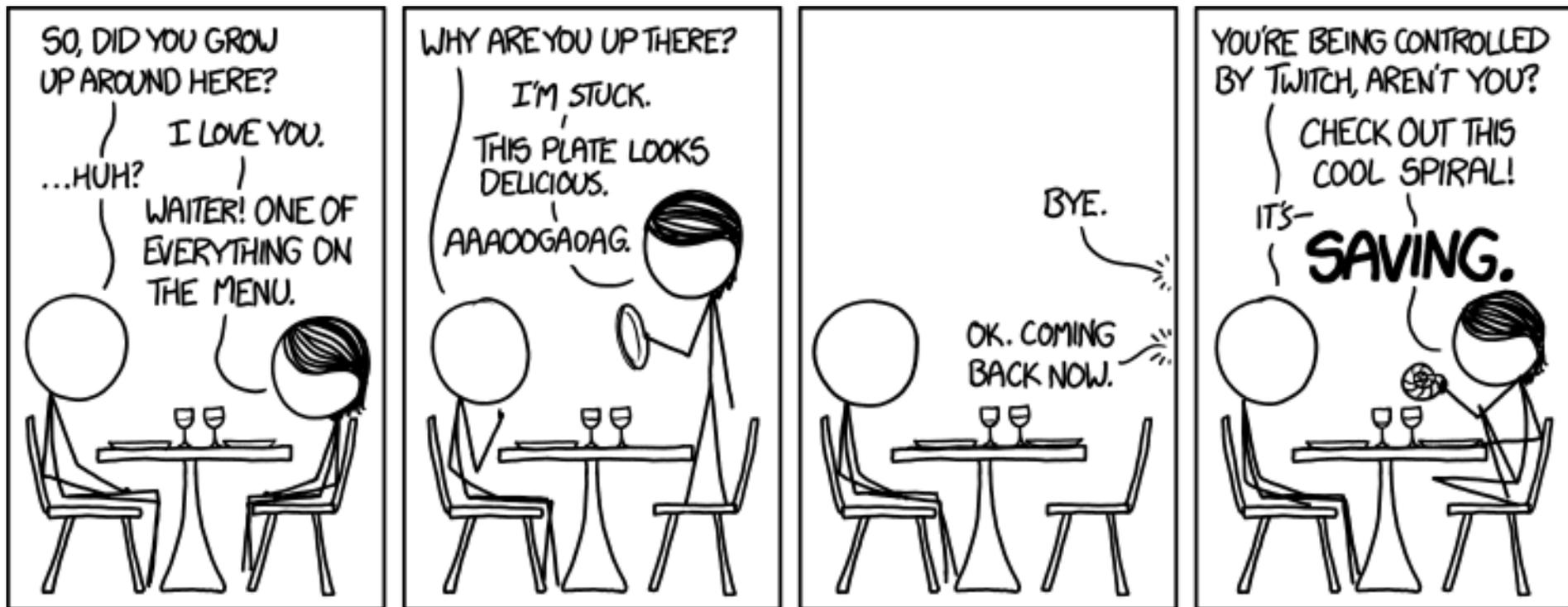
„Haben wir abgeschaltet“, murmelte er kurz. „War das neueste KI-Modell und fühlte sich total überlegen, nur eigene Lösungen hat sie nie gefunden. Aber ihre sprachlichen und mimischen Fähigkeiten waren schon echt genial...“

Autoreninformation 

Stefan Wichmann interessiert sich für das Thema forensische Untersuchungsmethoden im IT-Umfeld und vertiefte seine Kenntnisse an der Fachhochschule der Polizei. Er veröffentlichte EDV-Fachbücher, Lyrik und einen historischen Roman.

 [Teilen](#)

 [Kommentieren](#)



“First Date” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/1333/>

Siebter freiesMagazin-Programmierwettbewerb: Tron

Der letzte freiesMagazin-Programmierwettbewerb ist bereits zwei Jahre her. Eine Abstimmung im September 2013 [1] zeigte, dass die Leser und potentiellen Wettbewerbsteilnehmer das meiste Interesse daran hatten, den Film „Tron“ [2] bzw. das Lightcycle-Rennen darin umzusetzen. 2013 ergab es sich leider nicht, aber dieses Jahr ist es wieder so weit: Der siebte freiesMagazin-Programmierwettbewerb kann beginnen.

Aufgabe

Die Aufgabe ist sehr simpel: In einer Arena treten zwei Bots gegeneinander an. Jede Runde können sie sich entscheiden, nichts zu tun oder sich um 90 Grad nach rechts oder nach links zu drehen. Danach bewegen sie sich automatisch ein Feld vorwärts. Hierbei besetzt jeder Bot das Feld, das er gerade verlässt, welches dadurch für jeden unpassierbar bleibt (auch für den Bot selbst). Wer bei einer Bewegung gegen eine Wand bzw. gegen ein besetztes Feld läuft/fährt, scheidet aus.

Die Idee des Spiels ist nicht neu und leitet sich von dem Film „Tron“ ab, der 1980 aufgrund seiner technischen Raffinesse für Aufsehen gesorgt hatte. In dem Film treten Computerprogramme in Lightcycle-Rennen gegeneinander an und versuchen sich gegenseitig den Weg abzuschneiden. Zu diesem Prinzip gab es auch schon einige Implementierungen. Als Solo-Variante mit leichter

Variation ist vielen das Spiel unter dem Namen „Snake“ [3] bekannt.

Kommunikation

Wie in den letzten Programmierwettbewerben soll die Teilnahme möglichst vielen Personen ermöglicht werden. Aus dem Grund wird auf ein Client-Server-Konzept gesetzt, über das das Spiel läuft. Die Teilnehmer müssen nur einen Bot programmieren, der die Server-Befehle von der Standardeingabe (STDIN) liest und eine Antwort auf der Standardausgabe (STDOUT) ausgibt. Durch einen Pipe-Mechanismus (um den sich der Client aber nicht kümmern muss) kann der Server so direkt mit jedem Client kommunizieren.

Der Kommunikationsablauf eines Spiels ist dabei wie folgt (ein > ist dabei ein Kommando vom Server an den Client, ein < die Antwort vom Client an den Server; beides dient nur der Illustration und ist nicht Teil der echten Kommunikation):

```
> GAMEBOARDSTART 10,12
> .....#.....
> [...]
> GAMEBOARDEND
> SET 1
> POS 1 2,5 EAST
> POS 2 6,5 WEST
> ROUND 1
< LEFT
> POS 1 2,4 NORTH
> POS 2 5,5 WEST
```

```
> ROUND 2
< AHEAD
> [...]
> OUT 2
> END
```

Jeder Befehl vom Server bzw. an den Server ist durch ein Zeilenende (`\n`) abgeschlossen.

Spielbrettübertragung

Am Anfang wird die Arena (das Spielbrett) vom Server an alle Clients (Bots) übertragen:

- **GAMEBOARDSTART *n,m*** – Dies startet die Spielbrettübertragung mit Breite *n* und Höhe *m*. Das heißt, es werden *m* Zeilen mit jeweils *n* Zeichen übertragen.
- **.....#.....** – Eine Zeile des Spielbretts. Leere Felder sind durch einen Punkt dargestellt, besetzte Felder durch eine Raute #.
- **GAMEBOARDEND** – Ende der Spielbrettübertragung.

Setzen der Teilnehmernummer

Vor Beginn des eigentlichen Spiels wird jedem Bot die eigene Teilnehmernummer übergeben:

- **SET *p*** – Teilt einem Client mit, dass er die Nummer *p* hat, wobei *p* bei 1 anfängt.

Im Wettbewerb selbst wird es nur die Nummer 1 und 2 geben, die Bots sollten aber so program-

miert sein, dass sie auch mit mehr Teilnehmern umgehen können.

Wiederholter Rundenablauf

In jeder Runde wird zuerst allen noch nicht ausgeschiedenen Bots die Position aller anderen noch nicht ausgeschiedenen Bots mitgeteilt:

- **POS p n,m DIR** – Legt die Position eines Bots auf dem Spielbrett fest. **p** ist dabei die Botnummer, die bei 1 beginnt. **n** und **m** ist die Position auf dem Spielbrett in horizontaler bzw. vertikaler Richtung, die ebenfalls bei 1 beginnt. **DIR** gibt die Richtung an, in die der Bot gerade schaut und kann die Himmelsrichtungen **NORTH**, **EAST**, **SOUTH** und **WEST** annehmen.

Noch einmal der Hinweis: Die Position ausgeschiedener Bots wird nicht übertragen! Zusätzlich enthält **p** natürlich auch die eigene Spielnummer. Dies ist für die Startposition wichtig. Zusätzlich kann ein Client damit die eigene Position und Blickrichtung abgleichen.

Danach erfolgt der Befehl an den Client, eine Entscheidung zu treffen, was der Bot tun soll:

- **ROUND r** – Startet die Runde **r**. Der Wert **r** muss nicht ausgewertet werden.

Der Server erwartet nun vom Client eine Antwort. Ausgeschiedene Bots bekommen diesen Befehl natürlich nicht mehr zugesendet und müssen daher auch nicht antworten.

Client-Bewegung des Bots

Der Client hat drei Befehle zur Auswahl, wie er auf ein **ROUND** antworten kann:

- **RIGHT** – Dreht den Bot um 90 Grad nach rechts.
- **LEFT** – Dreht den Bot um 90 Grad nach links.
- **AHEAD** – Dreht den Bot nicht.

Nach der Übertragung des Befehls dreht der Server den Bot entsprechend und bewegt ihn ein Feld vorwärts.

Achtung: Es wird **erst** gedreht und **danach** die Bewegung ausgeführt!

Ausscheiden eines Bots

Fährt ein Bot gegen eine Wand bzw. ein besetztes Feld, scheidet er aus. Dieses Ausscheiden wird allen noch nicht ausgeschiedenen Bots (mit Ausnahme des gerade ausgeschiedenen) mitgeteilt:

- **OUT p** – Ein Bot ist ausgeschieden. **p** ist dabei die Botnummer und startet bei 1.

Nach dem Ausscheiden eines Bots werden an ihn keine Botpositionen und auch kein Rundenstart mehr gesendet. Ein Client kann sich nach dem Ausscheiden des eigenen Bots entweder direkt abmelden oder er wartet noch auf den Spielende-Befehl (siehe unten).

Spielende

Wenn ein Bot ausgeschieden ist, können sich alle noch nicht ausgeschiedenen Bots weiter auf

dem Spielfeld bewegen. Das heißt, das Spiel ist **nicht** vorbei, wenn es nur noch einen nicht ausgeschiedenen Bot gibt. Der letzte Bot darf sich auch weiterhin solange auf dem Feld ungestört bewegen, wie es möglich ist (siehe Auswertung unten).

Sobald auch der letzte Bot ausgeschieden ist, überträgt der Server den Ende-Befehl an alle Clients (auch an die zuvor ausgeschiedenen):

- **END** – Spielende.

Jetzt sollten die Clients sich beenden. Der Server bricht die Verbindung hiernach ab und das Spiel endet.

Implementierung

Der Server ist in C++ geschrieben, sodass man zur Übersetzung einen C++-Compiler benötigt. Hat man diesen installiert, kann man den Servercode herunterladen und entpacken

```
$ wget ftp://ftp.freiesmagazin.de/2014/↵
freiesmagazin-2014-10-contest.tar.gz
$ tar -xzf freiesmagazin-2014-10-contest↵
.tar.gz
$ cd freiesmagazin-2014-10-contest
```

Danach muss man noch Code kompilieren:

```
$ cd src
$ make # "make DEBUG=Y" für die ↵
Debug-Version
```

```
$ cd ..
```

Jetzt kann man den eigenen Bot testen und beispielsweise gegen den Dummybot (siehe unten) antreten lassen:

```
$ ./start.sh "fields/simple.txt" "bots/↵
eigenerbot/bot" "bots/dummybot/bot" --↵
```

bots/eigenerbot/bot ist dabei der Pfad zum eigenen Bot und kann ein beliebiges ausführbares Programm oder Skript sein, welches STDIN auslesen bzw. auf STDOUT schreiben kann.

Dummybot

Für alle Teilnehmer, die C++ beherrschen und sich mehr auf die Strategie als das Drumherum konzentrieren wollen, gibt es im Verzeichnis **bots/dummybot** eine Dummy-Implementierung in C++. Diesen Dummybot kann man als Vorlage (Dabei bitte die Lizenz beachten!) und/oder als Sparring-Partner für den einen eigenen Bot nutzen.

Den Dummybot kompiliert man über:

```
$ cd bots/dummybot
$ make
$ cd ../..
```

GUI zur Analyse

Zur Analyse des Botverhaltens kann man die gespielte Sitzung als Logfile ausschreiben lassen und speichern:

```
$ ./start.sh "fields/simple.txt" "bots/↵
eigenerbot/bot" "bots/dummybot/bot" --↵
log > bot.log 2>&1
```

Das Logfile kann man danach mit der GUI einlesen und den Ablauf sichtbar machen.

Zum Kompilieren der GUI wird ein C++-Compiler und die Qt4-Entwicklungsbibliotheken benötigt:

- **g++**
- **libqt4-dev**
- **qt4-qmake**

Danach kann man die GUI übersetzen:

```
$ qmake && make
```

Die GUI startet mit

```
$ ./gui
```

Eigene Spielfelder

Man kann eigene Spielfelder erstellen und für den Wettbewerb einreichen. Die Syntax ist dabei identisch zu den übertragenen Spielbrettdaten (siehe oben) inkl. **GAMEBOARDSTART** und **GAMEBOARDEND**. Zusätzlich kann man noch die Startposition der Spieler am Ende der Datei festlegen. Die Syntax ist dabei identisch zur Positionsübertragung jede Runde.

Ob diese Spielfelder genutzt werden, obliegt der **freiesMagazin**-Redaktion.

Auswertung

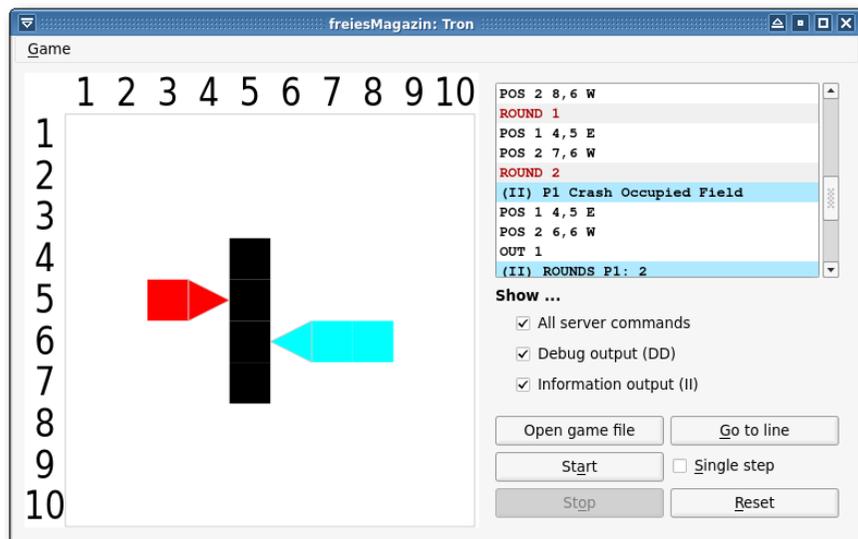
Im Wettbewerb treten jeweils immer zwei Bots gegeneinander an. Es wird dabei immer zwei Spiele geben, damit jeder Bot einmal die andere Startposition hat. Der Bot, der am längsten überlebt, erhält 3 Punkte. Sollten beide Bots gleichzeitig ausscheiden, gibt dies für beide 1 Punkt. Der Bot, der zuerst ausscheidet, erhält 0 Punkte (Dreipunkteregel [4]). Dies wird für alle vorhandenen Spielfelder wiederholt.

Nachdem jeder Bot gegen alle anderen auf allen Spielfeldern angetreten ist, werden die Punkte summiert und der Bot mit den meisten Punkten gewinnt den Wettbewerb. Ein Bot tritt nicht gegen sich selbst an.

An der Stelle noch einmal der Hinweis, dass man seinen Bot dennoch so programmieren sollte, dass er auch mit mehr als einem Gegner zurechtkommt. Dies ist vor allem für Tests (und die Leser später) interessant.

Zusätzlich werden in jedem Spiel die Anzahl der Runden gezählt, bis ein Bot ausscheidet. Diese Anzahl wird dann als Tie-Breaker benutzt, sollten zwei Bots am Ende des Wettbewerbs die gleiche Punktezahl haben.

Hinweis: Sollte es Bots mit Zufallselementen geben, werden mehrere Einzelspiele durchgeführt und der mathematisch gerundete Mittelwert der einzelnen Ergebnisse genutzt. Dies bitte bei der Einreichung mit angeben!



GUI zur Anzeige der Botbewegung. 🔍

Preise

Damit es neben der eigentlichen kreativen Anregung noch einen weiteren Anreiz gibt, am Wettbewerb teilzunehmen, wird es auch Preise geben:

- Der Gewinner des Wettbewerbs erhält ein Buch eigener Wahl im Wert von maximal 35 Euro.
- Der Zweitplatzierte erhält ein Buch eigener Wahl im Wert von maximal 20 Euro .
- Der Drittplatzierte erhält den Film „Tron“ (von 1986) auf DVD, auf dem die Idee des Wettbewerbs basiert.

Zusätzlich gibt es für alle anderen Teilnehmer den Doppel-CD-Sampler des Free!Music!Contest 2014 [5] als Trostpreis, falls gewünscht und solange der Vorrat reicht.

Teilnahmebedingungen

Die Teilnahme am Wettbewerb ist für jeden offen. Ausgeschlossen sind nur die Mitarbeiter der freiesMagazin-Redaktion. Es ist auch erlaubt im Team zu arbeiten, man muss sich dann die Preise aber teilen. Als Programmiersprache kann jede benutzt werden, die die Standardeingabe auslesen bzw. auf die Standardausgabe schreiben kann.

Folgende Bedingungen gibt es:

- Der Bot muss im Quellcode mit einer Kompilierungsanleitung (falls notwendig) vorliegen. Vorkompilierte Binärdateien werden nicht akzeptiert!
- Der Quellcode muss unter einer freien Lizenz [6] vorliegen. Wir helfen gerne bei der Auswahl einer passenden Lizenz, wenn es hierzu Fragen gibt.
- Jeder Teilnehmer darf nur mit einem Bot teilnehmen, darf diesen aber gerne mehrfach verbessern und einsenden.
- Durch die Einsendung des Quellcodes stimmt jeder Teilnehmer der Veröffentlichung des Codes auf der freiesMagazin-Webseite am Ende des Wettbewerbes zu.
- Die Laufzeit der Bot-Programme (inkl. Server-Ausführung) sollte so gering wie möglich sein. Für die Teilnahme sollte pro Runde maximal 1 Sekunde benötigt werden, da der Wettbewerb sonst zu lange dauert.

Der Wettbewerb läuft vom **1. Oktober 2014** bis zum **20. Dezember 2014**. Die Auswertung findet über die Weihnachtsfeiertage statt. Der Gewinner wird dann am **1. Januar 2015** bekannt gegeben.

Bis zum 20. Dezember 2014 können die Teilnehmer ihren Bot an redaktion@freiesMagazin.de senden. Jeder Teilnehmer darf auch gerne eigene Spielfelder zusenden, die dann ggf. für den Wettbewerb benutzt werden.

Hinweis: Die vergangenen Wettbewerbe haben gezeigt, dass es sinnvoll ist, seinen Bot frühzeitig einzureichen, damit dieser getestet werden kann. Nicht lauffähige Bots können nicht am Wettbewerb teilnehmen!

Download: [Quellcode zur Teilnahme](#)

Wir wünschen allen Teilnehmern viel Erfolg und freuen uns auf zahlreiche Zusendungen.

LINKS

- [1] <http://www.freiesmagazin.de/20130909-ergebnisse-der-abstimmung-fuer-programmierwettbewerb>
- [2] [https://de.wikipedia.org/wiki/Tron_\(Film\)](https://de.wikipedia.org/wiki/Tron_(Film))
- [3] <http://de.wikipedia.org/wiki/Snake>
- [4] <https://de.wikipedia.org/wiki/Punkteregel>
- [5] <http://musik.klarmachen-zum-aendern.de/free-music-contest-2014>
- [6] <http://opensource.org/licenses>



Teilen



Kommentieren

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

Administration von Debian & Co – Teil II

Der Artikel „Administration von Debian & Co im Textmodus“ ist sehr informativ. Dort finde ich die wichtigsten Kommandos für die Administration meiner Linux-Rechner. Schreibt bitte noch mehr Artikel auf diesem Niveau. **Franz-Josef Lücke**

Hier drei Anmerkungen zum Artikel:

1. Die Loopback-Adresse ist beim Ping-Befehl versehentlich als `127.0.1.1` geschrieben.
2. Der Befehl `dpkg --get-selections ...` lässt sich deutlich vereinfachen:

```
$ dpkg --get-selections | awk -v ORS=' ' '$2=="install"{print $1}'
```

3. Für die Paketmanagementbefehle, die rein lesend sind, sind keine root-Rechte nötig.

meillo (Kommentar)

Es wäre sehr schön gewesen den Linux-Neulingen gleich `ip` statt `ifconfig` beizubringen. `txt.file` (Kommentar)

Mathematik mit Open Source

Danke für den Artikel. [Ich] wollte nur drauf hinweisen, dass ein sehr mächtiges CAS vergessen [wurde]: Sage [1].

Andreas Ruscheinski (Kommentar)

Browser: alle doof.

Ich bin dann wohl auch behindert, wenn mich dunkler Kontrast und größere Schrift interessiert. Naja, [ich] denke, dass [das] auch viele gesunde Leute interessiert. Vielleicht, weil man beim langen Davorsitzen am PCs einfach eine dunkleren Hintergrund besser findet oder, wenn man den Monitor dank Monitorarm und vielleicht zu breitem Schreibtisch ein Stück weit weg hat, und dann sonst normal immer vorgelehnt sitzt, und mit 130% ganz entspannt davor sitzt.

Zu den Browsern hätte ich noch einen Vorschlag: Conkeror [2] – nicht zu verwechseln mit Konqueror, dem KDE-Browser. Wie einfach es zu installieren ist, variiert. Für Debian gibt es z. B. ein Paket bzw. es ist in den Standard-Debian-Repositories [enthalten].

Das größere Problem könnte dann aber die Konfiguration sein. Conkeror ist quasi der Emacs unter den Browsern, das hat Vor- aber auch Nachteile.

Man kann nicht per Maus in ein Settingsmenü und dort was umstellen. Man muss in eine Config-Datei mit einem Textbrowser reingehen.

Standardmäßig das Zoomlevel einstellen, ist kein Problem. Adblock auch nicht, da ist auch eine Erweiterung oder Funktion, die „darken_page“ heisst, und eben [...] die Seite dunkel macht. Dieses kann man auch mit einer Zeile Code direkt beim Erzeugen eines neuen Tabs machen.

Für Leute mit RSI-Fingerschmerzen oder sowas wäre das sicher eine gute Alternative, weil man einfach zu drückende Tastenkombinationen definieren kann. So kann man z.B. auf `W` „Tab schließen“ legen [...]. Bei Firefox muss man dafür mit dem kleinen Finger tendenziell `Strg`-Taste gedrückt halten, während man mit einem anderen Finger `W` drueckt. Erstens ist der kleine Finger am schwächsten. Zweitens sind solche gedrückt gehaltenen Keycodes sehr schlecht für die Finger. Und drittens vergrößert das starke Spreizen während des Tastenhaltens das Problem noch.

Im weitesten Sinne geht es ja bei der Ergonomie in dem Artikel primär um die Ausgabe, aber die Eingabe sollte man auch nicht vergessen.

Conkeror ist halt ein „hackbarer“ Browser – oder skriptbarer, das trifft's vielleicht noch genauer. Man kann ihn also seinen Bedürfnissen ziemlich weitgehend anpassen. Man könnte also sicher

seitenspezifisch dieses „darken _page“ aufrufen oder für alle Seiten. Es gibt auch Readability-Funktionen, die man aufrufen kann, die „clutter“ von den Seiten entfernen: z. B. „readability_arc90“.

Aber ja, die Einstiegshürden sind ein wenig da, aber es könnte sich lohnen. Zwei bis drei Stunden Zeit investieren, vielleicht auch ein wenig mehr oder weniger vs. jahrelang täglich für die Finger schlechte Tastenkombinationen drücken und sich langfristig vielleicht Gicht oder irgendwas zu holen.

Webjumps ist auch noch so ein nettes Feature und in der Adressliste direkt auch die Bookmarks als Autocomplete vorgeschlagen zu bekommen. External Editing mit irgendeinem gethemeten

Browser der Wahl oder auch Hinting, also dass man per Tastatur die Links oder Felderknöpfe auswählen oder klicken kann, sind auch alles Sachen, die teilweise bei bestimmten Behinderungen, aber auch für normale Menschen nützlich sein kann.

Was allerdings ein wenig problematisch ist, ist, dass der Browser nicht der beste ist, wenn man ihn mit der Maus bedienen will. Es geht wohl mit der aktivierbaren Menüleiste, aber das ist nicht gerade sehr komfortabel.

spdierbit (Kommentar)

Themenwünsche

☛ Möchtet Ihr nicht ein Review zum frisch erschienen OpenMediaVault [3] schreiben? Aktuell gibt es kein besseres Linux-basiertes NAS.

Marc Brinkmann

☛ Danke für den Vorschlag, das klingt in der Tat nach einem spannenden Artikel für *freiesMagazin*. Wir haben den Artikel in die Artikelwunschliste der Leser mit aufgenommen [4], in der Hoffnung, dass sich ein Autor dafür findet.

Christian Schnell

LINKS

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

[2] <http://conkeror.org/> 

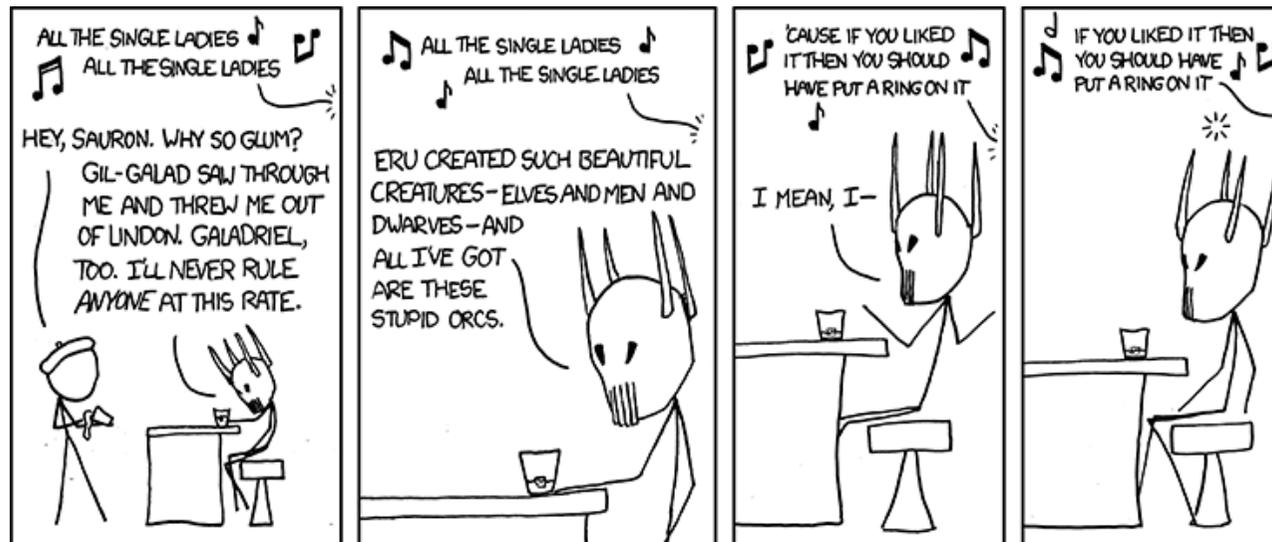
[3] <http://www.openmediavault.org/> 

[4] <http://www.freiesmagazin.de/artikelwuensche>

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

 [Teilen](#)

 [Kommentieren](#)



“Single Ladies” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/712/>

Veranstaltungskalender

Messen

Veranstaltung	Ort	Datum	Eintritt	Link
VCFB	Berlin	03.10.–05.10.2014	–	http://www.vcfb.de/
8. LIT-OL	Oldenburg	17.10.–18.10.2014	frei	http://www.lit-ol.de
Ubucon	Katlenburg	17.10.–19.10.2014	15 EUR	http://ubucon.de/
BeGeistert Treffen	Düsseldorf	25.10.–26.10.2014	45 EUR	http://www.begeistert.org/
OpenRheinRuhr	Oberhausen	08.11.–09.11.2014	–	http://www.openrheinruhr.de/
LinuxDay	Dornbirn	22.11.2014	frei	http://linuxday.at/
data2day	Karlsruhe	26.11.–28.11.2014	–	http://www.data2day.de/
Chaos Communication Congress	Hamburg	27.12.–30.12.2014	80 EUR	https://events.ccc.de/congress/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.

Vorschau

freiesMagazin erscheint am ersten Sonntag eines Monats. Die November-Ausgabe wird voraussichtlich am 2. November u. a. mit folgenden Themen veröffentlicht:

- Spacewalk – Teil 3: Automatisierung und Kickstart
- Rezension: Git – kurz & gut

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

Hauke Goos-Habermann und Maren Hachmann	S. 5
Valentin Höbel	S. 10
Franz-Josef Lücke	S. 20
Mathias Menzer	S. 3
Jochen Schnelle	S. 27
Sujeevan Vijayakumaran	S. 29
Dominik Wagenführ	S. 25
Stephan Wichmann	S. 31

ISSN 1867-7991

Erscheinungsdatum: 5. Oktober 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	

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.