

## Topthemen dieser Ausgabe

### Grundbegriffe der Virtualisierung

Voll-, Para-, Hardware-Assisted-, Desktop- und Anwendungsvirtualisierung, um nur ein paar Schlagworte zu nennen, die vor allem in Verkaufs- und Beratungsgesprächen mit Buzzwords wie Green-IT oder Virtual Clouds gepaart werden. Doch wer weiß, was sich hinter all diesen Begriffen versteckt? Dieser Artikel soll speziell für Einsteiger etwas Licht ins Dunkel bringen. ([weiterlesen](#))

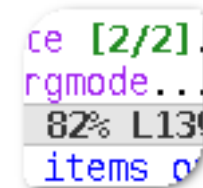
Seite 4



### Ordnung ins Chaos mit Org-Mode

Der Satz „Ich habe keine Zeit . . .“ ist die Standardantwort, die man oft genug zu hören bekommt, wenn man jemanden fragt, ob er eine Aufgabe übernehmen will. Interessanterweise haben alle Menschen genau gleich viel Zeit, nämlich 24 Stunden am Tag, und das Vorhandensein von „Extratagen“ oder „Zeitguthaben bei der Zeitbank“ gehört eher ins Reich der schriftstellerischen Fantasie. ([weiterlesen](#))

Seite 18



### Von der Schaltung zur fertigen Platine mit EAGLE

Mit dem von der Firma CadSoft entwickelten Programm EAGLE kann man Schaltungen und Platinenlayouts entwerfen. EAGLE ist ein Akronym, das „Einfach Anzuwendender Grafischer Layout-Editor“ bedeutet. Trotz dieses deutschen Akronyms ist die Software auf Englisch. Die kostenlose Version ist auf eine halbe Europlatine (100 mm × 80 mm) und zwei Schichten beim Platinenlayout beschränkt, ansonsten kann man fast alles damit machen, was auch mit der Vollversion geht. ([weiterlesen](#))

Seite 30



## Editorial

### Ein Wort ist ein Wort ist ein Wort ...

... und weder Mann noch Frau, möchte man hinzufügen, liebe Leser. Vielen Dank für Ihre zahlreichen Rückmeldungen (siehe „*Leserbriefe*“ auf [Seite 56](#)) auf unsere Frage nach der Verwendung einer geschlechterneutralen Sprache in [freiesMagazin](#).

Rückblende: Aufgrund einer Leseranfrage hatten wir in [freiesMagazin](#) 09/2009 [1] die Frage in den Raum gestellt, ob Sie, liebe Leser – und natürlich auch Leserinnen – die Verwendung einer so genannten „geschlechterneutralen“ Sprache in [freiesMagazin](#) begrüßen würden oder eben nicht. Nun, Ihren Leserbriefen ist deutlich zu entnehmen, dass wir in [freiesMagazin](#) so fortfahren sollen wie bisher, heißt: Wir verzichten auch in Zukunft auf die „geschlechterneutrale“ Sprache und schreiben, wie Sie es von uns gewohnt sind. Vielen Dank an dieser Stelle auch an die Leser, die uns den Artikel „Sprachfeminismus in der Sackgasse“ [2] vorgeschlagen haben.

#### Siezt Du noch oder Duzen Sie schon?

Doch damit nicht genug, denn schon sehen wir uns mit der nächsten Leseranfrage konfrontiert. Nicht, dass Sie uns falsch verstehen, wir lieben Leserfragen, denn Ihre Aufmerksamkeit ist ein wichtiger Grund für unsere Existenz – für die von [freiesMagazin](#), versteht sich. Worum geht es? Ein Leser ist der Meinung, dass es zum guten Ton in Open-Source-Projekten gehöre, wenn

man sich duzt. Unsere interne Regelung sieht anstelle dessen ein eher förmliches Sie vor. Dafür haben wir unsere Gründe, auch wenn diese sicher weder aus Beton noch für die Ewigkeit sind. Dennoch stellen wir uns und Ihnen die Frage: Wie stehen Sie dazu, liebe Leser, bevorzugen Sie in (deutschsprachigen) Open-Source-Projekten ein lapidares Du oder ein förmliches Sie?

#### Mit Ihnen in die Tiefe gehen

Oft liest man nach einer Veröffentlichung von [freiesMagazin](#) in zahlreichen Kommentaren im Internet, dass das Magazin in Zukunft eher Artikel über speziellere Themen veröffentlichen soll, die über Distributionstest und andere, schon häufig beschriebene Themen hinausgehen. Offengestanden: Das ist ein guter Vorschlag – die Umsetzung liegt aber zum großen Teil bei den Autoren. Und dazu laden wir jeden ein! Sollten Sie selten genutzte Programme zu Ihren Lieblingsanwendungen zählen, wie beispielsweise einige Leser des [Ubuntublog.ch](#) [3] von Roman Hanhart, und wollten Sie schon immer der Welt Ihr Wissen und Ihre Begeisterung darüber mitteilen, dann würden wir uns über einen Artikel von Ihnen sehr freuen!

Schreiben Sie uns einfach eine E-Mail an [redaktion@freiesMagazin.de](mailto:redaktion@freiesMagazin.de) und teilen Sie uns Ihren Artikelvorschlag mit.

### Inhalt

#### Linux allgemein

Grundbegriffe der Virtualisierung	S. 4
Gefahren von Fremdquellen	S. 9
Der September im Kernelrückblick	S. 11

#### Anleitungen

Java, Teil 1 – Einführung	S. 13
Ordnung ins Chaos mit Org-Mode	S. 18

#### Software

Kurzipp: Suchen und Finden mit Vim	S. 26
grep – Eine kleine Einführung	S. 28

#### Hardware

Von der Schaltung zur Platine mit EAGLE	S. 30
---	-------

#### Community

41. DANTE-Mitgliedertagung in Esslingen	S. 39
---	-------

#### Magazin

Editorial	S. 2
Erklärungen zum zweiten Programmierwettbewerb	S. 42
Die GUI zum Programmierwettbewerb	S. 51
FAQ zum Programmierwettbewerb	S. 53
Leserbriefe	S. 56
Veranstaltungen	S. 60
Impressum	S. 61

Vielen Dank auch an dieser Stelle an die Autoren, die dem Aufruf in der letzten Ausgabe nach mehr Beteiligung gefolgt sind. Nicht umsonst ist das Magazin diesen Monat wieder so umfangreich, wie man es gewohnt ist.

## Internet im Icon-Format

Die junge Firma embli.net hat sich als Ziel gesetzt, ausgewählte Webseiten auf Ihrem Portal zu kategorisieren und per Icon zu verlinken. Auch **freiesMagazin** zählt zu den Ausgewählten, was uns natürlich sehr freut. Sie finden unsere Verlinkung neben vielen weiteren Open-Source-Anwendungen auf der „Open Source und Freeware“-Seite von embli.net [4]. Da die Icons zufällig platziert werden, muss man die Seite gegebenenfalls neu laden, ehe man das **freiesMagazin**-Logo sieht.

## RSS-Feed für Kommentare

Eine Neuerung auf **freiesMagazin.de** [5] soll auch nicht verschwiegen werden: Ab sofort bieten wir die Möglichkeit an, RSS-Feeds für Kommentare zu abonnieren. Damit wollen wir unseren Lesern einen besseren Überblick über geschriebene Kommentare als auch die zeitnahe Information über neue Kommentare geben, letztendlich soll damit auch der Spaß am Kommentieren steigen. Wie immer ist der RSS-Feed über das Icon in der URL-Zeile Ihres Browsers zu abonnieren.

## Blecherne Roboter und Öltanks!

Die Auflösung unseres Rätsels [6] aus der letzten Ausgabe sollte den meisten **freiesMagazin**-

Lesern (wenn Sie denn die Webseite regelmäßig besuchen oder den RSS-Feed lesen) inzwischen schon klar sein: Gemeint war der Held unseres zweiten Programmierwettbewerbs, der sich einsam durch eine lebensgefährliche Fabrikhalle kämpft.

Kurz zusammengefasst: Die Aufgabe im zweiten **freiesMagazin**-Programmierwettbewerb ist es, mit Hilfe einer selbstgeschriebenen KI einen Roboter durch eine Fabrikhalle zu einer Fahne zu steuern. Die Steuerung wird dabei durch das gezielte Auswählen von Steuerkarten vorgenommen, die den Roboter vorwärts oder rückwärts fahren lassen bzw. ihn auf der Stelle drehen. Damit es etwas spannender wird, gibt es in der Fabrik aber einige Elemente, die den Roboter durch die Gegend schieben oder gar zerstören können. Wenn man diese Möglichkeiten aber geschickt bei der Kartenwahl ausnutzt, kommt man sogar schneller zum Ziel.

Der Wettbewerb wird in vier Kategorien unterteilt, wobei die ersten drei eine bestimmte Schwierigkeitsstufe vorgeben. Die vierte Kategorie richtet sich an alle kreativen Programmierer. Obwohl es schon eine grafische Oberfläche für den Wettbewerb gibt (siehe „*Die GUI zum Programmierwettbewerb*“ auf **Seite 51**), ist es natürlich schöner, wenn es eine 3-D-Fabrikhalle gäbe, durch die sich der Roboter bewegen kann. Hier sind also vor allem die Programmierer gefragt, die mit KI-Programmierung nichts am Hut haben.

Als Ansporn gibt es wie beim letzten Wettbewerb etwas zu gewinnen. Der erste Platz der ersten

drei Kategorien ist mit jeweils einem Buchgutschein in Höhe von 10, 20 und 30 Euro dotiert. Wer in der Kategorie 4 teilnimmt und die (nach unseren objektiven Bewertungskriterien) beste grafische Oberfläche schreibt, erhält einen Buchgutschein über 50 Euro.

Alle ausführlichen Informationen zum Wettbewerb finden Sie im Artikel „*Erklärungen zum zweiten Programmierwettbewerb*“ auf **Seite 42**. Wir hoffen, dass sich auch dieses Mal wieder zahlreiche Programmierer finden, die Spaß an einer solchen Aufgabe haben, und wünschen allen Teilnehmern viel Erfolg bei der Bewältigung dieser Aufgabe!

Die Spiele sind also hiermit eröffnet und wir wünschen Ihnen viel Spaß mit der neuen Ausgabe. Ihre **freiesMagazin**-Redaktion

## LINKS

- [1] <http://www.freiesmagazin.de/20090906-septemberausgabe-ersienen>
- [2] <http://www.bruehlmeier.info/sprachfeminismus.htm>
- [3] <http://ubuntublog.ch/applikationen/exotische-applikationen-der-erste-zwischenbericht>
- [4] <http://www.embli.net/index.php/opensource-freeware.html>
- [5] <http://www.freiesmagazin.de/>
- [6] <http://www.freiesmagazin.de/2009-09-05-ratespiel>

Das Editorial kommentieren 



## Grundbegriffe der Virtualisierung von Florian E.J. Fruth

**V**oll-, Para-, Hardware-Assisted-, Desktop- und Anwendungsvirtualisierung, um nur ein paar Schlagworte zu nennen, die vor allem in Verkaufs- und Beratungsgesprächen mit Buzzwords wie Green IT oder Virtual Clouds gepaart werden. Doch wer weiß schon, was sich hinter all diesen Begriffen versteckt? Dieser Artikel soll speziell für Einsteiger etwas Licht ins Dunkel bringen.

Eine Einteilung in Gruppen bzw. Kategorien fällt nicht immer ganz leicht, da manche Virtualisierungstechnologien mehreren zugeordnet werden können. Deshalb sei darauf verwiesen, dass die folgende Einteilung bei anderen Autoren auch anders ausfallen kann.

### Virtualisierung im Allgemeinen

Es gibt Virtualisierungstechniken, die weiter verbreitet sind als sämtliche kommerziellen Virtualisierungsprodukte von VMware, Citrix und Microsoft zusammen – und gleichzeitig die wohl unbekanntesten. Wer sich nun fragt, ob er diese auch schon verwendet hat, kann beruhigt sein. Denn wer jemals einen Computer benutzt hat, kann diese Frage mit „Ja“ beantworten. Eine prägnante Zusammenfassung der verschiedenen Technologien gibt es leider nur in der englischsprachigen Wikipedia [1]. Dort sind auch die Virtualisierungen aufgeführt, die jeder schon benutzt hat: zum Beispiel virtueller Arbeitsspeicher oder Datenträgervirtualisierung.

Die Virtualisierung des Arbeitsspeichers bezeichnet unter anderem die Zuordnung von virtuellem zu physikalischem Speicher. Dies kann zum einen bedeuten, dass ein Programm denkt, es verfüge über 1 GB zusammenhängenden Speicher, während dieser in Wirklichkeit in Stücken verteilt sein kann. Auf der anderen Seite ist es auch möglich, dass vielleicht die Hälfte davon – ohne Wissen des Anwendungsprogramms – auf der Festplatte ausgelagert wurde. Unter Windows werden diese Daten in der Auslagerungsdatei gespeichert und unter Linux auf der Swap-Partition. Diese Daten können dann auf virtualisierten Festplatten landen. Als einfaches Beispiel sei hier das Zusammenfassen von mehreren Datenträgern zu einem großen erwähnt. Dies kann zum Beispiel durch einen RAID-Verbund in Software und Hardware realisiert werden.

Mit dem Bewusstsein, dass jeder Computer-Benutzer täglich wahrscheinlich mehrere Arten der Virtualisierung verwendet, nun zu den Gebieten, die durch die IT-Branche in den letzten Jahren eine steigende Aufmerksamkeit erhalten haben.

### Betriebssystemvirtualisierung

Mit der Betriebssystemvirtualisierung soll die Gruppe der Technologien beschrieben werden, welche dafür eingesetzt werden, von einander unabhängige Betriebssystem-Instanzen auf ein und der selben Hardware zu betreiben („Abschottung der Betriebssystem-Instanzen“).

### Hardwarevirtualisierung

Die wohl am einfachsten zu verstehende Art ist die reine Hardwarevirtualisierung. Hierbei sind nicht die aktuelle Eigenschaften von Intel oder AMD-CPU's gemeint, sondern sogenannte Hardware-Partitions wie sie z. B. in IBMs System p [2] oder Fujitsus Sparc Enterprise Serie [3] realisiert sind. Vereinfacht ausgedrückt kauft der Kunde einen Schrank voller CPU's, Arbeitsspeicherriegel etc. Im Gegensatz zu normalen Servern muss jedoch bei der Erweiterung des Arbeitsspeichers nicht ein neues RAM-Modul aus dem Keller geholt und explizit in einen Server eingebaut werden. Vielmehr kann dies bei Hardware-Partitions „per Mausclick“ erfolgen, da die vorhandenen Ressourcen innerhalb des „Schanks“ frei auf die logischen Partitions verteilt werden können. Auf jeder so erstellten Partition kann dann ein entsprechendes Betriebssystem installiert werden. Die Virtualisierung erfolgt hierbei durch die Aufteilung der vorhandenen Hardware in kleinere Rechner (die erwähnten Partitions).

### Vollvirtualisierung

Vollvirtualisierung bezeichnet eine in Software verwirklichte Technologie. Sie kann sowohl als Hypervisor realisiert werden, das heißt die Virtualisierungsschicht läuft direkt auf der Hardware, oder als Hosted-Lösung (siehe unten) In diesen Abschnitt wird erst einmal nur die Hypervisor-Lösung betrachtet. Die Gast-Betriebssysteme müssen bei der Vollvirtualisierung nicht speziell



angepasst werden, da diese in der virtuellen Umgebung bekannte Hardware vorfinden. Während das Gastbetriebssystem denkt, auf physikalische Geräte (wie zum Beispiel Netzwerkkarten) zuzugreifen, sind diese jedoch nur virtuell vorhanden. Wenn nun ein Gastbetriebssystem mit einem anderen Rechner kommunizieren will, so sendet es beispielsweise über seine virtuelle Intel EPro1000-Netzwerkkarte Daten ins LAN. Diese Daten müssen nun von der Virtualisierungsschicht über die physikalische Netzwerkkarte weitergeleitet werden („Virtualisierungsoverhead“ für Ein- und Ausgabeoperationen). Reine Rechenoperationen werden direkt auf den physikalischen CPUs ausgeführt.

Die Problematik besteht hierbei darin, dass spezielle Befehle unterbunden werden müssen. Als Beispiel sei das Ausschalten einer virtuellen Maschine genannt. Um bei einem einfachen Modell zu bleiben, wird im Folgendem angenommen, dass hierfür das virtualisierte Betriebssystem den CPU-Befehl „Power-Off-Hardware“ ausführt. Würde dieser Befehl jedoch eins-zu-eins vom Hypervisor an den physikalischen Rechner weitergeleitet werden, würde der komplette Server mit allen VMs ausgeschaltet werden – anstatt einer einzelnen VM. Deshalb muss dieser Befehl abgefangen und anders verarbeitet werden – hier wird er durch den Befehl „Power-Off-VM“ ersetzt. Dies bedeutet, dass grundsätzlich sämtliche von virtuellen Maschinen ausgeführte Befehle auf solche kritischen Kommandos hin untersucht und bei Bedarf umgeschrieben werden müssen. Dieser Vorgang wirkt sich entsprechend

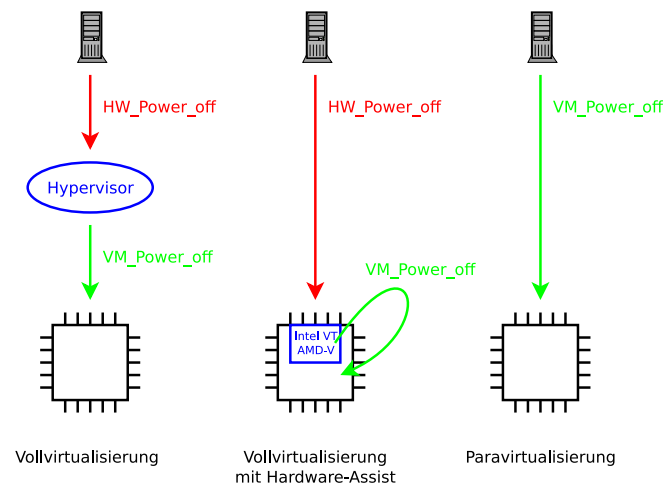
auf die Performance aus („Virtualisierungsoverhead“ für CPU-Befehle).

### Hardware-Assisted-Virtualisierungen

Genau hier setzen Hardware-Assisted-Virtualisierungen an. Bei diesen Lösungen übernimmt die Hardware das Umsetzen der Befehle von „Power-Off-Hardware“ zu „Power-Off-VM“. Dies hat zur Folge, dass die Virtualisierungsschicht sich nicht mehr so intensiv um die Befehle der virtuellen Maschinen kümmern muss. Die Technologie wird bei Intel-CPU's als Vanderpool Technology „Intel VT-x“ [4] bzw. bei AMD als AMD-Virtualization „AMD-V“ [5] (früher Code-name Pacifica) bezeichnet. In neueren Rechnergenerationen wird dieses Prinzip nicht mehr nur für CPUs sondern auch für Ein- und Ausgabe-Befehle verwendet (z. B. „Intel VT-d“).

### Paravirtualisierung

Als letzte Technologie in diesem Abschnitt ist die Paravirtualisierung zu nennen. Der Unterschied zu den bisher genannten Techniken ist, dass hierbei das Gastbetriebssystem angepasst werden muss. Die Anpassung sorgt dafür, dass die virtuelle Maschine weiß, dass sie nicht auf physikalischer Hardware läuft und somit direkt den Befehl „Power-Off-VM“ ausführt. Deshalb erzeugt Paravirtualisierung weniger Overhead mit dem Nachteil der nötigen Anpassung des Gastes. Dadurch war diese Technologie zuerst nur Linux-Kernel basierten VMs vorenthalten. Microsoft hat mittlerweile auch für Windows Server 2008 solche Anpassungen entwickelt („Enlightenments“ [6]), welche jedoch ausschließlich



Vereinfachtes Schema der Betriebssystemvirtualisierung.

von Microsoft-Virtualisierungslösungen (sprich von Microsofts Hyper-V) verwendet werden dürfen, wenn nicht explizit ein Vertrag zusätzlich mit Microsoft ausgehandelt wurde. Es ist davon auszugehen, dass Microsoft sich dies entsprechend honorieren lässt. Um dieses Problem etwas zu mildern, wurden spezielle Treiber entwickelt, welche bei einem vollvirtualisierten Gast zumindest Teile paravirtualisiert betreiben können (z. B. der paravirtualisierte SCSI-Treiber der VMware Tools).

## Anwendungsvirtualisierung

### Gehostete Vollvirtualisierung

Die gehostete Vollvirtualisierung wird in diesem Artikel der Gruppe der Anwendungsvirtualisierungen zugeordnet, während sie ebenfalls der Betriebssystemvirtualisierung zugeordnet werden könnte. Da jedoch die Virtualisierungsschicht



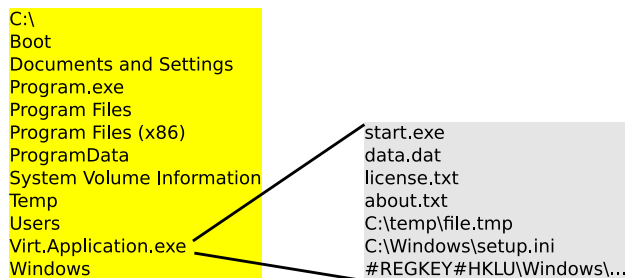


nicht direkt auf der Hardware läuft, sondern vielmehr ein Basisbetriebssystem erforderlich ist, innerhalb dessen die Virtualisierung als normale Anwendung läuft, wird sie diesen Abschnitt zugeordnet. Als bekannte Vertreter dieser Gattung wären unter anderem VirtualBox [7], Microsoft VirtualPC [8] und VMware Server [9]/Workstation [10] zu nennen. Die Vorteile liegen auf der Hand: Es ist nicht mehr die Virtualisierungsschicht für das Ansprechen der physikalischen Hardware verantwortlich, sondern das zu Grunde liegende Betriebssystem (z. B. Linux, Windows, Mac OS X). Somit müssen keine eigenen Treiber entwickelt werden. Beispielsweise unterstützt VMware ESX-Server 4 nur eine begrenzte Anzahl an Netzwerkkarten [11], während die Lösung mit VMware Server sozusagen alle von Linux und Windows unterstützten Geräte verwenden kann. Auch Citrix XenServer, welches auf einem Linux-Kernel basiert und somit theoretisch sämtliche Linux-Treiber verwenden könnte, unterstützt nur einen Teil derer. Der Hauptnachteil von Hosted-Lösungen ist jedoch,

dass das Virtualisierungsprogramm nur noch eines von vielen im Host-Betriebssystem ist und entsprechend auch als solches behandelt wird. Des Weiteren ist nicht garantiert, dass ein für Windows zertifizierter Treiber auch mit sämtlichen auf Windows ausführbaren Virtualisierungslösungen fehlerfrei funktioniert, während die Hersteller von Hypervisor-Produkten ihre Treiber genau für diesen Anwendungsfall testen.

### Terminal-Server

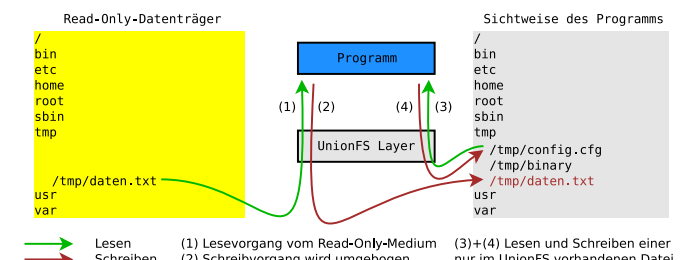
Die wohl bekannteste Art der Anwendungsvirtualisierungen sind Terminal-Server. Hierbei werden nicht verschiedene Betriebssystem-Instanzen parallel auf einem Rechner ausgeführt, sondern nur parallele Sitzungen mehrere Nutzer innerhalb eines Betriebssystems. Unter Windows wären Microsofts Terminal Services [12] und Citrix XenApp [13] zu nennen, während Linux mit einer Vielzahl von Lösungen aufwarten kann. Dies liegt höchstwahrscheinlich daran, dass Linux schon immer als Mehrbenutzer-System entwickelt wurde. So können bereits die Standard-Desktop-Manager (kdm, gdm) über XDMCP Sitzungen für Remote-Benutzer bereitstellen. Um Anwendungen in einer Mehrbenutzer-Umgebung voneinander abzuschotten, wird normalerweise auf Home-Verzeichnissen und Variablen gesetzt. Dabei liegen die Programmdateien nur einmal vor (z. B. **C:\Program Files\...** bei Windows bzw. **/usr/bin** bei Linux), während für das Speichern der Dokumente eine Variable **%HOMEPATH%** (z. B. **C:\Users\admin\**) bzw. **\$HOME** (z. B. **/home/admin/**) verwendet wird.



Aus Sicht der **start.exe** existiert die Datei **C:\Windows\setup.ini**, aus Sicht der **C:\Program.exe** nicht. 🔍

### Anwendungsvirtualisierungen

Eine verwandte Art sind die (für viele Anwender missverständlich) direkt als Anwendungsvirtualisierungen betitelten Lösungen. Linux-Benutzer kennen ein ähnliches Verfahren seit Jahren. Gemeint ist eine chroot-Umgebung [14]. Hierbei wird eine Anwendung in einer vom eigentlichen Betriebssystem abgekoppelten Umgebung („Sandbox“) ausgeführt. Die Technologien gehen hierbei heutzutage jedoch weiter und spiegeln eher das Verhalten von UnionFS [15] wieder. Die Produkte der führenden Hersteller nennen sich VMware ThinApp [16], Citrix XenApp Streaming [17] und Microsoft App-V [18]. Alle beruhen auf dem Prinzip, dass ein Programm denkt, es laufe ungehindert im Betriebssystem, während in Wirklichkeit jedoch sämtliche Lese- und Schreiboperationen abgefangen und bei Bedarf umgebogen werden können. Einem Programm kann somit simuliert werden, dass es z. B. Daten von **C:\temp** nach **D:\temp** kopieren kann. In Wirklichkeit existieren jedoch die Ordner nicht einmal im Dateisystem sondern nur innerhalb der virtualisierten Anwendungsumgebung. Unter Linux kann dies mit UnionFS abgebildet werden.



➡ Lesen (1) Lesevorgang vom Read-Only-Medium  
➡ Schreiben (2) Schreibvorgang wird umgebogen (3)+(4) Lesen und Schreiben einer nur im UnionFS vorhandenen Datei

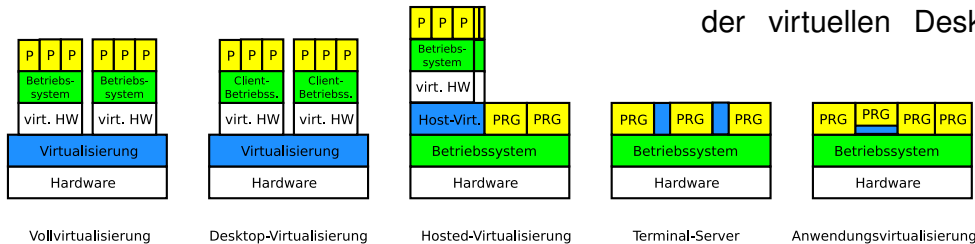
Funktionsweise von UnionFS. 🔍



Hierbei kann beispielsweise auf einer Read-Only-CD-Rom ein schreibbares Dateisystem simuliert werden, indem die Schreiboperationen entweder auf eine Festplattenpartition oder direkt in den Arbeitsspeicher umgeleitet werden.

### Desktopvirtualisierungen

Als Desktopvirtualisierung wird normalerweise die Virtualisierung von Client-Betriebssystem bezeichnet. Das heißt ein Windows XP läuft nicht mehr am Arbeitsplatz-Rechner sondern innerhalb einer virtuellen Server-Infrastruktur (ESX, XenServer, Hyper-V). Die Benutzer verwenden entweder einen Terminal-Services-Client oder direkt einen ThinClient mit Hilfe dessen sie die virtuellen Clients steuern können.



Anwendungsvirtualisierungslösungen in der Übersicht. 🔍

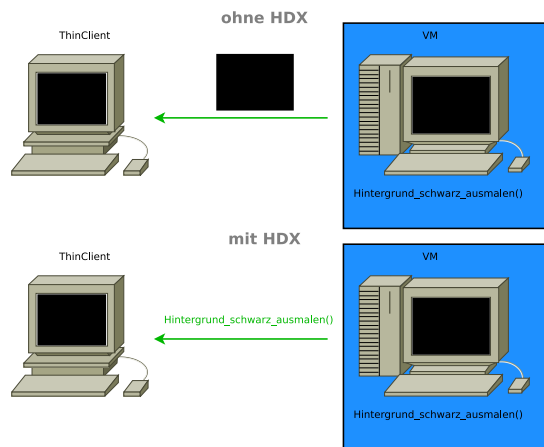
Vorteile sind hierbei, dass sämtliche Server- und Client-Betriebssysteme an einem zentralen Ort betrieben werden können (z. B. VMware vSphere-Umgebung im Serverraum). Somit ist eine zentrale Datenhaltung mit verbesserten Backup-, Performance- oder Kostenersparnis-Möglichkeiten gegeben. Für die Verwaltung

der virtuellen Desktop-VMs wurden VMware View [20] und Citrix Xen-Desktop [21] entwickelt. Einen entscheidenden Nachteil teilen sich jedoch alle Virtualisierungslösungen, die für die (Client-)Benutzer-Interaktion ausgelegt sind: die Grafikkartenperformance. Bei einem physikalischen Client-PC ist die Grafikkarte direkt mit dem Monitor verbunden, während bei einer Client-VM die Daten von der virtuellen Grafikkarte im Serverraum erst über das Netzwerk zur lokalen Grafikkarte und letztlich zum Monitor übertragen werden müssen.

	VMware	Citrix	Microsoft	OpenSource
Vollvirtualisierung	ESX3, ESXi3	–	–	Bochs, Qemu etc.
Hardware-Assisted-Vollvirtualisierung	ESX, ESXi, VMware Server, VMware Workstation, ..	XenServer im Vollvirtualisierungsmodus	Hyper-V im Vollvirtualisierungsmodus	Xen und KVM (+Derivate) im Vollvirtualisierungsmodus
Paravirtualisierung	ESX + ESXi für Linux und spezielle Treiber	XenServer im Paravirtualisierungsmodus für Linux und spezielle Treiber	Hyper-V im Paravirtualisierungsmodus für Win2k8 und spezielle Treiber für z. B. Win2k3	Xen und KVM (+Derivate)im Paravirtualisierungsmodus für Linux
Hosted-Virtualisierung	VMware Server, VMware Workstation, VMware Fusion	–	VirtualPC, XP-Mode in Windows7	HXEN (Hosted Xen), VirtualBox OSE etc.
Terminal-Server	–	XenApp	Microsoft Terminal-Services	XDMCP, LTSP, FreeNX,
Desktopvirtualisierung	VMware View	XenDesktop	– <sup>1</sup>	unbekannt
Anwendungsvirtualisierung	ThinApp	XenApp Streaming	Microsoft Desktop Virtualization (SCCM + App-V)	chroot, unionfs etc.
Client-PC-Betriebssystemvirtualisierung	VMware CVP2 <sup>2</sup>	XenClient <sup>2</sup>	–	–

<sup>1</sup>: Wie bei Microsoft nachzulesen ist [19], gibt es auch eine Microsoft Virtual Desktop Infrastructure. Dahinter verbirgt sich jedoch nach meinem Wissensstand nur das Lizenzmodell, Vista in einer virtuellen Umgebung betreiben zu dürfen, und kein reales Produkt seitens Microsoft.

<sup>2</sup>: Bisher nur angekündigt, noch nicht verfügbar



Funktionsweise von Citrix HDX.

Während VMware in der Servervirtualisierung der Marktführer ist, kann in diesem Bereich vor allem Citrix durch seine jahrelange Erfahrung im Terminal-Server-Bereich punkten. Unter anderem wird mit der HDX-Technologie [22] dieses Problem etwas gelindert. Ein Aspekt dieser Technik soll mit folgendem (wieder vereinfachten) Modell dargestellt werden: Innerhalb des virtuellen Clients wird der Befehl **Hintergrund\_schwarz\_ausmalen()** ausgeführt. Während dies normalerweise dazu führt, dass die virtuelle Grafikkarte nun den kompletten virtuellen Bildschirm schwarz ausmalt (und somit z. B. 1024x768 schwarze Pixel zum ThinClient übertragen werden müssen), sorgt HDX dafür, dass der eigentliche Befehl **Hintergrund\_schwarz\_ausmalen()** zum ThinClient übertragen wird. Somit müssen nur wenige Bytes für den Befehl versendet werden, welcher dann der lokalen (ThinClient-) Grafikkarte das Ausmalen des Bildschirms überlässt.

### Client-PC-Betriebssystemvirtualisierung

Ein weiterer Ansatz wird sowohl von Citrix mit XenClient [23] als auch von VMware mit VMware Client Virtualization Platform [24] verfolgt. Als Bezeichnung wäre hierfür am ehesten der Begriff Client-PC-Betriebssystemvirtualisierung angebracht, da wahrscheinlich die im ersten Abschnitt dieses Artikels aufgeführte Technik der Hardware-Assisted-Virtualisierung verwendet wird. Jedoch nicht nur für die CPU, sondern für sämtliche PCI-Geräte. Somit kann auch die Grafikkarte mittels HA-Virtualisierung performant von mehreren virtuellen Client-Betriebssystemen verwendet werden. Dies ist nötig, um aufwendige graphische Programme innerhalb der virtualisierten Systeme verwenden zu können. Als Beispiel sei hier der Support von 3D-Funktionalitäten wie DirectX oder OpenGL erwähnt.

### LINKS

- [1] <http://en.wikipedia.org/wiki/Virtualization>
- [2] <http://www.ibm.com/systems/p/>
- [3] <http://www.fujitsu.com/global/services/computing/server/sparcenterprise/>
- [4] [http://www.intel.com/technology/virtualization/technology.htm?iid=tech\\_vt+tech](http://www.intel.com/technology/virtualization/technology.htm?iid=tech_vt+tech)
- [5] <http://www.amd.com/virtualization>
- [6] <http://blogs.msdn.com/tvoellm/archive/2008/01/02/hyper-v-integration-components-and-enlightenments.aspx>
- [7] <http://www.virtualbox.org>
- [8] <http://www.microsoft.com/windows/virtual-pc/>
- [9] <http://www.vmware.com/products/server/>
- [10] <http://www.vmware.com/products/ws/>

- [11] <http://www.vmware.com/go/hcl>
- [12] <http://www.microsoft.com/windowsserver2008/en-us/rds-product-home.aspx>
- [13] <http://www.citrix.com/English/ps2/products/product.asp?contentID=186>
- [14] <http://de.wikipedia.org/wiki/Chroot>
- [15] <http://www.filesystems.org/project-unionfs.html>
- [16] <http://www.vmware.com/products/thinapp/>
- [17] <http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=163987>
- [18] <http://www.microsoft.com/systemcenter/appv/default.aspx>
- [19] <http://www.vmware.com/products/view/>
- [20] <http://www.citrix.com/English/ps2/products/product.asp?contentID=163057>
- [21] <http://hdx.citrix.com>
- [22] <http://www.youtube.com/watch?v=kxOWxjThzo0>
- [23] <http://www.vmware.com/company/news/releases/cvp-intel-vmworld.html>

### Autoreninformation

**Florian E.J. Fruth** beschäftigt sich seit über drei Jahren intensiv mit Virtualisierungslösungen von VMware, Citrix und der Open-Source-Community.

Diesen Artikel kommentieren





## Gefahren von Fremdquellen am Beispiel PPA von Martin Gräßlin

**F**remdpakete und -quellen gelten in den meisten Distributionen als gefährlich, und es wird immer wieder davor gewarnt, dass diese ein Update zerstören können. In diesem Artikel wird an einem Beispiel in Verbindung mit PPAs aufgezeigt, wie leicht ein einzelnes Paket ein Kubuntu-System unbenutzbar machen kann. Das Beispiel lässt sich leicht auf andere Distributionen wie openSUSE in Verbindung mit dem Build Service adaptieren.

### PPAs und Build Service

Personal Package Archives (kurz PPA) sind Bestandteil der Entwicklungsplattform Launchpad [1] der Firma Canonical und ermöglichen es jedem angemeldeten Nutzer, Debian-Pakete für verschiedene Architekturen zu erstellen und auf den Webservern zu hosten. Ein einzelnes PPA ist eine Paketquelle, welche sich wie jede andere in ein Ubuntu (oder jede Debian-basierte Distribution) integrieren lässt. openSUSE bietet mit dem Build Service [2] eine vergleichbare Architektur, welche jedoch nicht nur Pakete für die eigene Distribution, sondern auch für viele gängige Distributionen erstellen kann.

Für einen erfahrenen Paketbauer bieten beide Konzepte sehr gute Möglichkeiten, schnell und sicher Pakete zu erstellen und zu verteilen. Bei einem unerfahrenen Paketbauer besteht jedoch die Gefahr, dass ein „fehlerhaftes“ Paket erstellt

wird, welches bei einem Upgrade der Distribution zu Problemen führen kann.

#### Beispiel: Aurorae

Als fiktives Beispiel dient die KDE-Fensterdekoration Aurorae, welche erstmalig kurz vor der Veröffentlichung von KDE 4.3 auf kde-look.org zum Download angeboten wurde [3]. Aurorae hat jedoch eine zwingende Abhängigkeit auf KDE 4.3 und zwar in beide Richtungen, das heißt sie funktioniert weder mit KDE 4.2, noch wird der veröffentlichte Quellcode mit KDE 4.4 funktionieren. Ab KDE 4.4 wird die Dekoration direkt in KDEs Fenstermanager integriert und Bestandteil einer KDE-Standardinstallation sein. Für Ubuntu 9.10 „Karmic Koala“ wurde das Paket als **kwin-style-aurorae** in das Repository *universe* aufgenommen.

Für Ubuntu 9.04 „Jaunty Jackalope“ existiert jedoch kein Paket, da die Abhängigkeiten prinzipiell nicht erfüllt sind. Wer jedoch 4.3 verwendet, kann das Paket aus Karmic problemlos installieren. Ein Anwender – dies ist nun ein hypothetischer Fall – hat dennoch ein PPA eingerichtet und bietet die Dekoration als Paket **aurorae** für andere Anwender an. Der Name entspricht dabei nicht dem Namensschema für KDE-Fensterdekorationen in Debian/Kubuntu.

Bei einem Update auf Karmic Koala zeigt sich bereits das erste Problem: Es gibt nun zwei Pakete für die gleiche Software. Die Pakete enthalten

exakt die gleiche Version und exakt die gleichen Dateien.

Mit der Veröffentlichung von Karmic hat der Anwender das Interesse an seinem PPA verloren und benutzt selber das Paket aus den Quellen. Mögliche Aktualisierungen werden nicht mehr im PPA veröffentlicht. Wer Aurorae also aus dem PPA verwendet, wird mit möglichen Bugfixes nicht mehr versorgt. Das Paket in den Paketquellen würde jedoch gewartet werden oder Updates würden über die bekannten Kubuntu-PPAs veröffentlicht werden.

### Situation beim Upgrade

Nun ist ein weiteres halbes Jahr vergangen und es steht Ubuntu 10.04 „Lucid Lynx“ [4] vor der Tür. KDE 4.4 wurde mittlerweile veröffentlicht und Aurorae ist somit Bestandteil von KDE. Die Kubuntu-Maintainer passen ihre Abhängigkeiten entsprechend an und **kwin-style-aurorae** wird ein virtuelles Paket, welches von der neuen Version von **kde-window-manager** entfernt wird.

Wer also Aurorae aus den Paketquellen verwendet, hat beim Update keine Probleme. An Aurorae selbst wurde seit der initialen Veröffentlichung kaum gearbeitet, insbesondere werden die Dateien noch an die gleiche Stelle installiert. Von dem Paket aus dem PPA wissen die Maintainer jedoch nichts und sie haben ihre Abhängigkeiten daher nicht anpassen können. Das Paket wird also bei dem Upgrade nicht entfernt werden.



Nun gibt es immer noch Anwender, welche das Paket aus dem PPA verwenden. Die Installation liegt mittlerweile neun Monate zurück. Die Aktualisierung des Pakets **kde-window-manager** wird fehlschlagen, da es Dateien enthält, welche schon in einem anderen Paket enthalten sind. Das Update wird nicht fertig ausgeführt, der Anwender versteht zwar die Fehlermeldung nicht, hat aber gehört, dass man in so einem Fall über die Konsole das Upgrade beenden kann. Er macht auch den obligatorischen Neustart, meldet sich an, sieht den Desktop laden und kann plötzlich nichts mehr anklicken. Ein Fenster wurde sogar noch geöffnet, es fehlen aber die Dekorationen und es lässt sich nicht verschieben. Auch wenn versucht wird, das K-Menü zu öffnen, passiert nichts. KRunner startet noch, lässt sich aber nicht aktivieren. Der Rechner ist also praktisch unbenutzbar. Es lässt sich nicht einmal ein Browser öffnen, um im Forum nach Hilfe zu fragen. Was ist passiert?

Das Paket **kde-window-manager** konnte nicht aktualisiert werden. Die Abhängigkeiten wurden jedoch aktualisiert. Darunter das Paket **libkdecorations4**. Dieses stellt die Schnittstelle zwischen einer Fensterdekoration und dem Fenstermanager dar. Die Schnittstelle liegt nun in der Version 4.4 vor, KWin und Aurorae in der Version 4.3. Wie oben bereits erwähnt, ist Aurorae in der auf kde-look.org angebotenen Version nur für KDE 4.3 gedacht. Dies liegt daran, dass Aurorae eine Klasse verwendet, die „KCommonDecorationUnstable“ heißt. KWin bietet auf diese Klasse keine Binärkompatibilität an. Es ist nicht

garantiert, dass eine Dekoration, geschrieben gegen Version X der Klasse, mit Version X+1 funktionieren wird. Den Entwicklern ist dieses Problem bewusst und die Klasse ist eigentlich nur für Dekorationen gedacht, welche mit KWin veröffentlicht werden und immer direkt an die aktuelle Version angepasst werden. Es wird also angenommen, dass Dekorationen für 4.3 nur mit 4.3 betrieben werden, Dekorationen für 4.4 nur mit 4.4. Da Aurorae sich in KDE befindet, kann diese Annahme als korrekt angesehen werden. Durch das fehlgeschlagene Update ist eine Verletzung der Annahme aufgetreten.

Versucht man, den Fenstermanager manuell in der Konsole zu starten, so erhält man eine Fehlermeldung wie die folgende:

```
kwin: symbol-lookup error: /usr/lib/~/libkdeinit4\_kwin.so: undefined symbol~
: -ZTI26KDecorationFactoryUnstable
```

Das hier beschriebene Problem tritt in der Praxis bereits auf, da auf kde-look.org mehrere Dekorationen angeboten werden, welche die Unstable-API von KWin verwenden. Die Unstable-API wurde in KDE 4.2 eingeführt – mit dem Hinweis, dass sie nur in 4.2 gilt und in 4.3 wurde sie verändert, jedoch nicht entfernt. Es ist davon auszugehen, dass die Entwickler in 4.4 erneut Veränderungen vornehmen werden. Der Artikel zu KWin im ubuntuusers Wiki hat einen eigenen Abschnitt zu dem hier beschriebenen Problem [5]. Die aktuell vorgegebene Lösung würde jedoch nicht funktionieren, da vorausgesetzt wird, dass KWin und die Bibliothek in der gleichen Version vorliegen.

## Fazit

Dieser Artikel zeigt, wie leicht ein einziges Paket aus einer Fremdquelle bei einem Update das System „zerstören“ kann. Das Problem muss nicht einmal beim ersten Upgrade auftreten, sondern kann durchaus mehrere Versionen später erst erscheinen. Die Verbindung mit einem Fremdpaket herzustellen, welches man vor Monaten oder Jahren installiert hat, kann durchaus schwierig sein. Die Fehlersuche wird für einen Laien fast unmöglich.

Fremdpakete sollten daher immer mit Vorsicht benutzt werden.

## LINKS

- [1] <http://launchpad.net>
- [2] <https://build.opensuse.org>
- [3] <http://www.kde-look.org/content/show.php/Aurorae+Theme+Engine?content=107158>
- [4] <http://www.heise.de/open/news/meldung/145576>
- [5] <http://wiki.ubuntuusers.de/KWin#KWin-startet-nicht-mehr>

### Autoreninformation



**Martin Gräßlin** ist KDE-Workspace-Entwickler und unter anderem Autor der Fensterdekoration Aurorae.

*Diesen Artikel kommentieren*





## Der September im Kernel-Rückblick von Mathias Menzer

**B**asis 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.

### Kernel 2.6.31

Etwas später als in der vergangenen Ausgabe angekündigt (siehe „Der August im Kernel-Rückblick“, [freiesMagazin 09/2009 \[1\]](#)) wurde der Kernel 2.6.31 veröffentlicht [2]. Zu den herausragenden Neuerungen zählt die Unterstützung für USB 3.0, was dank der von der Intel-Entwicklerin Sarah Sharp beigesteuerten Treiber für xHCI (Extensible Host Controller Interface) ermöglicht wird (siehe „Der Juni im Kernel-Rückblick“, [freiesMagazin 07/2009 \[3\]](#)). Zwischenzeitlich ist auch Hardware in greifbare Nähe gerückt, die den neuen Standard nutzt, darunter ein Controller [4] sowie eine Festplatte [5].

Fastboot stammt ebenfalls aus Intels Schmiede und soll den Systemstart beschleunigen, indem die Suche und Initialisierung von Geräten und Partitionen sowie der Start von Diensten parallel abgewickelt wird. Die Verfügbarkeit der Kompressionsmethoden LZMA und BZIP2 bringen zwar keinen Leistungs- wohl aber einen Platzgewinn, da hierdurch das kompilierte Kernel-Abbild bis zu 33 Prozent gegenüber der herkömmlichen GZIP-Kompression verkleinert wird.

CUSE (Character Devices in Userspace) ist ebenfalls ein neuer Treiber, der aufbauend auf FUSE (Filesystem in Userspace) die Implementierung zeichenorientierter Geräte im Berechtigungskontext des Anwenders erlaubt. Erstes Beispiel ist die mitgelieferte Umsetzung von OSS Proxy [6], der eine Emulation des veralteten, jedoch noch immer weit verbreiteten, Open Sound Systems darstellt. Besitzer älterer ATI Radeon-Karten werden sich freuen, denn die Unterstützung für das in 2.6.29 eingeführte Kernel Mode Setting wurde für die Chipsätze R1xx, R2xx, R3xx, R4xx und R5xx erweitert, R6xx und R7xx befinden sich noch in Entwicklung. Verbesserungen beim Freigeben von Speicherbereichen sollen verhindern, dass Komponenten wie zum Beispiel verlinkte Bibliotheken, die von noch laufenden Programmen benötigt werden, aus dem Arbeitsspeicher entfernt werden. Dadurch wird ein flüssigeres Arbeiten am Desktop ermöglicht, da die langsameren Zugriffe auf die Festplatte zum Nachladen dieser Komponenten entfallen bzw. auf ein Minimum reduziert werden können.

### Con Kolvias

Ein Comic [7] hat den australischen Anästhesisten Con Kolvias dazu inspiriert, einen neuen Scheduler zu entwickeln, der insbesondere Systemen mit bis zu vier Prozessoren oder Kernen – also vorrangig Desktop-Systemen – besonders gute Leistungen zeigt [8]. Der „Brain Fuck Scheduler“ zeigt bei Anwendungen, die mit

vier Threads auf einem Vier-Kern-System ausgeführt werden bessere Leistungen, als mit beliebig vielen Threads gestartete Anwendungen mit dem aktuellen CFS (Completely Fair Scheduler). Kolvias, der sich bereits vor zwei Jahren mit dem Scheduler des Linux-Kernels beschäftigte und sich aus der Entwicklergemeinschaft im Streit über den CFS zurückzog, sieht das Problem darin, dass ein Scheduler nicht allen Anforderungen genügen kann: Ein Scheduler, der auf 4096 CPUs skaliert, kann trotzdem daran scheitern, ein Flash-Video auf einem System mit weniger als vier Kernen im Vollbildmodus flüssig abzuspielen.

Ingo Molnar, Autor des von Kolvias angegriffenen Completely Fair Schedulers, hat sich BFS mittlerweile angeschaut, bescheinigt ihm jedoch schlechtere Resultate [9]. Er erachtet BFS als spannenden und wagemutigen Ansatz. Die Priorität sieht auch er bei kleineren Systemen liegend, da 4096-CPU-Systeme eher selten sind und der Scheduler für diese dann auch optimiert würde. In diesem Zusammenhang bietet Molnar Kolvias die Zusammenarbeit an und verspricht, gute Ideen, die BFS liefert, auch zu folgen und in den Mainline-Kernel aufzunehmen.

### Hyper-V/Staging-Treiber

Beim Aufräumen seines staging-Zweiges, ein Bereich der Kernel-Code-Verwaltung, der der Aufnahme und Pflege kommender Treiber dient, stolperte Kroah-Hartmann über einige schlecht



oder ungepflegte Treiber. „*drivers/staging ist KEINE Mülldeponie für toten Code*“ beklagte er sich [10] und kündigte an, die nicht mehr erwarteten Treiber wieder aus dem Kernel zu entfernen. Traurig war dabei die Tatsache, dass auch der erst kürzlich von Microsoft freigegebene Treiber für Hyper-V (siehe „Der Juli im Kernel-Rückblick“, *freiesMagazin* 08/2009 [11]) davon betroffen war. Nachdem mit über 200 Patches der Treiber in eine Form gebracht wurde, die die Aufnahme in 2.6.32 ermöglichen, sollte er schon zu 2.6.33 wieder aus dem Kernel fliegen. Allein das Wiederauftauchen der Microsoft-Entwickler mit neuen Patches [12] nach dieser Ankündigung könnte Hyper-V vor diesem Schicksal bewahren. Für manche der 22 anderen Treiber auf Kroah-Hartmanns Abschussliste haben sich ebenfalls Maintainer gefunden, den verbliebenen wird jedoch der Rausschmiss blühen.

### Torvalds: „Linux ist aufgebläht“

Einer Intel-Studie zufolge verliert der Linux-Kernel mit jeder Version etwa zwei Prozent an Geschwindigkeit. Auf der in Portland stattfindenden LinuxCon äußerte Torvalds, dass er den Kernel inzwischen für zu aufgebläht halte. Eine Lösung dafür habe er jedoch nicht [13]. Tatsächlich nimmt das Volumen des Kernel immer weiter zu, wie auch die Studie „Who Writes Linux and Who Supports It“ (siehe „Der August im Kernel-Rückblick“, *freiesMagazin* 09/2009 [1]) belegt. Dies ist der stetig steigenden Anzahl an Funktionen und Treibern geschuldet, die mit einer ebenfalls größer werdenden Entwicklergemeinde ein-

hergeht. Allerdings sieht Torvalds dies nicht als so großes Problem, da die Stabilität des Kernels gut sei und Fehler rasch beseitigt würden.

### Echtzeit-Kernel für Debian

Pengutronix [14], Dienstleister für Embedded-Systeme unter Linux, stellt seit kurzem vorkompilierte Echtzeit-Kernel für Debian zur Verfügung [15]. Diese wurden mit PREEMPT\_RT [16], einer Echtzeit-Erweiterung für den Linux-Kernel, modifiziert und sollen Anwendern, die diese Funktionalität benötigen, das eigenhändige Kompilieren des Kernels ersparen. Einen Wermutstropfen gibt es allerdings: Die von Pengutronix bereitgestellten Kernel-Pakete basieren auf dem Vanilla-Kernel und enthalten nicht die Debian-spezifischen Kernel-Patches.

### Kernel 2.6.32-rc1

Fast drei Wochen nach Veröffentlichung von 2.6.31 beendete Torvalds das Zeitfenster, in welchem Neuerungen in den nächsten Kernel aufgenommen werden [17]. Der überwiegende Anteil der Änderungen von 2.6.32-rc1 liegen im Bereich der Treiber, gefolgt von Firmware und den Architekturen, wobei bei letzteren die ARM-Architektur stark im Vordergrund steht [18].

### LINKS

- [1] <http://www.freiesmagazin.de/freiesMagazin-2009-09>
- [2] <http://ikhaya.ubuntuusers.de/2009/09/10/der-kern-fuer-karmic-2-6-31-1673/>
- [3] <http://www.freiesmagazin.de/freiesMagazin-2009-07>

- [4] <http://www.heise.de/newsticker/meldung/145672>
- [5] <http://www.heise.de/newsticker/meldung/145828>
- [6] <http://userweb.kernel.org/~tj/oss/>
- [7] <http://xkcd.com/619/>
- [8] <http://www.linux-magazin.de/NEWS/Con-Kolivas-meldet-sich-mit-neuem-Scheduler-zurueck>
- [9] <http://www.linux-magazin.de/NEWS/Ingo-Molnar-testet-neuen-Brain-Fuck-Scheduler>
- [10] <http://lkml.org/lkml/2009/9/3/3>
- [11] <http://www.freiesmagazin.de/freiesMagazin-2009-08>
- [12] <http://lkml.org/lkml/2009/9/10/297>
- [13] <http://www.pro-linux.de/news/2009/14727.html>
- [14] [http://www.pengutronix.de/index\\_de.html](http://www.pengutronix.de/index_de.html)
- [15] <http://www.pro-linux.de/news/2009/14750.html>
- [16] <http://rt.wiki.kernel.org/>
- [17] <http://lkml.org/lkml/2009/9/27/166>
- [18] <http://www.linux-magazin.de/NEWS/Kernel-2.6.32-geht-in-die-abschliessende-Testphase>

### Autoreninformation



**Mathias Menzer** wirft gerne einen Blick auf die Kernel-Entwicklung, um mehr über die Funktion von Linux zu erfahren und seine Mitmenschen mit seltsamen Begriffen und unverständlichen Abkürzungen verwirren zu können.

*Diesen Artikel kommentieren*





## Java, Teil 1 – Einführung in eine moderne Sprache von Raoul Falk

**JVM, JFC, Applet, Swing ... Viele Begriffe schwirren seit nunmehr 13 Jahren in den Welten der IT herum, die ihren Ursprung in der Sprache Java haben. Dieser Artikel als erster Teil einer kleinen Serie soll die Java-Welt ein wenig beleuchten und den Leser in die Lage versetzen, selbst kleine Programme in Java zu schreiben und Code zu lesen, ohne sich in historischen oder technischen Einzelheiten zu verlieren.**

### Grundsätzliches

Um ein Java-Programm lauffähig zu machen, sind drei Schritte nötig. Zuerst schreibt der Entwickler den Quellcode als java-Datei. Diese Datei kann nun durch den Compiler in eine class-Datei kompiliert werden. Es entsteht ein Bytecode, der immer noch plattformunabhängig ist. Man kann diesen Bytecode auf jeder Plattform ausführen, auf der die Java-Laufzeitumgebung (JVM - Java Virtual Machine) installiert ist. Der letzte Schritt zur Ausführung ist die Übersetzung des Bytecodes in die Befehle der jeweiligen Plattform durch die Laufzeitumgebung.

### Die Wahl des Editors

Es ist eigentlich jeder Editor geeignet, um in Java zu programmieren. Man sollte sich aber den Komfort von Syntaxhervorhebung gönnen, um sich die Arbeit ein wenig zu erleichtern. Für GNOME eignet sich der hauseigene Editor gedit, für KDE beispielsweise Kate. Mit zunehmender Komplexität der Programme möchte man aber

auf eine ausgereifte Entwicklungsumgebung zurückgreifen.

Die wichtigsten Umgebungen zur Java-Entwicklung sind ohne Zweifel NetBeans [1] und Eclipse [2]. Beides sind freie Programme, die in den gängigsten Distributionen aus den Paketquellen zu beziehen sind und jeweils ihre eigenen Vorzüge und Schwächen haben. Immer wieder zu beobachtende „Glaubenskriege“ bezüglich der Wahl der Umgebung, die meist ergebnislos ausgefochten werden, sollten von keinem seriösen Entwickler ernst genommen werden. Der offensichtlichste Unterschied ist wohl, dass NetBeans einen grafischen GUI-Builder enthält, mit dem man per Drag & Drop optisch ansprechende GUIs erstellen kann, ohne tiefgehende Kenntnisse über die Swing-Bibliothek [3] zu benötigen.

Andererseits erzeugt NetBeans mit seiner Vorlage für eine grafische Applikation bereits eine erhebliche Menge Code (300 Zeilen), welcher für Einsteiger und vielleicht sogar für Amateure sehr schwer verständlich ist. Es besteht also durchaus die Gefahr, mit dem GUI-Builder in eine Sackgasse zu laufen, wenn man den produzierten Code nicht vollständig versteht. Daher ist ein Einstieg in Swing mit dem GUI-Builder nur eingeschränkt zu empfehlen. Für Eclipse ist ein GUI-Builder bis dato nur in einem kommerziellen Zusatzpaket erhältlich. In der nächsten Ausgabe wird die Grafikprogrammierung unter Java noch ausführlich

besprochen, ebenso die Vorbereitung einer Entwicklungsumgebung. In jedem Falle sind beide Umgebungen absolut ausgereift, reich an Features, leicht erweiterbar und nehmen eine absolut gleichberechtigte Stellung im Bereich der professionellen Java-Entwicklung ein.

### Installation und Vorbereitungen

Da Sun Java [4] sehr verbreitet ist, werden die meisten Nutzer bereits diese Laufzeitumgebung installiert haben. Um dies zu überprüfen, fragt man die aktuell benutzte Version mit

```
$ java -version
java version "1.6.0_10"
Java(TM) SE Runtime Environment
(build 1.6.0_10-b33)
```

ab. Die aktuellste Version von Sun Java ist „1.6.0\_16“. Sollte die Versionsabfrage etwas in der Art

```
java version "1.5.0"
gij (GNU libgcj) version 4.3.2
```

ausgeben, hat man kein Sun Java, sondern ein anderes Java (hier GNU Compiler for Java [5]) installiert, was aber auch nicht weiter schaden sollte.

Alternativ hilft auch ein Besuch der offiziellen Sun-Testseite [6], die aber natürlich nur ein Ergebnis liefert, wenn Java-Unterstützung im Browser eingerichtet ist.



Nun braucht man noch die notwendigen Bibliotheken, um Programme selbst kompilieren zu können. Zunächst muss man sich nun entscheiden, welche Version von Java man installieren möchte. Das Unternehmen Sun ist Urheber von Java und stellt mittlerweile Versionen für Linux bereit, die zumindest unter der i386- und amd64-Architektur laufen. Man benötigt das JDK (Java Development Kit), welches man nicht mit der JRE (Java Runtime Environment) verwechseln darf, welche nur das Ausführen von Java-Programmen ermöglicht. Das offizielle Java von Sun soll zukünftig unter der GPL veröffentlicht werden, so dass man diesbezüglich wohl keine Bedenken haben sollte. Man installiert das offizielle Java beispielsweise durch das Paket **sun-java6-jdk**. Die bereits jetzt komplett freie Version OpenJDK lässt sich in den gängigsten Distributionen durch das Paket **openjdk-6-jdk** installieren. Sollten dennoch außerordentliche Probleme auftreten, hilft Ubuntu-Nutzern die Wikiseite [7] in den meisten Fällen weiter.

## Das erste Programm

Das beliebte „Hello World“ ist in Java auf den ersten Blick komplex. Es wird jedoch jeder einzelne Begriff genau erklärt, um niemandem mit Halbwissen und Vermutungen zurückzulassen.

Der Quelltext wird als Datei **MyClass.java** gespeichert. Danach bewegt man sich in der Konsole in das entsprechende Verzeichnis und kompiliert das Programm mit

```
$ javac MyClass.java
```

Anschließend startet man das Programm mit

```
$ java MyClass
```

**Hinweis:** Die Endung **.class** wird nicht mit angegeben.

```
class MyClass {
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
    // Dies ist ein einzeiliger Kommentar.

    /* Dies ist ein Blockkommentar,
       der ueber mehrere Zeilen geht.
       */
}
```

Listing 1: *MyClass.java*

## Erklärung des Beispiels

Die erste Zeile leitet die Klasse ein, den äußeren Rahmen einer Java-Anwendung. Man beachte, dass der Bezeichner der Klasse (hier **MyClass**) groß geschrieben wird.

Die zweite Zeile ist die Kopfzeile der Hauptmethode der Klasse, die immer **main** heißen muss. Eine Methode ist ein Unterprogramm. Hat sie einen Rückgabewert, spricht man von Funktionen, ansonsten auch von Prozeduren. Sie ist der Einstiegspunkt, an dem ein Programm mit der Ausführung beginnt. Die Kopfzeile der Hauptmethode muss in jedem Programm genau so aussehen. **public** besagt, dass die Methode von

außerhalb der Klasse aufgerufen werden kann, **static** besagt, dass die Methode direkt aus der Klasse aufgerufen wird (ohne Objekt), **void** legt schließlich fest, dass die Methode keinen Rückgabewert hat. In den runden Klammern steht der Parameter, der dieser Methode übergeben wird.

In diesem Falle ein Array (eine Art von Liste) von Strings (Zeichenketten), auf die in der Methode unter dem Namen **args** (Argument) zugegriffen werden kann.

Nun beginnt eingeschlossen von den geschweiften Klammern der eigentliche Programmcode. Es wird die Methode **println** aufgerufen, die unter dem Pfad **System.out** erreichbar ist. Die Methode gibt eine Zeichenkette mit Zeilenumbruch auf der Konsole aus. Diese Zeichenkette wird in runden Klammern übergeben.

Am Ende jeder Anweisung muss ein Semikolon stehen (auch wenn es die letzte ist).

Schließlich wird die Klasse noch durch eine geschweifte Klammer geschlossen. Das sind zunächst eine Menge Schlüsselwörter für eine so simple Funktionalität. Doch ist gerade diese strikte Strukturierung unter Java in vielen Situationen vorteilhaft.

## Was zeichnet Java aus?

Wie am Beispiel erkennbar, ist Java sehr streng strukturiert. Dies ist nicht zu vergleichen mit Sprachen, die auch als Skripte ausgeführt werden



können wie zum Beispiel Python. Der Code wird dadurch umfangreicher. Allerdings erkennt der Compiler Fehler im Code sehr präzise und zwingt dem Entwickler einen sauberen und typsicheren Stil auf. Typsicherheit wird bei Java sehr groß geschrieben. Das bedeutet, dass zu jeder Zeit während der Laufzeit des Programms gewährleistet ist, dass eine Variable stets den richtigen Typ hat und so verwendet werden kann, wie es die Programmlogik erfordert. Es kann also nicht passieren, dass ein Programm anstandslos kompiliert, obwohl eine Zeichenkette (String) durch eine ganze Zahl (Integer) dividiert wird (was eine nicht definierte Operation wäre).

### Syntaktisches

Die wichtigsten Syntaxregeln sind bereits im Beispiel erkennbar. Klassen, Methoden und Blöcke (logisch zusammenhängende Anweisungen) werden von geschweiften Klammern umschlossen. Jede Anweisung wird mit einem Semikolon beendet. Methoden und Variablen werden mit Angabe ihres Pfades aufgerufen. Unterverzeichnisse werden mit Punkten getrennt.

Fügt man eine weitere Klasse hinzu und legt sie im gleichen Verzeichnis ab wie die andere Klasse, sind sich die Klassen untereinander bekannt.

```
class MyClass2
{
    static String text="Hallo
Welt";
}
```

Listing 2: *MyClass2.java*

Diese Klasse enthält nur eine Variable, nämlich eine Zeichenkette (**String**), die **text** heißt und den Inhalt **Hallo Welt** hat. Sie ist als **static** deklariert, damit sie eine Eigenschaft der Klasse ist, nicht von einem Objekt (dazu später mehr). Diese Klasse lässt sich kompilieren, jedoch nicht ausführen. Es fehlt die **main**-Methode. Doch die Klasse aus dem ersten Beispiel kann nun auf die Variable der zweiten Klasse zugreifen. Man ergänze die Hauptmethode der Klasse **MyClass** durch die Anweisung

```
System.out.println(
(MyClass2.text);
```

Sind beide Klassen (neu) kompiliert, wird auf der Konsole nun auch **Hallo Welt** ausgegeben:

```
$ javac MyClass.java
$ javac MyClass2.java
$ java MyClass
Hallo Welt
```

### Primitive Datentypen und Mathematik

In Java kann man sehr ähnlich wie in jeder anderen Programmiersprache mathematische Operationen durchführen. Die gängigsten Datentypen und Operationen sollen im folgendem Beispiel gezeigt werden.

```
class Beispiel {
    public static void main(String[] args) {

        // Die Variable zahl1 wird deklariert.
        int zahl1;

        // zahl1 wird initialisiert es wurde
        // ihr ein Wert zugewiesen.
        zahl1 = 7;

        // zahl2 wird in einer Anweisung
        // deklariert und initialisiert.
        int zahl2 = 3;
        System.out.println(zahl1 + zahl2);
        // Ausgabe: 10
        System.out.println(zahl1 * zahl2);
        // Ausgabe: 21
        System.out.println(zahl1 / zahl2);
        // Ausgabe: 2

        // "double" ist genauer
        double zahl3 = 7.0;

        System.out.println(zahl3 / zahl2);
        // Ausgabe: 2.33333...
    }
}
```

Listing 3: *Beispiel.java*

Warum wird bei der ersten Division als Ergebnis der Wert 2 ausgegeben? In dem Beispiel wird mit ganzen Zahlen (**int**) gerechnet. Die Nachkommastellen werden einfach abgeschnitten. Wird ein genaues Ergebnis gewünscht, muss ein an-



derer Datentyp benutzt werden, wie zum Beispiel **double**.

## Klassen und Objekten

Objekte können in Java für nahezu alle greifbaren Elemente des Programms stehen. Diese können simpel (eine Zahl) oder komplex (eine Liste von Hashmaps, in denen offene Datenbankverbindungen auf die jeweiligen Nutzer abgebildet werden) sein. Eine Tatsache muss allerdings beachtet werden: Primitive Datentypen sind in Java keine Objekte. Dies sind beispielsweise **int** oder **double** (später dazu mehr). Jedes Objekt hat einen bestimmten Typ. Im Beispiel **MyClass2** ist die Variable **text** ein Objekt vom Typ **String**. Somit ist sichergestellt, dass ein Objekt nur so benutzt werden kann, wie es für seinen Typ definiert ist. Ist für Objekte des Typs „Turnschuh“ beispielsweise die Methode „volltanken“ nicht definiert, obwohl sie im Code aufgerufen wird, würde der Compiler dies sofort erkennen.

```
class Person {
    public String vorname;
    public String nachname;

    public void printName()
    {
        System.out.println(vorname+" "+nachname);
    }
}
```

Listing 4: *Person.java*

Eine Klasse ist in Java nichts anderes als ein Bauplan für Objekte eines bestimmten Typs. Der

Bezeichner der Klasse ist gleichzeitig der Bezeichner des Typs der Objekte, die Instanzen dieser Klasse sind.

Es wurde ein neuer Datentyp namens **Person** erzeugt. Man beachte, dass die beiden Variablen **vorname** und **nachname** nun nicht als **static** deklariert werden. Das hat zur Folge, dass jedes Objekt vom Typ **Person** nun die beiden Variablen „hat“. Jede Person hat ihre eigenen Variablen. Somit sind die Variablen keine Eigenschaften der Klasse mehr, sondern von den Instanzen, die sie erzeugt. Die Methode **printName** ist ebenfalls nicht statisch, sie wird also nur von Objekten aufgerufen. Die runden Klammern nach dem Bezeichner sagen aus, dass die Methode keinen weiteren Parameter erwartet. Sie gibt den Vornamen, gefolgt von einem Leerzeichen (Zeichenketten werden mit + verbunden), und dann den Nachnamen auf der Konsole aus.

Das nächste Beispiel **Person2.java** erzeugt nun eine Person, weist den Variablen Werte zu und ruft die Methode **printName** auf.

Was geschieht in diesem Programm? Der Einstiegspunkt ist wieder die **main**-Methode. Es wird eine Variable vom Typ **Person** mit dem Bezeichner **einePerson** deklariert. Dieser Variable wird ein neu erzeugtes Objekt zugewiesen, das durch **new** erzeugt wird.

```
class Person {
    public String vorname;
    public String nachname;

    public void printName()
    {
        System.out.println(vorname+
            "+" "+nachname);
    }

    public static void main(String[] args)
    {
        Person einePerson = new
            Person();
        einePerson.vorname = "Hans";
        einePerson.nachname = "
            Maier";
        einePerson.printName();
    }
}
```

Listing 5: *Person2.java*

Dadurch wird der sogenannte Konstruktor aufgerufen, der eine Methode ist, die ein neues Objekt zurückgibt. Hier steht kein Parameter in den runden Klammern, da der immer implizit vorhandene parameterlose Konstruktor aufgerufen wird. Nun kennt die Laufzeitumgebung eine Person mit dem Bezeichner **einePerson**, die jedoch noch keine zugewiesenen Werte für **vorname** und **nachname** hat. Dies ändert sich in den nächsten beiden Anweisungen. Über die Punktnotation werden die Variablen aufgerufen, und ihnen ein Wert zugewiesen. Letztendlich werden diese Va-





riablen durch Aufruf von **printName** mit der Person **einePerson** auf der Konsole ausgegeben.

### Weiterführendes zu Klassen

Das Klassenkonzept kann sehr intuitive Programmierung ermöglichen. Es ist beispielsweise möglich, mit dem Schlüsselwort **extends** eine Klasse aus einer anderen Klasse abzuleiten. Man spricht von Vererbung. Man könnte etwa eine Klasse **Mitarbeiter** erstellen, welche die Variable **personalnummer** (deklariert als **int**, also als ganze Zahl) enthält, und diese Klasse von **Person** ableiten. Jeder Mitarbeiter hätte damit automatisch auch die Variablen **vorname** und **nachname**, sowie die Methode **printName**, ohne dass man dies noch einmal in den Code schreiben muss. Die Syntax wäre:

```
class Mitarbeiter extends Person {
    public int personalnummer;
}
```

Listing 6: *Mitarbeiter.java*

Nun „ist“ jeder Mitarbeiter auch eine Person. Und alle Methoden die für eine Person definiert sind, können auch von einem Mitarbeiter aufgerufen werden. Umgekehrt ist das natürlich nicht der Fall. Eine Person, die kein Mitarbeiter ist, hat die Variable **personalnummer** nicht. Der Compiler würde dies bemerken.

Es kann jedoch auch zu sehr abstrakten Beziehungen zwischen den Klassen kommen, die nicht unbedingt dem intuitiven Verständnis entsprechen. Es ist beispielsweise möglich, dass ein Objekt eine Variable hat, die eine Liste ist, in dem das Objekt selbst noch einmal enthalten ist. Der Fantasie sind dort keine Grenzen gesetzt. Man denke nur an ein Kind, das seinen Vater als Variable hat, während der Vater auch seine Kinder als Variablen hat.

### Ausblick

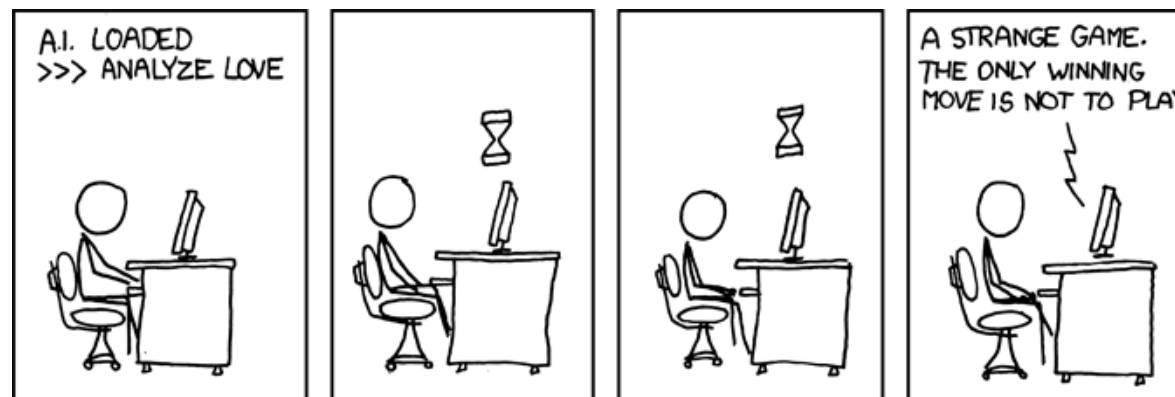
Im nächsten Teil der Serie wird die Grafikprogrammierung mit Java ausführlich beschrieben.

- [1] <http://www.netbeans.org/>
- [2] <http://www.eclipse.org/>
- [3] [http://de.wikipedia.org/wiki/Swing\\_\(Java\)](http://de.wikipedia.org/wiki/Swing_(Java))
- [4] <http://www.java.com/de/>
- [5] <http://gcc.gnu.org/java/>
- [6] <http://www.java.com/de/download/help/testvm.xml>
- [7] <http://wiki.ubuntuusers.de/Java>

### Autoreninformation

**Raoul Falk** programmiert Java und .NET an der Universität Duisburg-Essen.

[Diesen Artikel kommentieren](#)



„Game Theory“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/601>

## Ordnung ins Chaos mit Org-Mode von Rainer König

**D**er Satz „Ich habe keine Zeit ...“ ist die Standardantwort, die man oft genug zu hören bekommt, wenn man jemanden fragt, ob er eine Aufgabe übernehmen will. Interessanterweise haben alle Menschen genau gleich viel Zeit, nämlich 24 Stunden am Tag, und das Vorhandensein von „Extratagen“ oder „Zeitguthaben bei der Zeitbank“ gehört eher ins Reich der schriftstellerischen Fantasie.

### Literatur zum Thema Zeitmanagement

Das Thema Zeitmanagement betrifft sehr viele Menschen und es gibt eine große Anzahl von Büchern zum Thema. Angefangen beim deutschen Zeitmanagement-Papst Dr. Lothar Seiwert mit seiner Empfehlung „Wenn Du es eilig hast, gehe langsam“ bis hin zum amerikanischen Autor David Allen, dessen Buch „Getting Things Done“ einfach helfen soll, Dinge zu erledigen.

Die grundlegende Idee beim Zeitmanagement ist in jedem Fall die, dass man seine Aufgaben irgendwo aufschreiben soll. David Allen argumentiert hier sehr gut mit der Belastung eines Gehirns, das sich unzählige Sachen merken soll und ständig Angst hat, irgendetwas Wichtiges zu vergessen. Hat man hingegen alles irgendwo aufgeschrieben, dann reduziert sich der Merkwand darauf, sich zu erinnern, wo man es aufgeschrieben hat. Als direkte Folge dieses „Kopfleerens“ reduziert sich der Stress, den man empfindet, wenn man mit Aufgaben überschüttet wird.

Den Autoren der Zeitmanagement-Bücher ist es egal, ob diese Aufgabenlisten auf Papier oder in elektronischer Form vorliegen, aber als

```

emacs@seneca.nathansoft.local
File Edit Options Buffers Tools Agenda Help
* Freies Magazin :LINUX:
- Artikel für Freies Magazin
- Autoren-Richtlinien (PDF)
:PROPERTIES:...
** PROJ Artikel über orgmode schreiben
DEADLINE: <2009-08-26 Mi>
*** ENTWURF Literaturverzeichnis [5/5]...
*** ENTWURF Linkliste [2/2]...
*** ENTWURF Warum orgmode...
-1:--- privat.org 82% L139 (Org)-----
Global list of TODO items of type: ALL
Available with `N r': (0)ALL (1)TODO (2)NEXT (3)WAITING (4)SOMEDAY (5)DELEGATED
(6)PROJ (7)DONE (8)FORWARDED (9)CANCELLED (10)IDEE
(11)ENTWURF (12)FINAL
privat: PROJ Artikel über orgmode schreiben :LINUX::
privat: ENTWURF Literaturverzeichnis [5/5] :LINUX::
privat: ENTWURF Linkliste [2/2] :LINUX::
privat: ENTWURF Warum orgmode :LINUX::
privat: ENTWURF Grundlegende Org-Mode Prinzipien :LINUX::
privat: IDEE Kopf frei machen :LINUX::
privat: IDEE Tägliche Nutzung :LINUX::
privat: IDEE Review :LINUX::
-U:~*- *Org Agenda* All L4 (Org-Agenda Week Grid)-----
Modification not yet reflected in Agenda buffer, use `r' to refresh

```

Der Org-Mode in Emacs. 🔍

Computernutzer ist man natürlich bestrebt, den PC für so etwas zu nutzen.

### Zeitmanagement-Software

Software, die dem Nutzer beim Zeitmanagement helfen soll, gibt es fast wie Sand am Meer. Jeder Desktop bringt einen „Organizer“ mit, in dem man Tageskalender und Aufgabenlisten nutzen kann. Es gibt auch Programme wie ThinkingRock [1], die eine sehr detaillierte Umsetzung der Ideen aus David Allens Buch darstellen.

Andere Projekte haben es sich zum Ziel gesetzt, unter dem Begriff „Groupware“ das Zeit- und Aufgabenmanagement für ein ganzes

Team zu bewältigen. Solche Ansätze scheitern aber oft genug daran, dass das Team nicht homogen in der Struktur ist und unterschiedliche Ansichten bei den Teammitgliedern darüber bestehen, wie detailliert man Arbeitsfortschritte im Groupware-Tool dokumentieren soll.

Deswegen sollte man sich zunächst einmal mit dem „Selbstmanagement“ beschäftigen, mit dem man Ordnung in das eigene organisatorische Chaos bringt, wobei der Emacs Org-Mode helfen kann.

### Emacs als Organisationstalent

Org-Mode [2] ist eine Erweiterung des Texteditors Emacs [3]. So ist es nur konsequent, wenn einen die Org-Mode-Webseite mit dem Titel „Org-Mode: Your Life in Plain Text“ begrüßt. Org-Mode basiert tatsächlich auf der Interpretation des Textes im Editorbuffer und genau das macht ihn so einfach einsetzbar. Wenn andere Applikationen eine schöne grafische Benutzeroberfläche haben, dann sieht das zwar oft toll aus, ist aber beim schnellen Arbeiten eher hinderlich, weil



man zu oft zwischen Tastatur und Maus hin und her wechseln muss. Org-Mode beschränkt sich auf die Bedienung via Tastatur und wenn man ihn täglich benutzt, dann hat man die notwendigen Tastenkombinationen bald verinnerlicht.

Org-Mode bietet auch die Funktionalität eines „Outliners“. Damit kann man große Aufgaben (chinesisches Sprichwort: „Auch die längste Reise beginnt mit dem ersten Schritt.“) in kleinere „Unteraufgaben“ zerlegen, die dann nicht mehr so bedrohlich aussehen und an deren Erledigung man einen Projektfortschritt erkennen kann.

Da in Org-Mode alles Text ist, kann man ihn auch sehr schön zum Schreiben von Notizen zu den einzelnen Aufgaben nutzen. Diese Notizen können auch mit Zeitstempeln versehen werden, was letztlich einer lückenlosen Dokumentation der Tätigkeiten dient. Abgeschlossene Aufgaben können auch sehr einfach in anderen Dateien archiviert werden. So kann man diese Archive als „Gedächtniserweiterung“ nutzen.

### Org-Mode installieren

Org-Mode ist ein Emacs-Mode und damit wahrscheinlich bereits in der Emacs-Standardinstallation der verwendeten Linux-Distribution enthalten. Allerdings sollte man trotz allem die neueste Version von der Org-Mode-Webseite verwenden, da Org-Mode sehr häufig aktualisiert wird und ständig neue und nützliche Funktionen hinzukommen.

Auf der Org-Mode-Webseite findet man unter Download einen Link auf „gzipped tar archive“.

Dies ist der Link zur aktuellen stabilen Version [4]. Man speichert dieses Archiv in einem Ordner und entpackt es dort. Nach dem Wechsel in das entstandene Verzeichnis (in der Konsole) kann man mit

```
$ make
$ make install
```

den Quellcode durch den Emacs-Lisp-Compiler übersetzen und das Ergebnis danach installieren.

### Aktivierung

Um Org-Mode zu aktivieren, fügt man ans Ende der Datei `.emacs`, die sich im Heimatverzeichnis des jeweiligen Benutzers befindet, folgende Zeilen ein:

```
(add-to-list 'auto-mode-alist '("\.org\\'" . org-mode))
(global-set-key "\C-cl" 'org-store-link)
(global-set-key "\C-ca" 'org-agenda)
(global-set-key "\C-cb" 'org-iswitchb)
```

Listing 1: `.emacs`

Die erste Zeile sorgt dafür, dass Dateien mit der Endung `.org` automatisch im Org-Mode geöffnet werden. Die anderen Zeilen definieren Org-Mode-spezifische „Hotkeys“.

Um zu kontrollieren, ob alles geklappt hat, startet man Emacs mit einem Dateinamen, der auf `.org` endet (die Datei muss nicht existieren):

```
$ emacs example.org
```

Nun sollte Emacs den Buffer für diese Datei in den Org-Mode umschalten. Das erkennt man an dem „(Org)“ in der Statuszeile dieses Buffers. Außerdem gibt es ein Menü „Org“, in dem man den Punkt „Documentation » show version“ aufruft. Dann sollte in der untersten Zeile von Emacs „Org-Mode version XXX“ erscheinen, wobei XXX die jeweilige Version angibt.

### Grundlegende Prinzipien

Org-Mode arbeitet mit beliebigen Dateien. Hat der Dateiname die Endung `.org`, wird Emacs beim Öffnen der Datei gleich den Org-Mode aktivieren. Man kann so z. B. berufliche Aktionslisten von privaten Aktionslisten trennen und trotzdem beides in Org-Mode organisieren.

Ganz abstrakt betrachtet ist Org-Mode ein „Outliner“, in dem man Stichpunkte hierarchisch gliedern kann. Diese Stichpunkte (oder im Org-Mode-Sprachgebrauch „Kopfzeilen“

genannt) beginnen immer mit einem oder mehreren „Sternchen“. Die Anzahl der Sternchen bestimmt dabei die Hierarchie-Ebene:

```
* erste Ebene
** zweite Ebene
*** dritte Ebene
```

Mit der `Tab`-Taste kann die Sichtbarkeit der Ebenen geändert werden. Normalerweise sieht man nur die oberste, erste Ebene. Mit dem ersten



Drücken der `Tab`-Taste wird die erste Unterebene geöffnet, mit einem zweiten Drücken rekursiv alle Unterebenen und mit einem dritten Drücken wird wieder nur die erste Ebene angezeigt.

### ToDo-Listen

Ein weiteres wichtiges Konzept sind die **TODO**-Schlüsselwörter. Wenn die Kopfzeilen „Aktionen“ beschreiben sollen, dann soll hier ein Schlüsselwort stehen, das aussagt, was zu tun ist, oder ob diese Aktion schon erledigt ist. Eine ein wenig an David Allens „Getting Things Done“ angelehnte Liste solcher Schlüsselwörter kann so aussehen:

```
#+TODO: TODO(t) NEXT(n) WAITING(w) ~
SOMEDAY(s) DELEGATED(g) PROJ(p) | ~
DONE(d) FORWARDED(w) CANCELLED(c)
```

Eine solche Zeile (tatsächlich nur eine Zeile, also ohne Umbruch) am Beginn einer Org-Mode-Datei definiert diese Schlüsselwörter für die Benutzung in dieser Datei. Alles, was links vom `|` (Pipe-Symbol) steht, sind dabei Schlüsselwörter für unerledigte Aktionen, alles, was rechts davon steht, sind Schlüsselwörter für erledigte Aktionen. Die Buchstaben in Klammern definieren dabei, mit welcher Taste dieses Schlüsselwort belegt ist, was wichtig ist, wenn sich später einmal der Status einer solchen Aktion ändert.

Schreibt man nun in einer Kopfzeile beispielsweise `* TODO`, dann erkennt Org-Mode das Schlüsselwort und hebt es farblich hervor. Unerledigte Aktionen sind mit roten Schlüsselwörtern versehen, erledigte mit grünen.

### Tags

Das nächste Konzept von Org-Mode heißt „Tags“. Damit kann man nochmals für Kopfzeilen bestimmte Begriffe (Tags) vergeben. Später kann man dann seine Org-Mode-Datei nach diesen Tags filtern lassen. Was man hier verwendet bleibt der eigenen Kreativität überlassen. Eine Möglichkeit wäre beispielsweise, den Begriff **PHONE** für alle Aktionen zu verwenden, die mit einem Telefonanruf zu erledigen sind. Dann kann man später recht schnell die Liste der noch zu tätigen Anrufe erstellen. Eine Vorgabe für Tags kann man ebenso wie die Schlüsselwörter am Anfang der Org-Mode-Datei definieren.

```
#TAGS: PHONE(o) URGENT(u) MAIL(m)
```

Will man eine Kopfzeile mit einem Tag versehen, reicht es, den Cursor in dieser Zeile zu positionieren und die Tastenkombination `Strg+C` einzugeben. Sprich, zweimaliges Drücken von `Strg+C` öffnet den Dialog, um die Tags für diese Kopfzeile auszuwählen.

Nach all dieser grauen Theorie wird es nun Zeit, den Org-Mode auch mal am Beispiel [5] einzusetzen. Im Beispiel wird nun Org-Mode verwendet, um die Aufgaben zu organisieren, die notwendig sind um einen Artikel über Org-Mode zu schreiben. Nachdem die Idee zum Artikel in der Diskussion auf der Webseite von [freiesMagazin](#) [6] entstand, war es nur konsequent, die Arbeit mit Org-Mode zu organisieren. Dazu legt man eine neue Org-Mode-Datei an oder erweitert eine bestehende um die neuen Aufgaben.

### Schritt 1 – Kopf frei machen

Um einen Artikel zu schreiben, braucht man einen klaren Kopf. Sinnvoll ist es also, sich vorher einige Stichwörter zum Aufbau des Artikels zu notieren. Am Beispiel des Artikelschreibens bedeutet das, die grobe Struktur des Artikels in Org-Mode zu modellieren und Org-Mode als Stoffsammlung zu benutzen. Wichtig ist in jedem Fall, dass alles, was einem im Kopf herum spukt und was man aber nicht sofort umsetzen kann, aufgeschrieben wird, denn damit schützt man sich vor dem Vergessen von Ideen, die genial waren, aber leider wieder „weg“ sind.

Also frisch ans Werk und einfach mal eine grobe Struktur erstellt. Da Artikelschreiben eine Art Workflow darstellt, erweitert man die Datei um noch ein paar Schlüsselwörter:

```
#+TODO: IDEE(i) ENTWURF(e) FINAL(f) ~
| DONE(d) CANCELLED(c)
```

Alle Abschnitte des Artikels beginnen mit einer Idee, die dann zu einem ersten Entwurf ausformuliert, danach nochmals ein wenig „geschliffen“ und letztlich einer „finalen Kontrolle“ unterzogen wird. Danach ist dieser Teil entweder erledigt (**DONE**) oder man hat irgendwann die Idee doch wieder verworfen (**CANCELLED**).

Diese Zeile mit weiteren Schlüsselwörtern kann im Kopfbereich der Datei eingegeben und mit der Tastenkombination `Strg+C` (dies ist in Org-Mode sozusagen der Universalschlüssel) aktiviert werden. Diese Aktivierung muss nur erfolgen, wenn man das gerade erst getippt hat. Wird





eine Datei mit solchen Definitionen am Dateianfang geöffnet, aktiviert Org-Mode diese natürlich selbst.

Nun schreibt man eine Kopfzeile und etwas Text:

```
* Freies Magazin :LINUX:
- Artikel fuer Freies Magazin
- Autoren-Richtlinien \(PDF\)
:PROPERTIES:
:ARCHIVE: %s_freiesmagazin::
:END:
** PROJ Artikel ueber orgmode schreiben
DEADLINE: <2009-08-26 Mi>
*** ENTWURF Literaturverzeichnis [5/5]...
*** ENTWURF Linkliste [2/2]...
*** ENTWURF Warum orgmode...
*** ENTWURF Grundlegende Org-Mode Prinzipien...
*** IDEE Kopf frei machen...
*** IDEE Taegliche Nutzung...
*** IDEE Review...
```

Hier sind alle Aktivitäten für „Freies Magazin“ unter einer Ebene-1 Kopfzeile zusammengefasst. Dieser Kopfzeile könnte man das Tag „LINUX“ zuweisen. Darunter, hinter dem Bindestrich, verfasst man einen Kommentar, worum es hier geht, und setzt für die Zukunft gleich ein Link auf die Autorenrichtlinien. Ja, Org-Mode kann auch Links. Diesen Link gab es in den Kommentaren auf der Webseite [7], also kopiert man einfach diese Linkadresse, drückt im Org-Mode an der Stelle an der der Link eingefügt werden soll `Strg+C` `Strg+L` und in der untersten Zeile des Emacs-Fensters kann man die Zieladresse ein-

geben. Hier kann mittels `Strg+Y` die Adresse aus der Zwischenablage eingefügt werden. Danach `Return` drücken, den Text zum Link eingeben und fertig. Von jetzt an kann man direkt aus Org-Mode heraus die Autorenrichtlinien von der

Webseite abrufen. Wenn Emacs sauber konfiguriert ist, öffnet das den Browser und dieser das Programm zur PDF-Anzeige.

In den nächsten Zeilen kann man wieder eine Besonderheit von Org-Mode sehen, die „Properties“. Da ist sozusagen eine Schublade, in der besondere Eigenschaften definiert sind. Im Beispiel ist das der

Dateiname, in dem erledigte Aufgaben archiviert werden. `%s` steht für den Dateinamen der Org-Mode-Datei und alles, was später an Aktionen unterhalb der Kopfzeile „Freies Magazin“ archiviert wird, landet in einer Datei, die als Suffix im Namen `_freiesmagazin` enthält.

Die nächste Kopfzeile auf Ebene 2 ist einfach als **PROJ** (Projekt) gekennzeichnet und dient erst mal nur der Gliederung. Redaktionsschluss für die September-Ausgabe von **freiesMagazin** war der 26. August, mit `Strg+C` `Strg+D` kann man dann für dieses Projekt sehr einfach die Deadline definieren.

In der nächsten Ebene ist dann die eigentliche Stoffsammlung zum Artikel vorhanden. Also alles was man berücksichtigen will. Da das Listing während des Artikelschreibens entstand, sind einige Punkte schon von der „Idee“ zum „Entwurf“ gereift. Die drei Punkte am Ende jeder Kopfzeile signalisieren, dass es darunter noch was zu sehen gibt, was aber in der aktuellen Ansicht ausgeblendet ist.

```
*** ENTWURF Linkliste [2/2]
- [X] Orgmode.org
- [X] ThinkingRock GTD
- [2009-08-05 Mi] Linkliste in
OpenOffice.org muss mit
Literaturverzeichnis
kombiniert werden.
```

Hier wurde eine einfache „Checkliste“ angelegt, die helfen soll, nichts zu vergessen. Die Kombination `- [ ]` definiert einen Punkt zum Abhaken. Auf der Kopfzeile definiert `[ / ]` einen Statuszähler der Art „[erledigt/zu erledigen]“. `[%]` würde eine prozentuale Angabe zur Folge haben. Hier sind 2 von 2 Punkten auf der Checkliste abgehakt, daher die Darstellung in Grün. Wäre noch ein Punkt offen, dann würde das auf der Kopfzeile in roter Farbe zu sehen sein. Das Abhaken geschieht in der entsprechenden Zeile mit dem Universalwerkzeug `Strg+C` `Strg+C`.

Weiterhin gibt es eine Notiz mit Datumstempel. Hier ist einfach dokumentiert, warum der Artikel in OpenOffice.org nun Linkliste und Literaturverzeichnis vermischt. Den Datumstempel



kann man einfach mit `Strg+C` `Strg+Shift+1` einfügen. Will man es ganz genau haben und auch die Uhrzeit hier aufnehmen, verwendet man `Strg+U` `Strg+C` `Strg+Shift+1`.

Im Gegensatz zu der Deadline-Definition hat der Datumsstempel eckige Klammern. Der Unterschied wird später in der Agenda-Ansicht klar werden, dort sind Datumsstempel mit spitzen Klammern sichtbar, solche mit eckigen Klammern hingegen unsichtbar.

Global list of TODO items of type: ALL

Available with 'N r': (0)ALL (1)TODO (2)NEXT (3)WAITING (4)SOMEDAY (5)DELEGATED  
(6)PROJ (7)DONE (8)FORWARDED (9)CANCELLED (10)IDEE  
(11)ENTWURF (12)FINAL (13)SKIPPED (14)COPIED

```
privat:    PROJ Artikel ueber Org-Mode schreiben           :LINUX::
privat:    ENTWURF Literaturverzeichnis [5/5]             :LINUX::
privat:    ENTWURF Linkliste [2/2]                        :LINUX::
privat:    ENTWURF Warum orgmode                        :LINUX::
privat:    ENTWURF Grundlegende Org-Mode Prinzipien      :LINUX::
privat:    IDEE Kopf frei machen                         :LINUX::
privat:    IDEE Taegliche Nutzung                        :LINUX::
privat:    IDEE Review                                   :LINUX::
```

Hier hat man also tatsächlich eine Art von Mikro-Blog, in dem man sehr schnell irgendwelche Erkenntnisse, Fortschritte oder auch Fehlschläge dokumentieren kann. Solche Aufzeichnungen können gerade im Beruf sehr nützlich werden, sei es, um ein wenig in der Vergangenheit zu wühlen, wenn man auf „bekannte Probleme“ stößt, sei es, um deutlich dokumentieren zu können, dass man nicht daran schuld ist, dass der Großauftrag für 10 Millionen Euro verloren wur-

de. Das ist auch bekannt unter dem Schlagwort „Cover-Your-Ass“-Security.

Die Org-Mode-Datei ist nun zum „externen Gedächtnis“ geworden, sie enthält alle Ideen und Konzepte und sollte daher auch einem regelmäßigen Backup-Zyklus unterworfen werden. In dem Beispiel waren das jetzt zwar nur die paar Zeilen zum Artikel schreiben, aber im wirklichen Leben kann so eine Org-Mode-Datei schon ein paar hundert oder sogar ein

paar tausend Zeilen haben, abhängig davon, wie intensiv man Org-Mode auch zum Notizen machen nutzt.

### Schritt 2 – Tägliche Nutzung

Jetzt stellt sich für den Nutzer natürlich die Frage, was er mit der langen Liste der zu erledigenden Aufgaben anfangen soll. Irgendwie ist es ja nicht gerade motivierend, wenn man den Mount Everest seiner unerledigten Aufgaben vor sich sieht. Daher heißt der nächste Schritt: „Planen, welche

Aufgaben wann sinnvoll zu erledigen sind“ und Nutzung der „Agenda-Ansicht“.

Bevor die Agenda sinnvoll genutzt werden kann, muss die aktuelle Datei zu der Liste der zu beachtenden Dateien hinzugefügt werden. Dies geschieht über `Strg+C` `AltGr+8`. Danach wird die Agenda-Ansicht über die Tastenkombination `Strg+C` `A` aufgerufen. Darauf hin teilt sich das Emacs-Fenster und im unteren Bereich sieht man ein Auswahlmeneü. Mit der Taste `T` bekommt man wieder eine Aufgabenliste.

Diese Ansicht kann nun auf bestimmte Schlüsselworte eingegrenzt werden. So kann man mit der Tastenkombination `1` `0` `R` die Aufgabenliste auf alle Aufgaben mit dem Schlüsselwort „IDEE“ eingrenzen. Oder, wenn man mal schnell nachsehen will, welche Aufgaben man an andere delegiert hat, reicht in dieser Ansicht `5` `R` und schon erhält man die Liste aller delegierten Aufgaben.

Mit der Taste `Q` wird diese Ansicht wieder geschlossen. So richtig nützlich wird die Agenda-Ansicht aber erst, wenn man sich einen Wochen- oder Tagesplan damit generieren lässt. Aber wie plant man Aufgaben in Org-Mode?

Bislang ist nur die „Deadline“ bekannt, welche mit `Strg+C` `Strg+D` definiert wurde. Wenn man sich nun für einen bestimmten Tag eine Aufgabe vornimmt, dann kann man diesen Plantermin mittels `Strg+C` `Strg+S` („Schedule“) definieren. Im Org-Mode-Kontext ist dieser Termin der Start



der Arbeiten an dieser Teilaufgabe. Mit `Strg` + `C` `A` `A` ruft man einen Wochenplan auf.

Wenn man weiß, dass Aufgaben eine gewisse Zeit in Anspruch nehmen, kann man das auch

wünschen konfigurierbar. In der Praxis kann das so aussehen: Morgens im Büro wird als erstes ein Tagesplan aufgerufen, der nur Aufgaben aus der Datei **work.org** enthält (Privates macht man schließlich daheim) und der auf den aktuellen Tag beschränkt ist. Dann weiß man schon am Morgen, welche Aufgaben geplant sind, welche vielleicht schon überfällig sind oder was im Kalender steht, denn natürlich kann man bei den „Scheduled“-Zeitstempeln auch die Uhrzeit mitangeben.

```
Week-agenda (W33):
Mon    10 August 2009 W33
Die    11 August 2009
Mit    12 August 2009
  privat:  Sched. 4x:  IDEE Kopf frei machen           :LINUX::
  privat:  Scheduled: IDEE Taegliche Nutzung           :LINUX::
  privat:  In 14 d.:  PROJ Artikel ueber Org-Mode schreiben :LINUX::
Don    13 August 2009
Fre    14 August 2009
Sam    15 August 2009
  privat:  Scheduled: IDEE Review                       :LINUX::
Son    16 August 2009
```

Eine andere selbst definierte Agenda-Ansicht zeigt nur offene Aufgaben, die noch keinen Plantermin haben. Somit bekommt man schnell Übersicht, was noch zu planen ist. Das geht auch direkt aus der Agenda-Ansicht mit Hilfe des Hotkeys `Strg` + `C` `Strg` + `S`.

Wie man sieht, ist das Listing am 12. August entstanden, aus der Sicht also als „Heute“ zu bezeichnen, und wird kursiv und fett formatiert. Man kann drei Einträge sehen (Hinweis: Diese Agenda-Ansicht wurde auf die Kopfzeile „Freies Magazin“ und die darunter liegende Hierarchie eingeschränkt). Das **privat:** am Zeilenanfang ist ein Hinweis auf die Quelle, in diesem Fall die Datei **privat.org**. Die erste Zeile zeigt **Sched. 4x** was letztlich bedeutet, dass vor vier Tagen die Arbeit an „Kopf frei machen“ begonnen hat. Für den heutigen Termin steht ein **Scheduled:** in grün und signalisiert, dass die Arbeit am Kapitel „Tägliche Nutzung“ geplant ist. Am 15. August steht dann ein Kapitel über „Review“ auf dem Plan. Und schließlich erinnert Org-Mode daran, dass in 14 Tagen die Deadline für das Projekt „Artikel über Org-Mode schreiben“ ansteht.

in Org-Mode definieren. Wenn man für das Projekt einen Zeitbereich definiert wie „<2009-08-07 Fr>-<2009-08-22 Sa>“, dann wird in der Agenda für jeden dieser Tage ein Eintrag eingeblendet.

Mit Org-Mode kann man auch hervorragend wiederkehrende Aufgaben planen. Als Beispiel sei

```
Mit    12 August 2009
  privat:  Sched. 4x:  IDEE Kopf frei machen           :LINUX::
  privat:  Scheduled: IDEE Taegliche Nutzung           :LINUX::
  privat:  (6/16):  PROJ Artikel ueber Org-Mode schreiben :LINUX::
  privat:  In 14 d.:  PROJ Artikel ueber Org-Mode schreiben :LINUX::
Don    13 August 2009
```

Nun gibt es am heutigen 12. August einen Eintrag mit dem Hinweis **(6/16)**, der besagt, dass nun 6 von 16 veranschlagten Tagen für das Projekt vergangen sind.

hier die Planung einer regelmäßigen Reinigung der Geschirrspülmaschine genannt (Ja, die Dinge müssen auch hin und wieder gereinigt werden).

Die Agenda-Ansicht von Org-Mode ist extrem flexibel einsetzbar und beliebig nach Anwender-

Die nächste Reinigung ist also für Samstag, den 15. August geplant. Das unscheinbare **+12w**



```

** TODO Spielmaschine reinigen :RAINER:
SCHEDULED: <2009-08-15 Sa +12w>
- Siebe reinigen
- Maschinenpflege auf hoechster Stufe durchlaufen lassen

```

hat es in sich. Wenn am Samstag die Reinigung erfolgt ist und der Status mit `[Strg]+[C]` `[Strg]+[T]` auf **DONE** gesetzt wird, dann erhält diese Aufgabe – falls gewünscht – einen Logbuch-Eintrag für die Erledigung. Die Aufgabe verharrt aber im Status **TODO** und das Plandatum schaltet exakt um die 12 Wochen weiter. Das bedeutet, dass Org-Mode in exakt 12 Wochen wieder daran erinnern wird, dass eine Spülmaschinenreinigung ansteht.

Diese Funktion ist extrem hilfreich, wenn es um Dinge geht, die man gerne „vergisst“, also angefangen von Haushaltsarbeiten bis hin zum Zahnarzt, den man ja auch hin und wieder besuchen soll.

Man kann im Org-Mode-Buffer auch auf Zeitstempel klicken und bekommt dann einen Tagesplan für den betreffenden Tag (oder einen Mehrtagesplan wenn man auf eine Bereichsdefinition klickt). Ein Klick auf den Scheduled-Zeitstempel in obigem Beispiel bringt dann eine Agenda-Ansicht zum Vorschein. Also die bekannten

```

Day-agenda (W33):
Sam      15 August 2009
privat:   Scheduled: TODO Spielmaschine reinigen :RAINER:
privat:   Scheduled: IDEE Review :LINUX::
privat:   (9/16): PROJ Artikel ueber Org-Mode schreiben :LINUX::

```

Aufgaben zum Schreiben dieses Artikels und die neu hinzugekommene Spülmaschinenreinigung. Alle Möglichkeiten der Agenda-Ansicht hier aufzählen zu wollen, würde wohl ein Buch füllen, daher sei hier nochmals auf die sehr gute Dokumentation von Org-Mode verwiesen.

### Schritt 3 – Reviews

Eine der goldenen Regeln im Zeitmanagement ist: „Zeitmanagement ohne regelmäßige Reviews ist kein Zeitmanagement“. Immerhin nutzt man all das Zeitmanagement ja, um sich Aufgaben und Ziele zu setzen und da sollte man hin und wieder schon auch mal eine „Positionsbestimmung“ machen, um zu sehen, ob man noch auf Kurs ist oder nicht.

Diese Reviews plant man natürlich auch wieder mit Org-Mode. So kann man beispielsweise jeden Freitag eine Aufgabe **TODO** mit dem Titel „Weekly review“ in der Aktionsliste haben, welche dann automatisch bei Erledigung auf den nächsten Freitag vorgetragen wird.

Als Hilfsmittel dient eine Checkliste, die unter dem Dateinamen **weeklyreview.org**

abgespeichert wurde und die mit einem einfachen Link per Mausklick geöffnet werden kann. Diese Checkliste enthält dann alles, was regelmäßig am Freitag kontrolliert und eventuell erledigt werden sollte.

Dazu zählen beispielweise:

- Können Aktionen mit dem Schlüsselwort **SOMEDAY** in den Status **TODO** oder gar **NEXT** wechseln? Sprich, wird es langsam Zeit, das was man sich für „irgendwann“ vorgenommen hat, jetzt mal konkret anzugehen? Die Vorgehensweise: Agenda-Ansicht über alle Aufgaben, dann limitieren auf Schlüsselwort **SOMEDAY**.
- Gibt es irgendwelche Erkenntnisse zu **DELEGATED**-Aktionen? Hat sich derjenige, der mit der Erledigung beauftragt wurde, schon gemeldet? Ist die Aktion abgeschlossen? Wie oben, jetzt aber Agenda-Ansicht auf Schlüsselwort **DELEGATED** limitieren.
- Gibt es Aufgaben ohne „Plantermin“? Können einige dieser Aufgaben für die nächste Woche geplant werden? Hierzu kann man eine benutzerspezifische Agenda-Ansicht mit Einschränkung auf Aufgaben ohne Plantermin erstellen (siehe Org-Mode-Dokumentation [8]).
- Brainstorming und Kopf wieder frei machen, neue Ideen in Org-Mode eintragen.
- Überprüfen aller als **URGENT** markierten Aufgaben. Hierzu benötigt man keine Agenda-Ansicht, sondern einen „sparse tree“, in dem die so gefilterten Aufgaben hervorgehoben



sind. Solche eingeschränkten Listen erzeugt man mit der Tastenkombination `Strg + C` `Shift + 7` zusammen mit einer Taste für das, nach dem man suchen will. Also beispielsweise `Shift + T` (wie Tags) und dann als Markierung **URGENT**. Das Vervollständigen mit der `Tab`-Taste funktioniert hier, es reicht also, die Eingabe von `U` `R` `Tab` um das Schlüsselwort **URGENT** zu wählen. Die Hervorhebung der so gefundenen Überschriften kann man später wieder mit `Strg + C` `Strg + C` aufheben.

Natürlich kann man diese Checkliste beliebig erweitern. Wer beispielsweise seine Einnahmen und Ausgaben in einem Programm wie GNU-Cash verwaltet, könnte hier eine Aktion „Kassensurz“ machen, um zu sehen, ob er wirklich noch soviel Bargeld hat, wie GNUCash behauptet. Kontrolle ist ja immer besser ...

Eine Arbeit bei einer solchen Durchsicht kann auch darin bestehen, erledigte Aufgaben in die entsprechenden Archivdateien zu übertragen, damit die Aufgabenliste nicht mit längst Erledigtem zugemüllt ist. Das Übertragen einer Aufgabe und aller in der Hierarchie darunter liegenden Aufgaben geschieht mit den Tasten `Strg + C` `Strg + X` `Strg + S`, also ähnlich wie das Speichern in Emacs (`Strg + X` `Strg + S`), nur hier wieder mit vorgestelltem `Strg + C`. Damit schiebt man den ganzen Baum in die Archivdatei, die man später auch problemlos wieder in Org-Mode öffnen kann. Es ist dann kein Problem zu sehen, was man damals alles gemacht hat.

## Fazit

Mit dem Org-Mode von Emacs steht man dem alltäglichen Wahnsinn gut gewappnet gegenüber. Zwar mag die Bedienung zunächst ein wenig gewöhnungsbedürftig erscheinen, jedoch – und wie eingangs schon erwähnt – gehen einem schon nach kurzer Zeit die zentralen Prinzipien und Tastaturkürzel in Fleisch und Blut über, sodass der der „Ordnung im Chaos“ nichts mehr im Wege steht.

*Verwendete Printmedien:*

1. van Herck, Paul: „Framstag Sam“, Heyne, 1981
2. Ende, Michael: „Momo“, Thienemann Verlag, 1973
3. Seiwert, Lothar: „Wenn Du es eilig hast, gehe langsam“, Campus Verlag, 2005
4. Allen, David: „Getting Things Done“, Penguin Books, 2001

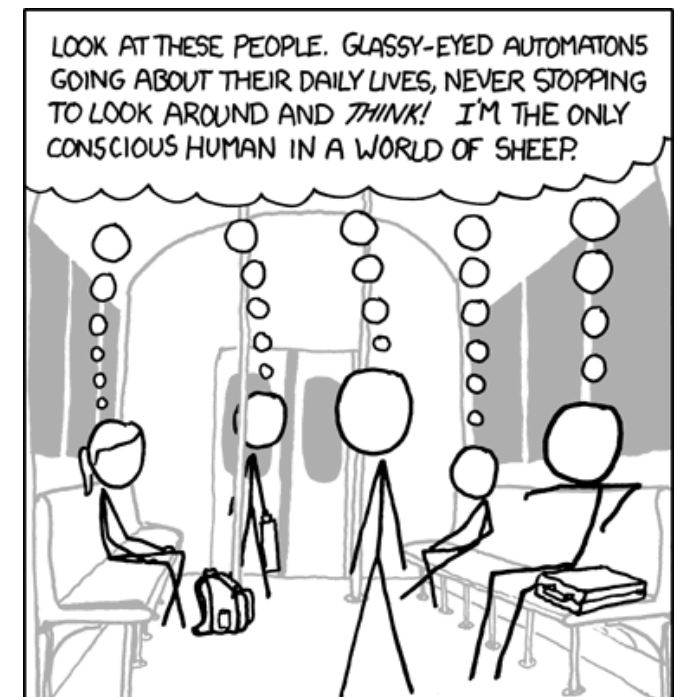
## LINKS

- [1] <http://sourceforge.net/projects/thinkingrock/> 
- [2] <http://www.orgmode.org/> 
- [3] <http://www.gnu.org/software/emacs/> 
- [4] <http://orgmode.org/index.html#sec-3> 
- [5] <http://www.freiesmagazin.de/mobil/2009-10-listings/privat.org>
- [6] <http://freiesmagazin.de/20090801-freiesmagazin-sucht-unterstuetzung#comment-148>
- [7] <http://freiesmagazin.de/20090801-freiesmagazin-sucht-unterstuetzung#comment-149>
- [8] <http://orgmode.org/index.html#sec-4> 

## Autoreninformation

**Rainer König** ist seit dem letzten Jahrtausend mit Linux unterwegs und arbeitet bei einem PC-Hersteller am Thema Linux-Hardware-Kompatibilität für PCs.

*Diesen Artikel kommentieren* 



„Sheeple“ © by Randall Munroe  
(CC-BY-NC-2.5), <http://xkcd.com/610>

## Kurztipp: Suchen und Finden mit Vim von Steve Klicek

**M**an trifft ihn auf vielen unixoiden Systemen: *Vim*. Für den Eingeweihten mag er ein geschätzter, alter Bekannter sein, für Nichteingeweihte erscheint er eher als sperriger Zeitgenosse. Gut, wenn man ihn zu nehmen weiß.

Der sperrige Zeitgenosse ist nämlich auch dann noch da, wenn von grafischen Oberflächen nichts mehr zu sehen, was – Murphys Gesetz [1] folgend – genau der typische Zeitpunkt ist, um Änderungen an Konfigurationsdateien vornehmen zu müssen. Gut, wenn man weiß, wie man *Vim* zum Suchen, Finden und Ersetzen überredet.

### Kurze Beschreibung

*Vim* steht als Abkürzung für „Vi IMproved“, also für einen „verbesserten Vi“ (frei übersetzt). Hierbei handelt es sich, genau wie bei Vi, um einen Texteditor für die Konsole. Er ist nicht nur unter Linux verfügbar, sondern auch unter vielen anderen Betriebssystemen wie Windows, MacOS, AmigaOS etc. Das Programm Vi selbst ist eine kommerzielle Anwendung und daher nicht für Linux verfügbar. Das macht aber nichts, denn hier steht ja sein Open-Source-Pendant *Vim* zur Verfügung, das zu Vi kompatibel ist und darüber hinaus zahlreiche Verbesserungen und Erweiterungen bietet.

Wikipedia meint hierzu: „*Vim funktioniert wie der vi-Editor. Die Bedienung erfolgt dann ausschließlich über die Tastatur. Das ist historisch bedingt*

*durch die zu Ur-vi-Zeiten Mitte der 1970er Jahre üblichen Bildschirmterminals mit einer langsamen, seriellen Verbindung zum Hauptrechner. Aus der Limitierung damaliger Terminals resultiert sein modales Konzept, das sich jedoch auch heute noch gut bewährt. Vim ist nahezu vollständig abwärtskompatibel zu vi, hat jedoch eine Vielzahl an Verbesserungen, und mittlerweile gibt es eine moderne grafische Benutzerschnittstelle mit Menüs (GVim) sowie eine vereinfachte Version für Einsteiger (eVim). Auch die Benutzung der Maus wird unterstützt [2].“*

*Vim-Neulingen* wird zuvor der Artikel „Vim – Ein Schnelleinstieg“ aus [freiesMagazin](#) 08/2008 [3] empfohlen.

### Hilfestellung

Unter Linux ist die erste Anlaufstelle für eine ausführliche Hilfe zu Befehlen das man-Kommando. Durch die Eingabe von

```
$ man vim
```

gelangt man zur „Manpage“ und erhält dort eine ausführliche, allgemeine Hilfestellung. (Man verlässt die Manpage durch Drücken von **Q**.)

Für das im Folgenden beschriebene Thema „Suchen und Finden“ gibt es in der Online-Hilfe nützliche Informationen. Durch die Kommandos **:help search** und **:help substitute** in *Vim* gelangt man zu vielen hilfreichen Tipps und Beispielen.

Eine andere Hilfe ist das Buch „Linux“ des bekannten Autors Michael Kofler. Der „Kofler“ gilt als das Nachschlagewerk für alle Linux-Anwender.

### Wer sucht, der findet

#### Vorwärts suchen

Um in einem geöffneten Text zu suchen, gibt man im Standardmodus ein / ein, d.h. man drückt **Shift** + **7**. Anschließend fügt man den Suchbegriff ein: Also z. B. „Wort“. Abschließend bestätigt man den Suchauftrag über die Eingabetaste **Enter**.

Der Cursor bewegt sich nun an die erste Stelle im Text, an der der Begriff „Wort“ gefunden wird. Die Taste **N** wiederholt die Suche, der Cursor springt zur nächsten Fundstelle.

#### Rückwärts suchen

Mit **Shift** + **N** wird rückwärts im Text gesucht, also ab der aktuellen Cursorposition in Richtung Textanfang.

Man kann mit der Eingabe von **?**, also **Shift** + **B**, und einem nachfolgenden „Wort“ dem Editor auch mitteilen, dass er von vornherein rückwärts suchen soll.

#### Suchmuster

In den Suchbegriff eingebettet kann man nach verschiedenen Kriterien suchen und zwar (nach Betreten des Suchmodus mit **Shift** + **7**):

- W** sucht nach W und einem beliebigen Zeichen
- ^W** sucht nach W am Zeilenanfang
- W\$** sucht nach W am Zeilenende
- W[a-e]** sucht W und einem Zeichen von a bis e

### Groß- und Kleinschreibung umgehen

*Vim* unterscheidet bei der Suche immer zwischen Groß- und Kleinschreibung. Wenn man dies nicht möchte, kann man das mit einem angehängten `\c` umgehen. Das gilt anschließend nur für diese eine Suche. Damit es für alle weiteren Suchen gilt, führt man das Kommando `:set ignorecase` aus. Die Eingabe `screen \c` sucht also nach „screen“ und nach „Screen“.

### Inkrementelle Suche

Das Kommando `:set incsearch` aktiviert die inkrementelle Suche. Dies ist sinnvoll, wenn man nach dem Beginn eines Wortes sucht. Hier wird der Cursor schon während der Eingabe des Suchbegriffes an den ersten Übereinstimmungsort bewegt. Die `Enter`-Taste beendet diesen Spuk, mit der `Escape`-Taste wird die Suche abgebrochen. Nach der Suche bleiben die Übereinstimmungen im Text markiert, bis man entweder eine neue Suche beginnt oder das Kommando `:nohlsearch` eingibt.

### Anmerkungen zum `:set`-Kommando

Die Optionen, die mit `:set` gesetzt werden, gelten nur für die aktuelle *Vim*-Sitzung. Sollen hingegen die Einstellungen permanent gesetzt bzw. nicht gesetzt sein, so muss dies in `/etc/vimrc`

(gültig für das ganze System) oder `~/.vimrc` (gültig nur für den aktuellen Benutzer) erfolgen.

### Suchen und Ersetzen

Hier nun einige Funktionen, wie man Text suchen und ersetzen kann. Das Kommando, um zum Beispiel „abc“ durch „efg“ zu ersetzen, lautet `:%s/abc/efg`.

Um das Suchen und Ersetzen noch besser zu kontrollieren, bestehen darüber hinaus folgende Möglichkeiten, die durch das Anfügen an das obige Kommando erreicht werden:

- `/g` ohne Rückfrage werden alle Vorkommen von „abc“ durch „efg“ ersetzt
- `/gc` mit Rückfrage werden alle Vorkommen von „abc“ durch „efg“ ersetzt
- `/gi` ersetzt alle Vorkommen von „abc“ durch „efg“ ohne Groß- und Kleinschreibung zu berücksichtigen

**Hinweis:** Der ganze Ausdruck muss aneinander geschrieben werden: z. B. `:%s/abc/efg/gi`.

### Abspeichern und Verlassen

Ganz am Schluss der Bearbeitung sollte man nicht vergessen, die Datei abzuspeichern. Dazu einfach das Kommando `:w` absetzen. Dann verlässt man die Datei mit einem `:q`. Ein `:wq` erledigt dies in einem Schritt.

### Fazit

Wer erst einmal den *Vim*-Editor in seinen Grundfunktionen beherrscht, der kann zum Beispiel viel schneller Änderungen an Konfigurationsdateien

vornehmen, als so manch anderer Benutzer, der mit einem grafischen Editor unterwegs ist.

Voraussetzung dafür ist aber das ständige und konsequente Benutzen des Editors. Am Besten macht man sich nebenher auf einem Stück Papier eine Übersicht der gängigsten Befehle und legt das neben die Tastatur. So gewöhnt man sich schneller an den Umgang und wird immer vertrauter mit diesem Urgestein der Editoren.

*Verwendete Printmedien:*

1. „Linux“ (8. überarbeitete und erweiterte Auflage), Michael Kofler, Addison Wesley, ISBN: 978-3-8273-2478-8

### LINKS

- [1] [http://de.wikipedia.org/wiki/Murphys\\_Gesetz](http://de.wikipedia.org/wiki/Murphys_Gesetz)
- [2] <http://de.wikipedia.org/wiki/Vim>
- [3] <http://www.freiesmagazin.de/freiesMagazin-2008-08>

### Autoreninformation

**Steve Klicek** benutzt Linux seit Suse Linux 6.2, wobei z. Z. openSUSE seine erste Wahl ist. Den Editor *Vim* schätzt er als zuverlässigen Editor für seinen täglichen Umgang mit Linux.

*Diesen Artikel kommentieren* 

## grep – Eine kleine Einführung von Philipp Walther

Unter praktisch jedem Unix, Linux und deren Derivaten ist das Kommandozeilenprogramm **grep** vorhanden. Es dient der Textsuche und Filterung von Dateien, dabei geht es zeilenweise vor. Es können einfache Wörter gesucht oder aber auch mit Hilfe von regulären Ausdrücken [1] komplexe Suchen realisiert werden.

### Varianten

Neben **grep** existieren noch weitere Varianten wie **egrep** oder **fgrep**. Diese verhalten sich wie **grep** und können auch mit der entsprechenden Option (**-E** bzw. **-F**) aufgerufen werden. Beim Durchsuchen von komprimierten Dateien helfen **zgrep** und **bzgrep**.

### Aufruf von grep

Der Aufruf erfolgt nach dem Muster

```
$ grep [Optionen] Suchmuster [↵
Datei]
```

Wird keine Datei angegeben, verwendet **grep** die Standardeingabe.

### Optionen

Die einzelnen Optionen von **grep** sind vielseitig. So kann **grep**

- zählen: **--count**,
- Groß- und Kleinschreibung ignorieren: **--ignore-case**,
- Zeilennummern zurückgeben: **--line-number**,

- das Suchmuster umdrehen: **--invert-match**
- und vieles mehr.

Hierzu lohnt es sich, sich mit der Manpage von **grep** auseinanderzusetzen und zu experimentieren.

### Anwendung von grep

Verwendung findet **grep** auch in vielen Skripten, die beim Start des Betriebssystems ausgeführt werden. Die praktische Anwendung im Alltag besteht in der Selektion relevanter Teile aus Programmausgaben oder in der Filterung von Log-Dateien.

### Zählen mit grep

Für Skripte ist es oft wichtig, Zustände des Systems zu wissen. Beispielsweise kann man mit **grep** herausfinden, ob vom Kernel ein bestimmtes Dateisystem unterstützt wird. So liefert die folgende Zeile eine Zahl größer 0, wenn der gerade geladene Kernel dies unterstützt:

```
$ grep --count ext2 /proc/↵
filesystems
```

### Zeilennummer

Bei der Suche in größeren Skripten oder in Quellcode-Dateien ist es von Interesse, zu wissen, an welcher Stelle das gesuchte Wort vorkommt. **grep** ist in der Lage, die Zeilennummer dazu anzugeben, in der das gesuchte Wort steht. So würde bei folgender Eingabe

```
$ grep --line-number audio /etc↵
/group
```

die Ausgabe **22:audio:x:29:anon** bedeuten, dass das Wort **audio** in der Zeile 22 vorkommt.

### Dateiname

In Kombination von Zeilennummer und der Ausgabe des Dateinamens lassen sich rekursiv auch ganze Verzeichnisstrukturen auf das gesuchte Ergebnis hin durchsuchen.

```
$ grep --with-filename --line-↵
number --recursive root /etc
```

liefert jedes Vorkommen der Zeichenkette **root** unter Angabe des Dateinamens und der Zeilennummer.

### Abgestürzter MPlayer

Das Beenden eines hängenden Prozesses erledigt man mit **kill**. **kill** möchte als Parameter nicht den Namen des Programmes, das es beenden soll, wissen, sondern die dazugehörige Prozessnummer, die das laufende Programm eindeutig identifiziert. Der Befehl **ps** liefert eine Auflistung aller laufenden Prozesse. Diese Auflistung ist in der Regel sehr unübersichtlich:

```
$ ps aux
```

Hier kommt **grep** ins Spiel:

```
$ ps aux | grep mplayer
```



**ps aux** gibt alle Prozesse aus. Die Ausgabe wird mit dem Pipe-Zeichen an **grep** übergeben und auf die gesuchte Zeichenfolge gefiltert.

Die Ausgabe könnte in etwa so lauten:

```
username      4057  2.3  2.8  67588 25504 tty1      S   17:06   1:54~
  gmplayer /home/username/Musik/xyz.mp4
username      5552  0.0  0.0   3204   768 pts/2     R+  18:27   0:00~
  grep mplayer
```

Als Ergebnis wird nun auch **grep** selbst geliefert, da es mit der gesuchten Zeichenfolge als Parameter aufgerufen wurde. Die gesuchte Information findet sich aber in diesem Fall in der ersten Zeile. Nun weiß man, dass der MPlayer hier als **gmplayer** die Prozessnummer 4057 trägt. Jetzt kann dieser mit

```
$ kill 4057
```

beendet werden.

### Probleme mit dem X-Server

In vielen Foren trifft man Menschen, die **cat** in Kombination mit **grep** einsetzen, um in einer Datei eine spezifische Zeile zu finden. Dies ist häufig ein unnützer Umweg. So wird oft die Filterung der **/var/log/Xorg.0.log** mit Hilfe der Befehlszeilenfolge

```
$ cat /var/log/Xorg.0.log | ~
grep WW
```

vorgenommen. Dabei wird die Ausgabe von **cat** der Datei **Xorg.0.log** in **grep** umgeleitet und von

diesen Zeilen mit dem Vorkommen von **WW** auf der Standardausgabe ausgegeben. Was man erreichen wollte, ist die Auflistung von Warnungen. Dies ist bei alleiniger Verwendung von **grep** jedoch einfacher und treffsicherer zu erreichen.

```
$ grep -e "^(WW)" /var/log/Xorg~
.0.log
```

liefert unter Verwendung von regulären Ausdrücken alle Zeilen der Datei, die mit **(WW)** beginnen. Diese lassen sich auch kombinieren, sodass in einem Rutsch Fehlermeldungen und Warnungen ausgegeben werden:

```
$ grep -e "^(WW)" -e "^(EE)" /~
var/log/Xorg.0.logo
```

### grep und history

Nützlich machen kann sich **grep** auch in Kombination mit **history**, wenn lange Befehlszeilen mehrmals eingegeben werden müssen. So liefert

```
$ history | grep grep
```

die bisherigen **grep**-Aufrufe des jeweiligen Benutzers. Darunter der Aufruf aus dem vorherigen Beispiel.

```
518 grep -e "^(WW)" -e "^(EE)"~
/var/log/Xorg.0.log
```

Man kann sich die nochmalige Eingabe dieser längeren Befehlszeichenfolgen ersparen, indem man diese mit

```
$ !518
```

ein weiteres Mal ausführt.

### Weitere Informationen

Weitere Informationen sind auf der Homepage des GNU-Projektes [2] zu finden. Ausführliche Informationen zur Verwendung von **grep** sind im Ubuntuusers-Wiki [3] und in der zu **grep** gehörenden Manpage nachzulesen.

### LINKS

- [1] [http://de.wikipedia.org/wiki/Regul%C3%A4rer\\_Ausdruck](http://de.wikipedia.org/wiki/Regul%C3%A4rer_Ausdruck)
- [2] <http://www.gnu.org/software/grep/> 
- [3] <http://wiki.ubuntuusers.de/Shell/grep>

### Autoreninformation

**Philipp Walther** hat seine ersten Linux-Erfahrungen mit SUSE-Linux gemacht. Es folgten einige Jahre der Linux-Abstinenz. Nach einer Zeit des Experimentierens mit verschiedenen Distributionen ist er schließlich bei Debian GNU/Linux angekommen.

[Diesen Artikel kommentieren](#) 

## Von der Schaltung zur fertigen Platine mit EAGLE von Marcel Jakobs

**M**it dem von der Firma CadSoft entwickelten Programm *EAGLE* [1] kann man Schaltungen und Platinenlayouts entwerfen. *EAGLE* ist ein Akronym, das „Einfach Anzuwendender Grafischer Layout-Editor“ bedeutet. Trotz dieses deutschen Akronyms ist die Software auf Englisch. Die kostenlose Version ist auf eine halbe Europlatine (100 mm × 80 mm) und zwei Schichten beim Platinenlayout beschränkt, ansonsten kann man fast alles damit machen, was auch mit der Vollversion geht.

Warum hier keine freie Alternative vorgestellt wird, hat den einfachen Grund: *EAGLE* ist gerade im Bereich Mikrocontroller ein Quasi-Standard. Wenn jemand eine Schaltung oder ein Platinenlayout online stellt, wird dieses mit Sicherheit für *EAGLE* sein. Wenn es irgendwo eine Bauteilbibliothek für ein seltenes Teil gibt, ist sie sehr wahrscheinlich für *EAGLE*.

### Installation und Start

Unter einigen Distributionen – wie z. B. Ubuntu – kann *EAGLE* über die Paketverwaltung installiert werden. Ansonsten kann man die Binärdateien des Programms auch von der offiziellen Homepage [2] beziehen.

Ist das Programm installiert und gestartet, kann man über „File » New“ ein neues Projekt beginnen. Ein Projekt kann dabei aus mehreren Schaltungen und Platinen bestehen.

### Icons

In diesem Artikel wird reger Gebrauch der einzelnen Werkzeuge und Schaltflächen des Programms gemacht. Daher hier eine Übersicht über die verwendeten Symbole:

### Schaltung entwerfen

Als Erstes erstellt man die Schaltung („Schematic“), in der die einzelnen Bauteile definiert und dabei die Verbindungen zwischen ihnen festgelegt werden.

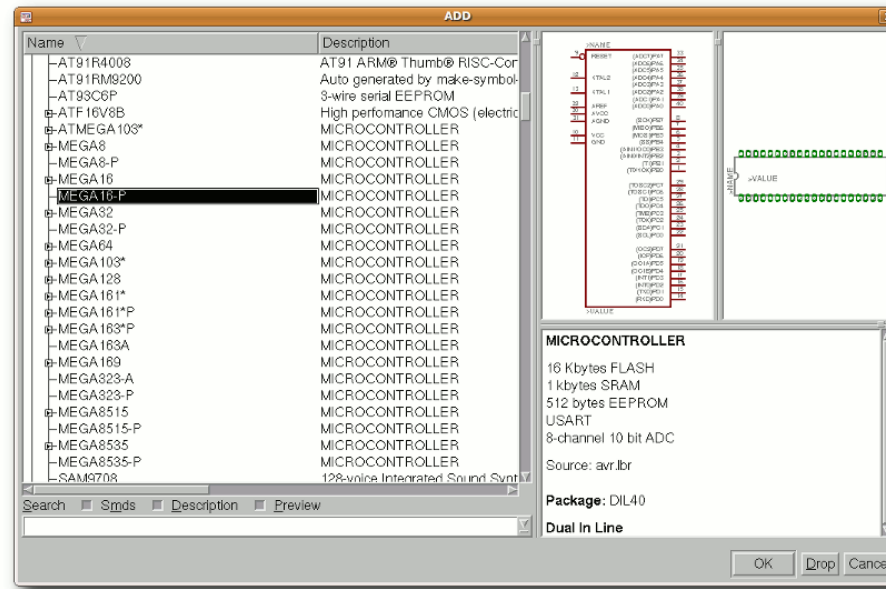
EAGLE-Icons		
„Add“		Öffnet einen Auswahldialog, um Bauteile hinzuzufügen.
„Auto“		Startet den Autorouter.
„Board“		Wechselt zur Boardansicht.
„Bus“		Werkzeug zum Generieren von Bussen.
„Change“		Mit diesem Tool können Eigenschaften von Bauteilen und anderen Elementen geändert werden.
„Copy“		Kopiert ein Bauteil.
„Celete“		Löscht ein Bauteil oder ein Netz.
„Display“		Öffnet einen Dialog, in dem man auswählen kann, welche Elemente wie angezeigt werden sollen.
„Drc“		Startet den Design Rule Check.
„Erc“		Startet den Electrical Rule Check.
„Go“		Wendet das ausgewählte Werkzeug auf alle Elemente an.
„hole“		Dient zum Vormerken von Bohrlöchern.
„Info“		Zeigt Informationen über Bauteile und Netze an.
„Move“		Werkzeug zum Verschieben von Bauteilen.
„Name“		Werkzeug zum Benennen von Bauteilen und Netzen.
„Net“		Werkzeug zum Zeichnen von Netzen.
„Polygon“		Werkzeug zum Zeichnen von Polygonen.
„Ratsnest“		Dieses Werkzeug optimiert die Verbindungen im Boardlayout.
„Ripup“		Werkzeug zum Auftrennen von Leitungen.
„Route“		Werkzeug zum Verlegen von Leitungen.
„Show“		Zeigt den Verlauf eines Netzes an.
„Text“		Werkzeug zum Setzen von Text auf den Schaltplan oder die Platine.
„Ulp“		Öffnet einen Dialog zum Starten von Benutzerskripten.
„Value“		Werkzeug, um Werte von Bauteilen einzustellen (z. B. den Wert eines Widerstands).
„Via“		Werkzeug zum Erstellen von Durchkontaktierungen.

## Bauteile platzieren

Um Bauteile auszuwählen, klickt man auf das Symbol „Add“ in der Werkzeugleiste, woraufhin sich ein Auswahldialog öffnet, in dem sehr viele Bauteile zur Verfügung stehen. Das richtige Bauteil zu finden, ist am Anfang nicht immer leicht. Es ist jedoch wichtig, die richtigen Bauteile zu verwenden, damit später das Platinenlayout stimmt und die Bauteile auch darauf passen. Daher hier einige Tipps:

- Widerstände, Kondensatoren und Induktoren befinden sich in der Gruppe RCL (Resistor = Widerstand, Capacitor = Kondensator und Inductor = Induktor). Darin sollte man die EU-Gruppen wählen.
- VCC und GND können über die Suche recht leicht gefunden werden.
- Atmel-Mikrocontroller sind im Paket **atmel** aufgelistet.
- Stift-/Buchsenleisten findet man in den Paketen **con-1sta** (female, also Buchsen) und **con-1stb** (male, also Stecker).
- Wannenstecker befinden sich in **con-harting-ml**.
- LEDs gibt es im Paket **led**.

Ansonsten empfiehlt es sich, die Pakete nach dem Hersteller oder der Artikelbezeichnung zu durchsuchen. Oft befinden sich auf der Herstellerseite oder auf anderen Internetseiten Bibliotheken mit Bauteildefinitionen zum Herunterladen. Eine ausgiebige Suche im Internet lohnt sich da oftmals.



Der Auswahldialog für Bauteile. 🔍

Es gibt auch die Möglichkeit, Bauteile selbst zu definieren. Dies wird hier jedoch nicht behandelt, da die gängigen Bauteile normalerweise bereits verfügbar sind.

Ist das Bauteil ausgewählt, kann es mit der rechten Maustaste um je 90° gedreht und mit einem Linksklick platziert werden. Man kann den meisten Bauteilen auch noch Namen geben sowie die Werte von Widerständen, Kondensatoren etc. ändern. Dafür gibt es die beiden Icons „Name“ und „Value“.

Möchte man ein bereits platziertes Bauteil verschieben, so kann man dazu das Werkzeug „Move“ nutzen. Ist „Move“ ausgewählt, kann man mit der linken Maustaste auf ein Bauteil (oder Netz)

klicken. Wenn mehrere Bauteile oder Netze in der Nähe sind, wird eins rot markiert (Netze werden hellgrün markiert). Sind keine anderen Bauteile in der Nähe, wird das Bauteil direkt ausgewählt, sodass man es mit der Maus verschieben kann. Wenn das markierte Bauteil das Richtige ist, kann man erneut mit der linken Maustaste drücken und es ist ausgewählt. Mit der rechten Maustaste wird ein anderes Bauteil (in der Nähe) rot markiert. So kann man zwischen den markierten Bauteilen umherschalten

bis das richtige markiert ist und dieses dann auswählen. Hat man das Bauteil an seine neue Position verschoben, kann man es noch mit der rechten Maustaste pro Klick um 90° drehen und dann mit der linken Maustaste ablegen.

## Bauteile verbinden

Um die Bauteile zu verbinden, nutzt man am besten das Werkzeug „Net“. Damit klickt man auf einen Anschluss eines Bauteils und kann nun das Netz zum nächsten Bauteil ziehen, wo es enden soll.

Wenn man viele Leitungen von einem Bauteil zu einem anderen ziehen muss, sind Busse sehr gut geeignet. Dazu nimmt man sich das Bus-Werkzeug und zieht damit eine Linie über die



Strecke, auf der die Leitungen gleich verlaufen sollen. Als Name gibt man dann einen Namen gefolgt von einer Art Array an, wie z. B. „PortA[0..7]“, um acht Leitungen als Bus zusammen zu fassen. Nun kann man die einzelnen Leitungen des Busses mit Pins von Bauteilen verbinden. Dazu wählt man das Werkzeug „Net“ aus und klickt damit auf den Bus. Es erscheint ein Auswahlmenü mit allen Netzen, die der Bus besitzt (in diesem Beispiel „PortA0“ bis „PortA7“). Dann kann man das Netz wie gewohnt mit dem Pin des Bauteils verbinden. Natürlich geht es auch umgekehrt, und man kann ein Netz von einem Bauteil aus mit dem Bus verbinden. In diesem Fall erscheint beim Klicken auf den Bus auch hier das Auswahlmenü.

Manchmal kommt es vor, dass man zwei Netze miteinander verbindet (z. B. wenn man einen Widerstand mit GND verbindet). In diesem Fall fragt *EAGLE* nach, ob man das wirklich tun möchte und welchen Namen das Gesamtnetz haben soll. Im obigen Beispiel würde man GND wählen.

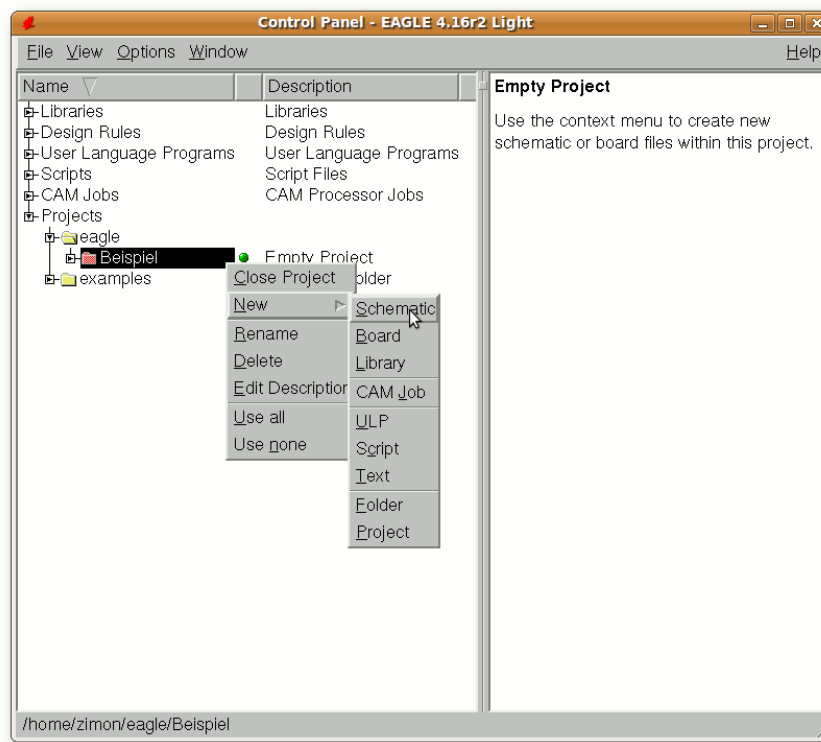
### Sonstige Techniken

Mit dem Werkzeug „Delete“ lassen sich Bauteile und Netze wieder löschen.

Das Werkzeug „Text“ kann genutzt werden, um Anmerkungen zu machen oder verschiedene Bereiche zu kennzeichnen. Dazu

sind die anderen „Malwerkzeuge“ auch recht praktisch. Um einen Text auf die Platine zu schreiben, muss man das „Text“-Werkzeug in der Boardansicht benutzen, dazu später mehr.

Beim Ziehen von Netzen und Bussen kann ein Richtungswechsel der Leiterbahn durch einen Linksklick eingeleitet werden. Um eine Leitung enden zu lassen, ohne an ein Bauteil angeschlossen zu sein, klickt man zweimal an die gleiche Stelle. Netze können auch verzweigt werden, wenn viele Bauteile mit einem Netz verbunden werden (z. B. GND).



Ein neues Schema in EAGLE erstellen. 🔍

Man kann die Schaltung noch auf Plausibilität testen (Electrical Rule Check – Erc). Dabei wird geprüft, ob Kurzschlüsse existieren und alle benötigten Pins eines Bauteils angeschlossen sind. Um diesen Test auszuführen, drückt man auf die Schaltfläche „Erc“.

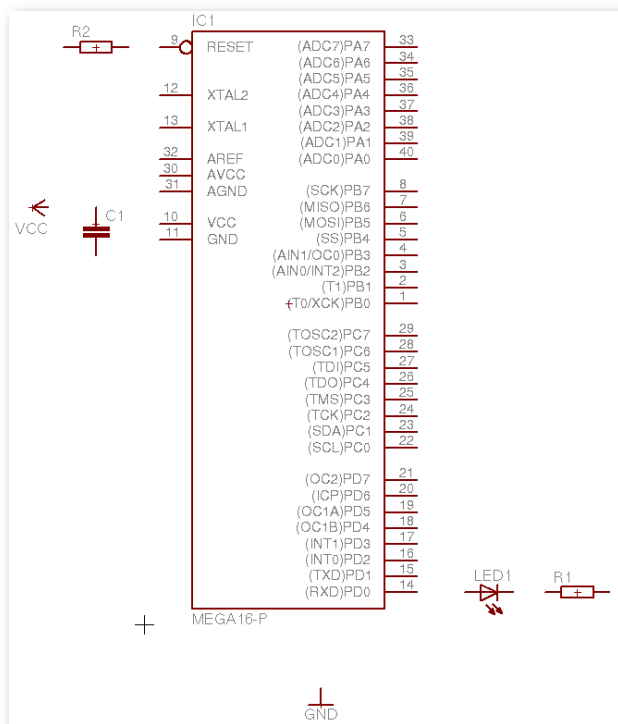
### Beispiel

Als Beispiel soll nun die Schaltung entworfen werden, die auch in „AVR-Mikrocontroller unter Linux“ ([freiesMagazin 06/2009 \[3\]](#)) gezeigt wurde. Man startet *EAGLE* und legt über „File » New“ ein neues Projekt an. Wenn man mit der rechten Maustaste auf das neue Projekt klickt, so kann man über „New“ den Menüpunkt „Schematic“ auswählen.

Nun klickt man auf das Icon „Add“ und scrollt im linken Fenster bis zum Paket **atmel** herunter. Dort klickt man auf das kleine Dreieck neben „atmel“, damit sich das Paket öffnet. In diesem Paket sucht man nun „MEGA 16-P“, dies ist ein ATmega16 im DIL-Gehäuse. Sobald das Bauteil ausgewählt ist sieht man auf der rechten Seite die Darstellung im Schema. Rechts daneben ist die Darstellung für das Boardlayout. Darunter sind weitere Informationen über das Bauteil angegeben. Nun klickt man auf „OK“ und platziert den Mikrocontroller mit der linken Maustaste irgendwo in die Schaltung.

Als nächstes wird die LED ausgewählt. Nachdem man auf „Add“ geklickt hat, gibt man unten links in der Suche „led“ ein und drückt **Return**. In den Ergebnissen sucht man nun das Paket **led** und darin das Paket **LED**. Dort gibt es eine



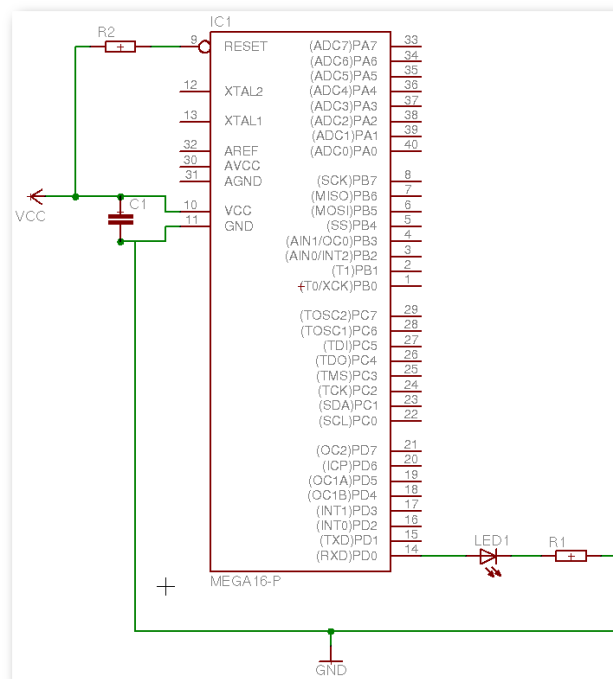


Das Schema mit platzierten Bauteilen. Diese müssen noch verbunden werden. 🔍

„LED5MM“, die man auswählt und neben den Pin „PD0“ des Mikrocontrollers platziert.

Ähnlich werden nun ein Kondensator und ein Widerstand aus dem Paket `rc1` eingefügt. Den Widerstand kann man mit dem „Copy“-Werkzeug kopieren. Ein Widerstand kommt vor den RESET-Pin und einer neben die LED. Der Kondensator gehört neben die „VCC“- und „GND“-Pins des ATmega16.

GND und VCC findet man am besten, indem man sie genau so in die Suche eingibt.



Das fertige Schema der Platine. 🔍

Die Bauteile werden nun wie in der Schaltung aus [freiesMagazin 06/2009](#) mit Netzen verbunden und unter einem beliebigen Namen gespeichert. Die Dateiendung ist `.sch`.

Zum Schluss kann man noch den Erc-Test ausführen. Dieser ergibt drei Fehler, da die Pins AREF, AGND und AVCC nicht verbunden sind. Diese werden jedoch nur für die Konvertierung von Analog zu Digital gebraucht.

## Platinenlayout entwerfen

Sobald die Schaltung fertig ist, kann man über die Schaltfläche „Board“ zur Ansicht des Board-

layouts wechseln. Hier sieht man jetzt die Umriss einer Platine und daneben alle genutzten Bauteile. Diese sind mit gelben Linien entsprechend den Netzen verbunden. Die Bauteile kann man nun wie beim Entwurf der Schaltung platzieren.

Die weißen Ränder der Platine können mit dem „Move“-Werkzeug verschoben werden. In der Freeware-Version sind dem jedoch Grenzen gesetzt – verkleinern geht aber immer.

*EAGLE* kann auch vielschichtige Platinen erstellen. Wenn man die Platine selber ätzen möchte, empfiehlt es sich, das Layout auf eine Schicht (Layer) zu begrenzen.

## Routing

Da die gelben Linien nur die logischen Verbindungen repräsentieren, müssen die Leitungen noch gezogen werden (die Platine wird „geroutet“). *EAGLE* unterstützt den Benutzer dabei mit seinem Autorouter. Immer wenn Bauteile verschoben wurden, sollte man einmal das „Ratsnest“-Symbol anklicken, damit die Verbindungen optimiert werden. Danach kann man den Autorouter starten. Bei „Preferred Directions“ sollte man „TOP“ auf „N/A“ stellen, damit die Platine nur auf einer Seite mit Leiterbahnen bestückt wird. Doppelseitige Platinen sind relativ schwer zu ätzen. „BOTTOM“ sollte auf „\*“ eingestellt werden, sodass die Leitungen keine bevorzugte Richtung haben. Das Routing Grid kann ruhig auf 20-30 gestellt werden. Die voreingestellten 50 reichen für dieses Beispiel aber auch. Dies gibt lediglich an, wie fein das Gitter (grid) ist, auf dem die Bahnen verlegt werden können. Je geringer der Wert,



desto länger dauert der Routingvorgang, aber desto höher ist auch die Wahrscheinlichkeit, dass ein gutes Layout gefunden wird.

In diesem Beispiel gibt es nur wenige Verbindungen, sodass der Autorouter alle Leitungen sehr schnell verlegen kann. Je nach Rechenleistung und Umfang der Platine kann der Routingvorgang mehrere Minuten bis Stunden dauern. (Dieses Problem ist NP-Vollständig [4].) Trotzdem schafft der Autorouter meist nicht alle Verbindungen und man muss selbst einige Leitungen ziehen. Das Routen von Hand erfordert etwas Erfahrung und man kann sich tagelang damit beschäftigen. Oft ist es leichter – wenn auch nicht immer schön – Drahtbrücken zu nutzen.

Um eine Leitung (wieder) aufzutrennen, kann man das Werkzeug „Ripup“ nutzen. Wenn es ausgewählt ist, klickt man auf die gewünschte Leitung. Wenn man alle Leitungen entfernen möchte (um ein neues Autorouting zu starten), klickt man erst auf „Ripup“ und dann auf „GO“.

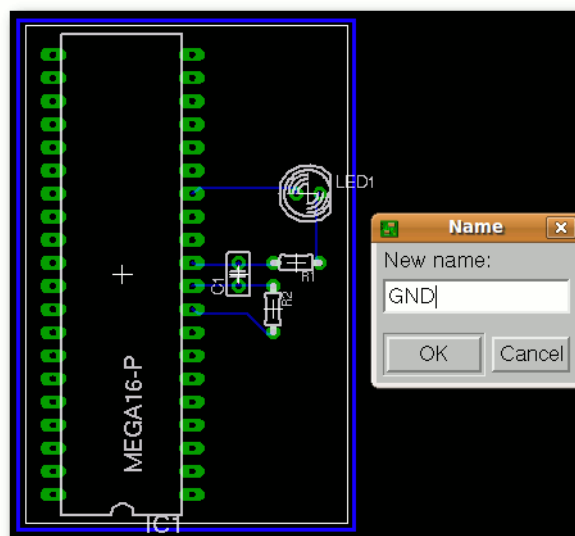
### Techniken

Einige Anwendungsfälle, wie z. B. Hochfrequenztechnik, benötigen Masseflächen an bestimmten Stellen der Platine. Aber auch sonst ist es nicht verkehrt, jede freie Fläche als Kupferfläche zu modellieren, da so beim Ätzzvorgang weniger Kupfer weg geätzt werden muss und das Ätzbad nicht so schnell verbraucht wird.

Um eine Massefläche zu erstellen, nutzt man das „Polygon“-Werkzeug und zieht ein Rechteck großzügig um die Platine herum. Für eine Masseflä-

che gibt man diesem den Namen „GND“ und drückt „Ratsnest“. Oft bleiben noch einige freie Flächen übrig, die sich nicht mit Masse verbinden lassen. In diesem Fall kann man ein zweites Polygon erstellen, dessen Namen man nicht verändert. Nach einem weiteren Ratsnest sollte es nun keine freien Flächen mehr geben.

Die Masseflächen sollte man immer vor dem Routen einfügen, da dadurch das Routing oft vereinfacht wird.



Erstellen einer Massefläche. 🔍

### Drahtbrücken

Eine Drahtbrücke ist ein Kabel, welches eine Leitung ersetzt und normalerweise auf der Oberseite der Platine verlegt wird. Über Löcher mit darunter liegenden Kupferflächen (Pads) lassen sich Drahtbrücken auf die gleiche Weise wie andere Bauteile einlöten. Zum Modellieren von Draht-

brücken wird eine Leitung auf einem anderen Layer verlegt, welches später nicht ausgedruckt sondern durch die entsprechenden Kabel ersetzt wird.

Um eine Drahtbrücke zu setzen, erstellt man zuerst zwei Pads mit dem Tool „Via“. Nun wechselt man auf das Werkzeug „Route“ und verbindet die Pads mit den Leitungen, die mit der Drahtbrücke verbunden werden sollen. Nun wechselt man auf das Layer „1 TOP“. Dort kann man beide Pads miteinander verbinden.

Um zwei Masseflächen miteinander zu verbinden, gibt man den Pads den gleichen Namen wie den Masseflächen: „GND“. Diese brauchen dann nur noch auf Layer 1 verbunden werden.

### Bohrungen

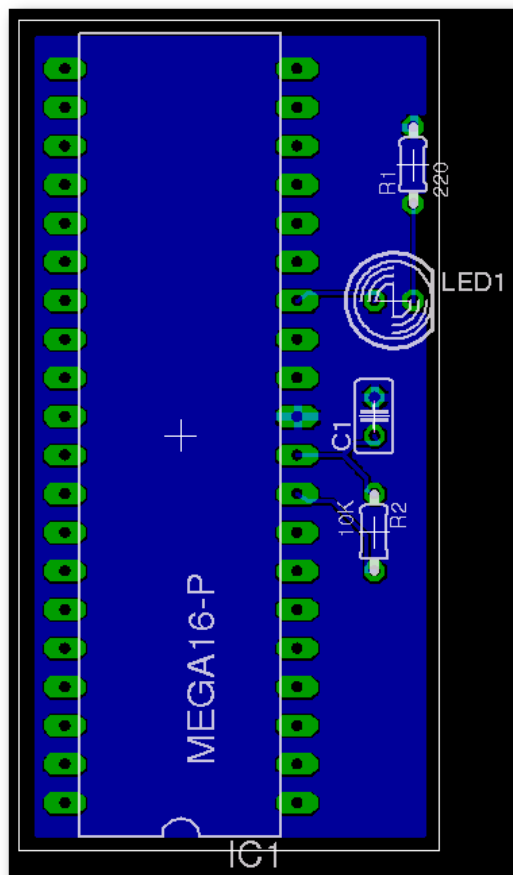
Möchte man die Platine später mit Schrauben fixieren, ist es sinnvoll mit dem Werkzeug „Hole“ die Bohrungen zu markieren. Die Größe der Löcher kann über die Einstellung „Drill“ bestimmt werden.

### Text

Oft ist es sinnvoll, auch die Platine selbst zu beschriften. Eine Versionsnummer z. B. hilft später ungemein, eine Platine mit Verbesserungen wieder zu erkennen. Wenn einem Jahre später nochmal die Platine in die Hand fällt, ist es sicher auch nützlich, wenn darauf steht, wofür sie ist (oder war). Auch zum Belichten ist eine Beschriftung praktisch, weil man dann am besten erkennt, ob die Folie richtig herum aufliegt (und nicht spiegelverkehrt).



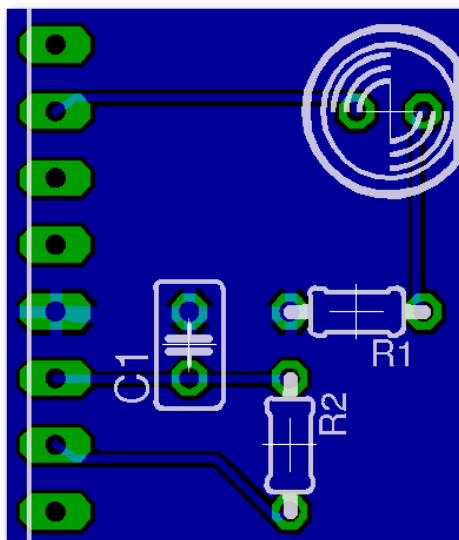
Daher sollte man mit dem Werkzeug „Text“ ganz zum Schluss auf einen freien Bereich der Platine klicken und in das sich öffnende Fenster den Namen und die Version der Platine angeben (man kann natürlich auch noch andere Beschriftungen machen). Die Schrift wird auf dem BOTTOM-Layer gespiegelt angezeigt. Mit dem Werkzeug „Change“ kann man Schriftart und -größe verändern.



Das fertig geroutete Board. 🔍

### Einstellungen festlegen und testen

Beim Platinenlayout hat man neben dem Erc noch die Möglichkeit einzustellen und zu testen, wie groß die Isolationsschichten, also die Zwischenräume zwischen zwei Leiterbahnen, sein sollen. Dieser Design Rule Check (Drc) wird mit der Schaltfläche „Drc“ aufgerufen. Im sich öffnenden Dialog kann man nun die Abstände zwischen verschiedenen Typen wie z. B. Netzen oder Vias bestimmen. Nach einem Klick auf „Apply“ wird versucht, das Layout entsprechend anzupassen. Mit „Check“ kann man nun überprüfen lassen, ob dies überall möglich war. Gab es Probleme, so öffnet sich ein Fenster, in dem alle Fehler aufgelistet werden. Der in diesem Fenster ausgewählte Fehler wird auch auf der Platine angezeigt, sodass man versuchen kann, das Problem zu beheben.



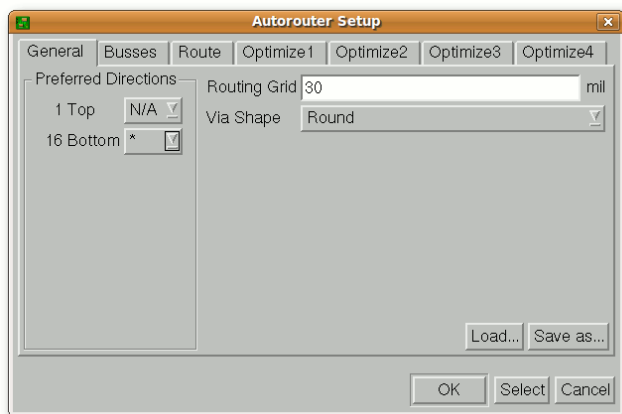
Ein Ausschnitt des gerouteten Boards. 🔍

### Beispiel

Nachdem das Schema fertig ist, soll nun das Boardlayout erstellt werden. Dazu wechselt man mit dem Symbol „Board“ auf die Boardansicht.

Zuerst werden die Bauteile nun in die Fläche der Platine platziert. Die Größe der Platine kann man dabei auch anpassen. Nun empfiehlt es sich eine Massefläche zu definieren. Dazu klickt man auf die Schaltfläche „Polygon“ und klickt im (oder gegen den) Uhrzeigersinn einmal an jede Ecke der Platine, sodass ein Rechteck um die Platine herum entsteht. Mit dem Werkzeug „Name“ kann man nun auf das Rechteck klicken und ihm den Namen „GND“ geben. Danach drückt man einmal „Ratsnest“, damit die Massefläche angezeigt und die Verbindungen optimiert werden. Wie man sieht, sind alle Masseverbindungen nun verschwunden und die Bauteile haben an den entsprechenden Pins eine direkte Verbindung zur Massefläche erhalten. Nun kann man die Teile meistens noch günstiger (also enger) anordnen.

Sind die Bauteile fertig platziert, kann man mit dem Routingvorgang beginnen. Zuerst noch einmal auf „Ratsnest“ klicken, damit die Verbindungen auch wirklich optimal sind. Danach klickt man auf die Schaltfläche „Auto“ und wählt die oben vorgeschlagenen Einstellungen aus. Nach einem Klick auf „OK“ beginnt der Routingvorgang. Bei dieser kleinen Beispielplatine ist er sehr schnell abgeschlossen und es sollten auch alle gelben Verbindungslinien verschwunden sein. Ein Klick auf „Ratsnest“ zeigt in der



Einstellungsdialog des Autorouters. 🔍

Statusleiste an, ob noch offene Verbindungen bestehen (und wenn ja, wieviele).

Nun kann man noch die Tests Erc und Drc durchlaufen lassen und das Layout speichern. Die Dateiendung für Platinenlayouts ist **.brd**.

### Tipps zum (manuellen) Routen

Manchmal schafft es der Autorouter einfach nicht, alle Verbindungen zu routen. Man kann alle Leitungen auftrennen und die Bauteile etwas verschieben, um den Autorouter noch einmal laufen lassen. Das bringt oft erstaunlich gute Ergebnisse. Doch auch damit lassen sich oft nicht alle Verbindungen routen. In diesem Fall muss man die restlichen Leitungen manuell routen.

Dafür klickt man auf das Werkzeug „Route“ und dann auf die Leitung oder den Pin, der noch nicht verbunden ist. Nun kann man eine Linie auf die gleiche Weise ziehen wie beim Schaltplan ein Netz. Oft empfiehlt es sich, die verschiedenen

Winkelooptionen des Werkzeugs zu nutzen, die man oben in der Leiste anklicken kann. Alternativ kann man auch beim Ziehen der Leitung die rechte Maustaste drücken, um den Winkel zu ändern.

Auch die Leitungsdicke lässt sich über die Auswahl „Width“ ändern, sodass man zwischen den Pins eines Bauteils durchkommt. Wenn eine große Massefläche besteht, sieht man die neu gerouteten Leitungen oft nicht. Mit einem Klick auf Ratsnest wird das Layout neu gezeichnet.

Um sich den Verlauf einer Leitung anzusehen, kann man auf die Schaltfläche „Show“ und dann auf das gewünschte Netz klicken (das funktioniert auch im Schema). Mit dem Werkzeug „Info“ kann man sich zu jeder Leitung und jedem Bauteil Informationen wie Name, Wert und Position anzeigen lassen (funktioniert im Schema und beim Boardlayout).

Auf jeden Fall sollte man sich am Anfang die Ziele (gerade was die Größe der Platine angeht) nicht zu hoch stecken und ruhig regen Gebrauch von Drahtbrücken machen.

### Vorschau bild der Platine erzeugen

Mit dem Skript *EAGLE3D* [5] kann man sich eine POV-Ray-Datei (siehe „Der Raytracer POV-Raytracer“ *freiesMagazin* 09/2009 [6]) erzeugen lassen, welche die Platine darstellt.

Das Script kann man einfach irgendwo im Dateisystem unpacken. Die Bauteilbibliothek gehört ins Verzeichnis **povray** des *EAGLE3D*-Ordners

(sie sollte dort im Ordner **img** liegen). In der Boardansicht klickt man nun auf die Schaltfläche „ULP“ und wählt das Skript **3d341.ulp** aus dem Verzeichnis **ulp** des *EAGLE3D*-Ordners aus.

Die Einstellungen können so bleiben und man braucht nur noch auf „POV-Datei schreiben“ oder „POV-Datei schreiben und Ende“ zu klicken. Nun wird man noch gefragt, welche Logos auf den ICs abgebildet werden sollen, ob die ICs gesockelt sein sollen, welche Farbe die LEDs haben sollen und Ähnliches. Danach wird die POV-Ray-Datei geschrieben (der Pfad und Dateiname kann im ersten Dialog von *EAGLE3D* bestimmt werden), welche man mittels

```
$ povray -W800 -H600 dateiname.pov
```

rendern kann. (POV-Ray muss dafür installiert sein). Dabei steht **-W800** für die Breite von 800 Pixeln und **-H600** für die Höhe von 600 Pixeln. Diese Angaben können nach Belieben geändert werden. Je größer man das Bild macht, desto länger dauert das Rendern. Das Ergebnis wird in **dateiname.png** gespeichert.

Wenn man die Werte der Zeilen 44–50 in der POV-Ray-Datei ändert, wird dadurch die Position der Kamera verschoben, wodurch man sehr schöne Nahaufnahmen erzeugen kann. Für das obige Beispiel kann man einfach mal folgende Werte ausprobieren:

```
#local cam_x = 20;
#local cam_y = 14;
#local cam_z = -45;
```



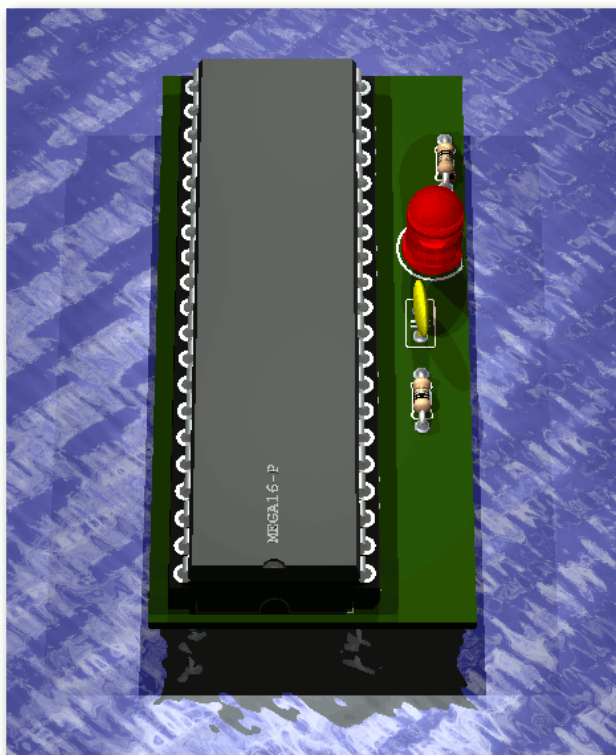


```
#local cam_a = 20;
#local cam_look_x = 10;
#local cam_look_y = 2;
#local cam_look_z = 3;
```

Nach jeder Änderung der Datei muss das Bild mit dem POV-Ray-Befehl erneut gerendert werden.

## Die Platine ätzen

Um die Platine zu ätzen, braucht man eine Menge Zubehör. Neben der Fotoplatine, die man bei fast allen Elektronikhändlern beziehen kann, benötigt man eine Lichtquelle, ein Entwicklerbad



Das fertig gerenderte Bild. 🔍

und ein Ätzbad. (Man kann beides bei Elektronikversandhändlern erhalten. Dosierungsanweisungen sind normalerweise dabei). Ein Bohrer mit einem Gestell ist auch sehr praktisch. Zum Entfernen des überschüssigen Fotolacks wird zum Schluss noch normaler Spiritus genutzt.

Ein Belichtungsgerät kann man sich selber bauen. Dafür gibt es eine Anleitung von Andreas Behnisch [7] (vom mir nicht getestet).

Zum Drucken der Schaltung schaltet man in der Boardansicht mit der Schaltfläche „Display“ alle Anzeigen bis auf „Bottom“ (bzw. „Top“ für die Oberseite von doppelseitigen Platinen), „Pads“, „Vias“ und „Dimension“ aus. Danach kann man die Schaltung am besten auf Projektorfolien drucken (die Folien, die sonst für Tageslichtprojektoren genutzt werden).

Nachdem man Schaltung und Platine zurecht geschnitten hat, macht man das Licht aus und zieht die Schutzfolie von der Platine.

Nun wird die ausgeschnittene Schaltung mit durchsichtigem Klebeband auf der Fotoplatine befestigt (die Schrift auf der Schaltung sollte lesbar sein) und auf das Belichtungsgerät gelegt.

Die Belichtungsdauer ist von der Intensität der Lampe und der verwendeten Platine abhängig. Am Anfang muss man meist etwas probieren, bis man das gewünschte Ergebnis erreicht hat. Meist reichen 2-3 Minuten Belichtungszeit.

Nach dem Belichten kommt die Platine in den Entwickler. Sobald die Schaltung gut erkennbar

ist und die Reste des entwickelten Fotolacks verschwunden sind, kann man die Platine aus dem Entwickler nehmen und unter klarem Wasser abspülen. (Ab jetzt kann das Licht auch wieder angeschaltet werden.) Nun kommt sie ins Ätzbad, in dem sie so lange bleibt, bis alle belichteten Kupferflächen aufgelöst sind.

Mit dem Spiritus kann man nun die Fotolack-Reste entfernen. Danach sollte man die Platine mit einem Leitungsprüfer durchmessen, um zu sehen ob alle Verbindungen leiten und kein Kurzschluss vorhanden ist.

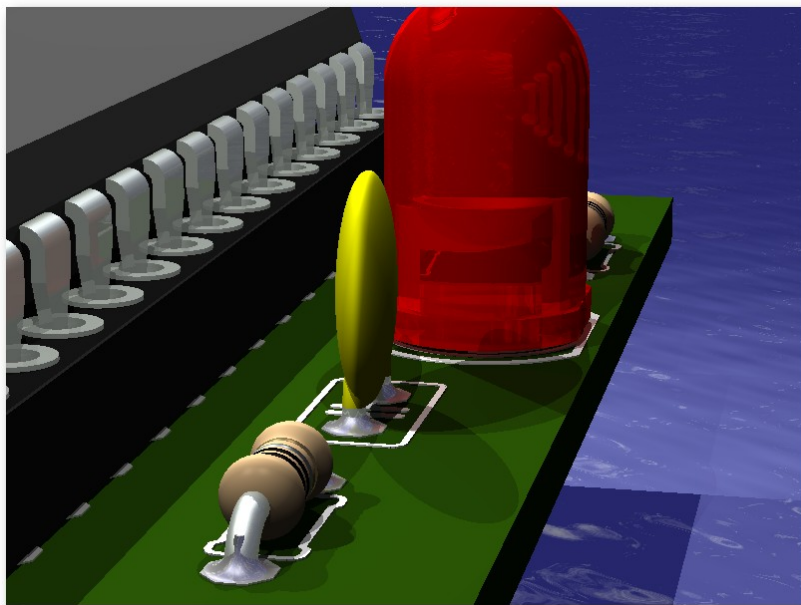
Als nächstes müssen alle Löcher gebohrt werden. Danach kann die Platine bestückt und gelötet werden.

Weitere Informationen zum Ätzen gibt es im Wiki-book „Platinen selber herstellen“ [8].

## Tipps für doppelseitige Platinen

Das Wichtigste beim Erstellen von doppelseitigen Platinen ist, sehr genau zu arbeiten und jeden Schritt sorgfältig zu überlegen. Es ist viel Arbeit, eine solche Platine anzufertigen, und wenn man nicht sorgfältig arbeitet, klappt es meist nicht.

Bei der Arbeit mit dem Autorouter empfiehlt es sich, mehrere Ansätze auszuprobieren. So kann man zuerst eine Seite routen und die übrig gebliebenen Verbindungen auf der anderen Seite erneut mit dem Autorouter routen. Dafür stellt man das erste Mal „TOP“ auf „N/A“ und „BOTTOM“ auf „\*“ und für den zweiten Durchlauf genau umgekehrt.



Das fertige Bild mit geänderter Kameraposition. 🔍

Ein anderer Ansatz ist es, beide Seiten gleichzeitig zu routen und für „TOP“ „/“ zu wählen und für „BOTTOM“ „-“. Das ganze sollte man auch mal umgekehrt testen, um zu sehen, ob dann bessere Ergebnisse erzielt werden (also „TOP“ auf „-“ und „BOTTOM“ auf „/“).

Damit der Autorouter Bauteile wie Wannenstecker nicht auf der Oberseite (TOP-Layer) verbindet (dies ist sehr ungünstig zu löten), kann man mit dem „Polygon“-Werkzeug Bereiche definieren, die vom Autorouter nicht betreten werden dürfen. Dazu wählt man das Layer „tRestrict“ aus („bRestrict“ verbietet das Betreten im BOTTOM-Layer) und rahmt damit die entsprechenden Bauteile ein.

Nach dem Routing ist es meist vorteilhaft, die vom Autorouter gesetzten Vias mit dem Werkzeug „Change“ über den Menüpunkt „Drill“ zu verkleinern.

Zum Belichten hat es sich bewährt, mit durchsichtigem Klebeband eine Tasche aus den beiden ausgedruckten Schaltungen zu basteln und die Platine dort hineinzulegen. Dadurch verrutscht die Platine nicht so leicht. Das Belichten ist einer der kritischsten Punkte beim Erstellen von doppelseitigen Platinen, da beide Seiten passgenau aufeinander liegen müssen. Wenn die beiden Seiten nicht genau aufeinander liegen sind die Bohrlöcher

später auf einer Seite falsch.

Beim Bohren sollte man erst Löcher in der Nähe der vier Ecken bohren, um sicher zu gehen, dass die Schaltung richtig passt. Wenn man erst nach der Hälfte der zu bohrenden Löcher bemerkt, dass die Seiten nicht genau aufeinander liegen, hat man sich eine Menge Arbeit umsonst gemacht.

### Weitere Informationen

Das Handbuch von *EAGLE* befindet sich in den Installationsdateien. Wurde *EAGLE* über die Paketverwaltung installiert, so sollte es sich unter `/usr/share/doc/eagle-data/help` befinden. Ansonsten kann die FAQ [9] bei vielen Pro-

blemen weiter helfen. Einige gute Tipps gibt es auch im Wiki von Mikrocontroller.net [10].

### LINKS

- [1] <http://www.cadsoft.de>
- [2] <http://www.cadsoft.de/download.htm>
- [3] <http://www.freiesmagazin.de/freiesMagazin-2009-06>
- [4] [http://de.wikipedia.org/wiki/NP\\_%28Komplexit%C3%A4tsklasse%29](http://de.wikipedia.org/wiki/NP_%28Komplexit%C3%A4tsklasse%29)
- [5] <http://www.matwei.de/doku.php?id=de:eagle3d:eagle3d>
- [6] <http://www.freiesmagazin.de/freiesMagazin-2009-09>
- [7] <http://www.jogis-roehrenbude.de/Leserbriefe/Platinenbelichter/Platinenbelichtung.htm>
- [8] [http://de.wikibooks.org/wiki/Platinen\\_selber\\_herstellen](http://de.wikibooks.org/wiki/Platinen_selber_herstellen)
- [9] <http://www.cadsoft.de/faq.htm>
- [10] [http://www.mikrocontroller.net/articles/Eagle\\_im\\_Hobbybereich](http://www.mikrocontroller.net/articles/Eagle_im_Hobbybereich)

### Autoreninformation



**Marcel Jakobs** hat für viele seiner Projekte selbst Platinen entwickelt. Dafür nutzte er jedes Mal EAGLE. Die Bilder aus EAGLE3D haben schon in mancher seiner Arbeiten Verwendung gefunden.

Diesen Artikel kommentieren





## 41. DANTE-Mitgliedertagung in Esslingen von Dominik Wagenführ

**A**m 12. September 2009 fand in Esslingen am Neckar die 41. Mitgliedertagung der Deutschsprachigen Anwendervereinigung TeX e.V. (DANTE) [1] statt. Im Rahmen der Veranstaltung gab es zahlreiche Vorträge zu verschiedenen  $\LaTeX$ -Themen.

### Mitgliederversammlung

Klaus Höppner begrüßte um 9 Uhr die circa 40-50 Teilnehmer zur 41. Mitgliederversammlung [2]. Sehr ausführlich berichtete er danach über die verschiedenen TeX-Tagungen, die weltweit in diesem Jahr stattfanden. So erfuhr man, wie die TUG 2009 in Notre Dame, Indiana, USA, verlaufen ist und sah auch einige Bilder des schönen Campus. Auch von der EuroTeX 2009 und dem 3. ConTeXt-Meeting, die dieses Jahr in den Niederlanden stattfanden, wurde berichtet. Neben dem Hinweis auf kommende Tagungen, gab es auch einen Rückblick auf die verschiedenen deutschen Linux-Veranstaltungen, wie die Chemnitzer Linux-Tage [3] und den LinuxTag in Berlin [4], auf denen DANTE mit Ständen und Vorträgen vertreten war.

Bei der Gelegenheit habe ich auch auf meine zwei  $\LaTeX$ -Workshops am 17. Oktober 2009 hingewiesen, die ich im Rahmen der Ubucon [5] an der Universität in Göttingen halten werde.

Im Anschluss wurden verschiedene Interna besprochen, vor allem aber Infos zur TeXnischen Komödie [6] wurden behandelt. Dabei gab es

auch von Seiten der Teilnehmer einige Verbesserungsvorschläge und Wünsche. So würde ein Teilnehmer gerne mehr einsteigerfreundliche Artikel in der Komödie sehen. Hier trifft aber zu, was alle Magazine betrifft, die keine Festangestellten haben: Es muss sich erst einmal ein Autor finden, der etwas dazu schreibt.

### Projekt: PostScript und PDF

Nach einer halbstündigen Pause gab es den ersten Vortrag. Herbert Voß, Autor vieler hilfreicher  $\LaTeX$ -Lehrbücher [7], die auf der Tagung auch zum Kauf angeboten wurden, stellte ein Projekt vor, bei dem es darum geht, PostScript-Grafiken, die bisher nur über den Umweg **latex** » **dvips** » **ps2pdf** » **includegraphics** in PDF-Dokumenten genutzt werden konnten, direkt mittels **pdflatex** übersetzen zu können. Mittels des Perl-Programmes **pst2pdf** [8] werden Präampel, Text und Grafik eines  $\LaTeX$ -Dokumentes aufgespalten, die Grafiken einzeln konvertiert und am Ende alles wieder zu einem Dokument zusammengefügt, aus dem man das PDF erstellen kann.

Das Programm hat dabei aber noch Einschränkungen. Zum Beispiel ignoriert es alle lokalen Änderungen für Umgebungen wie **pspicture**, die außerhalb dieser definiert werden und auch nicht global in der Präambel gegeben sind. Darüber hinaus gibt es Probleme, wenn solche Grafiken in **verbatim**-Umgebungen eingebettet sind. Von einer neu definierten Umgebung auf Basis der

**pspicture**-Umgebung sollte man daher wohl auch Abstand nehmen.

Ingesamt ist das aber ein spannendes Vorhaben, welches man auch in Zukunft beachten sollte, nimmt es doch einige Hürden, die einem  $\LaTeX$ -Anwender heute noch in den Weg gestellt werden, wenn man schöne Grafiken darstellen will.

### Biber – Der BibTeX-Nachfolger

Martin Sievers stellte im folgenden Vortrag Biber [9] vor, welches als Nachfolger von BibTeX angesehen werden kann. Mittels BibTeX und **biblatex** kann man sehr leicht Literaturdatenbanken erzeugen und schnell in einem Dokument referenzieren. Der Nachteil ist, dass BibTeX vor allem bei der Sortierung von Unicode-basierenden Quellen (also auch deutschen Umlauten) Probleme hat und diese falsch sortiert. Daneben kann es bei großen Literaturdatenbeständen zu einem Überlauf des Speichers kommen. All dies soll mit Biber behoben werden.

Dazu gibt es auch eine neue XML-basierende Sprache BibLaTeXXML, mit der man die Datenbankeinträge in einer XML-Sprache leichter beschreiben kann.

### Graphviz-DOT

Nach der Mittagspause stellte Michael Niedermair das Programm Graphviz [10] vor, mit dem man anhand von einfachen Regeln leicht Strukturdiagramme, Pfeildiagramme, MindMaps und Netzpläne erstellen kann. Dabei gibt man nur die





einzelnen Zuordnungen der Knoten und Verbindungen an und auf welche Art die Grafik erzeugt werden soll.

Aufgrund einiger Einschränkungen bei der grafischen Darstellung (keine besonderen Linien, kein Mathemodus als Beschriftung, keine Schriftarteinbindung) wurde auch der Export mittels `dot2tex` [11] dargestellt. Über dieses kann man ein erstelltes Diagramm in  $\TeX$ -Code exportieren und dann mittels des **PSTricks**-Paketes [12] verändern.

## Dynamische Bücher

Im Anschluss folgte ein sehr interessanter Vortrag von André Wobst. Für die Erstellung von Tagungsbänden wurde ein Online-Programm erstellt, was es erlaubt, einzelne Beiträge zu selektieren und diese nach eigenen Sortierungsverfahren als PDF auszugeben. Dabei werden die Texte aus der Datenbank genommen und in  $\LaTeX$ -Code abgelegt, die dann mittels HEVEA [13] in ein einfaches HTML-Format konvertiert oder eben per **pdflatex** übersetzt werden können.

Eine Anforderung ist dabei, dass der TeX2HTML-Konverter sehr schnell arbeitet, da bei einem Tagungsband mit mehreren tausend Beiträgen nicht erst Stunden auf das Online-Ergebnis gewartet werden kann. HEVEA soll dabei diese Anforderungen erfüllen.

Meine persönlichen Erfahrungen mit HEVEA zeigen leider ein anderes Bild, insbesondere was die allgemeine Anwendung angeht. Die Alternativen `tex4ht` [14] und `tth` [15] sind hier aber auch

nicht viel besser und funktionieren oft gar nicht oder nur eingeschränkt. (Das hoffnungslos veraltete `latex2html` [16] bleibt außen vor.)

## Komfortable Makefiles

Danach folgte ein weiterer Vortrag von Michael Niedermair, der die Verwendung von Makefiles [17] für  $\TeX$ -Dokumente erklärte. Dabei wird anhand verschiedener Regeln ein  $\TeX$ -Dokument nur dann kompiliert, wenn sich Inhalte geändert haben. Das Ganze kann man auch so erweitern, dass beispielsweise Grafiken aus dem Programm bei Änderungen korrekt in PDF konvertiert und dann im  $\LaTeX$ -Dokument eingebunden werden.

Zur Hälfte der Zeit wurde der Vortrag aber sehr technisch und durch die zahlreichen Beispiele zu ausführlich, sodass zu der Zeit kaum noch etwas Neues mitgenommen werden konnte.

## Having Fun with $\LaTeX$ II

Nach der Kaffeepause stellte Adelheid Grob einige interessante  $\LaTeX$ -Pakete vor, die die meisten Zuhörer sicherlich noch nicht kannten. Neben sehr speziellen Paketen wie **cooking**, mit dem man Rezepte setzen kann, deren Layout sich an dem der Dr.-Oetker-Kochbücher orientiert oder **guitar**, welches für die Akkorde über Liedtexten zuständig ist, wurden auch sehr hilfreiche Pakete wie **pseudocode** für die Darstellung von Algorithmen als einfachen Pseudocode, **eemeir** zum Setzen von „geschlechtsunabhängigen“ Dokumenten und **uebungsblatt** zur Erstellung von Übungsblättern und Klausuren vorgestellt.

## Was gibt es Neues bei $\TeX$ Live?

Den Abschluss bildete Klaus Höppner, der einen Vortrag von Karl Berry zum aktuellen Stand von  $\TeX$  Live 2009 zum Besten gab, welches im Herbst 2009 erscheinen soll.

Dabei beschränken sich die Änderungen gegenüber der Version 2008 auf Kleinigkeiten. So werden mittels **epstopdf** EPS-Grafiken in  $\LaTeX$ -Dokumenten, die mit **pdflatex** übersetzt werden, automatisch nach PDF konvertiert.

Mit **write18** kann man externe Befehle aus  $\TeX$  heraus aufrufen, welches aufgrund der Sicherheit aber nur bestimmte Kommandos ausführen kann. (Man kann diesen Sicherheitsmechanismus aber sehr leicht aushebeln.)

Vorgelegt wurde auch der neue minimalistische  $\TeX$ -Editor TeXworks [18], der einen eingebauten PDF-Betrachter besitzt. Man kann mit ihm auf diese Art im PDF-Dokument eine Stelle markieren und auf den zugehörigen  $\TeX$ -Quellcode im Editor springen, was sicherlich in manchen Fällen eine hilfreiche Funktion ist.

## Schlussbemerkung

Die Veranstaltung war sehr locker, alle waren freundlich und es gab viele interessante Vorträge. Am Vorabend des Freitags und am Samstagabend traf man sich in verschiedenen Restaurants, bei dem man sich auch über andere Themen außer  $\LaTeX$  unterhalten konnte – was natürlich auf der Tagung selbst auch möglich war. Am Sonntag gab es noch eine touristische Führung durch Esslingen.





Wer an  $\text{\LaTeX}$  interessiert und bei einer der nächsten Tagungen in der Nähe ist (die nächsten werden in Dortmund, Aachen und Bremen stattfinden), sollte sich unbedingt dort anmelden. Auch als nicht DANTE-Mitglied kann man bei den Vorträgen zuhören und einiges an Wissen mitnehmen.

LINKS

- [1] <http://www.dante.de/>
- [2] <http://www.dante.de/events/mv41.html>
- [3] <http://chemnitzer.linux-tage.de/>
- [4] <http://www.linuxtag.org/2009/>
- [5] <http://www.dante.de/DTK.html>

- [6] <http://projekte.dante.de/DanteFAQ/Literatur>
- [7] <http://texcatalogue.sarovar.org/entries/pst2pdf.html>
- [8] <http://biblatex-biber.sourceforge.net/>
- [9] <http://www.graphviz.org/>
- [10] <http://www.fauskes.net/code/dot2tex/>
- [11] <http://tug.org/PSTricks/>
- [12] <http://hevea.inria.fr/>
- [13] <http://www.lrz-muenchen.de/services/software/sonstiges/tex4ht/mn.html>
- [14] <http://hutchinson.belmont.ma.us/tth/>
- [15] <http://www.latex2html.org/>
- [16] <http://www.ijon.de/comp/tutorials/makefile.html>

- [17] <http://code.google.com/p/texworks/>

Autoreninformation

**Dominik Wagenführ** ist unter anderem für das Layout in freiesMagazin zuständig, was mit  $\text{\LaTeX}$  gesetzt wird. Aus dem Grund interessiert er sich auch immer für neue Pakete und Ansätze auf diesem Gebiet.

Diesen Artikel kommentieren



FUN FACT: DECADES FROM NOW, WITH SCHOOL A DISTANT MEMORY, YOU'LL STILL BE HAVING THIS DREAM.

„Students“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/557>

## Erklärungen zum zweiten Programmierwettbewerb

**A**ufgrund des Erfolges des ersten Programmierwettbewerbes von freiesMagazin im April dieses Jahres [1], bei dem 13 Teilnehmer eine KI eingereicht haben, haben wir uns entschlossen, erneut einen Programmierwettbewerb zu veranstalten. Dieses Mal wird die Aufgabe etwas anspruchsvoller, aber auch etwas lustiger. Wir hoffen erneut auf eine rege Teilnahme.

### Spielidee

Die Idee basiert dieses Mal auf dem Spiel „Robo-Rally“ von Bill McQuillan aus dem Hause Wizards of the Coast [2]. Man steuert über Spielkarten einen Roboter durch eine Fabrik auf dem Weg zu verschiedenen Zielpunkten. Dabei kommen einem oftmals andere Roboter in die Quere und zusätzlich gibt es in der Fabrik verschiedene Objekte, die die Bewegungen des Roboters beeinflussen können.

Aufgrund der zeitlichen Begrenzung des Wettbewerbs wurde aber auf die Interaktion mit anderen Robotern und auf einige Spielelemente verzichtet. Auf diese Art soll die zu programmierende Künstliche Intelligenz (KI), d. h. das Programm, welches den Roboter steuert, nicht zu komplex werden.

### Das Spiel im Detail

#### Der Roboter

Wie bereits erwähnt, ist es diesmal die Aufgabe der Teilnehmer, eine KI für einen Roboter zu pro-

grammieren. Dieser kann durch Spielkarten vorwärts und rückwärts fahren oder sich nach rechts und links drehen.

Ziel des Spiels ist es, den Roboter unbeschadet von der Startposition ins Ziel zu führen.

#### Die Spielkarten

Zu Beginn jeder Runde werden acht Karten ausgeteilt, mit denen man den Roboter bewegen oder drehen kann. Von diesen acht wählt man fünf aus, deren Befehle dann sequenziell (also der Reihe nach, so wie sie ausgewählt wurden) abgearbeitet werden.

Es stehen folgende Karten (mit zugehöriger Anzahl im Kartenstapel) zur Auswahl:

Karten		
Titel	Anzahl	Wahrscheinlichkeit
Bewegung 3 Felder vorwärts	400	0.071
Bewegung 2 F. vorwärts	800	0.143
Bewegung 1 F. vorwärts	1200	0.214
Bewegung 1 F. rückwärts	400	0.071
Nach links drehen (90 Grad)	1200	0.214
Nach rechts drehen (90 Grad)	1200	0.214
Umdrehen (180 Grad)	400	0.071
Gesamt	5600	

**Hinweis:** Es gibt also keine Karte, mit der ein Roboter einfach nur still stehenbleibt!

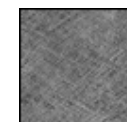
Ingesamt heißt das, es wird die erste von den fünf gewählten Karte ausgespielt, der Roboter

bewegt sich und bleibt stehen. Dann bewegen ggf. die Elemente auf dem Spielbrett (siehe unten) den Roboter und dann wird die nächste Karte ausgespielt usw.

Die hohe Anzahl der Spielkarten ergibt sich daraus, dass es bei einer zu geringen Zahl zu leicht dazu kommen kann, dass die Karten aufgebraucht sind, bevor der Roboter das Ziel gefunden hat. Mit der obigen Anzahl konnte die Referenz-KI (siehe unten) das Ziel aber jedes Mal erreichen.

#### Das Spielbrett

Wie erwähnt, gibt es auf dem Spielbrett verschiedene Spielelemente, die den Roboter beeinflussen oder behindern können.



**Boden:** Wie langweilig! Ein leeres Fabrikfeld.



**Ziel:** Das Zielfeld, zu dem der Roboter geführt werden muss.



**Wände:** Wände wirken sehr einschränkend auf die Bewegungsfreiheit des Roboters, denn sie können nicht durchbrochen werden. Fährt ein Roboter gegen die Wand, passiert aber zumindest nichts Schlimmes (außer vielleicht einer kleinen Delle)



**Kiste:** Eine Kiste sperrt ein Feld komplett. Man kann nicht anders, als das Hindernis zu umfahren.



**Loch:** Wenn ein Roboter dieses Feld aus Versehen betritt oder darauf geschoben wird, verschwindet er auf Nimmerwiedersehen.



**Öl:** Besonders glitschig! Wenn ein Roboter dieses betritt oder darauf geschoben wird, rutscht er bis zum nächsten Nichtölfeld weiter oder bis er von einer Wand gestoppt wird. Wenn er sich auf einem Ölfeld dreht, wird die Drehung verdoppelt. Aus einer Links- oder Rechtsdrehung wird zum Beispiel eine Drehung um 180 Grad.



**Drehrad:** Drehwurm garantiert. Wenn man auf diesem Feld zum Stehen kommt, wird der Roboter um 90 Grad in die jeweils angezeigte Richtung gedreht.



**Förderband:** Nur für Faule! Bleibt man auf einem Förderband stehen, bewegt es den Roboter einen Schritt in die jeweilige Richtung. Aufpassen sollte man hier, wenn am Ende der Abgrund wartet.



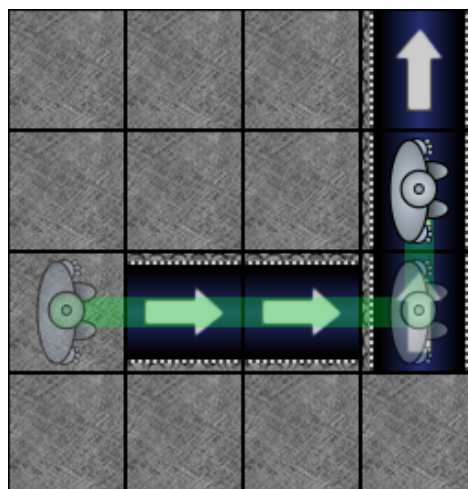
**Schieber:** Deprimierend, ständig wird man durch die Gegend geschoben. Bleibt man vor dem Schieber stehen, schiebt er den Roboter ein Feld vorwärts.



**Schrottpresse:** Zwischen den beiden Schiebern einer Schrottpresse sollte man nicht stehen bleiben, sonst landet man als kleiner zusammengepresster Würfel auf dem Müll.

Förderbänder haben noch eine Besonderheit, denn sie können, wenn ein Förderband den Roboter auf ein anderes Förderband schiebt, auch die Blickrichtung des Roboters ändern. In allen anderen Fällen bleibt die Blickrichtung des Roboters erhalten.

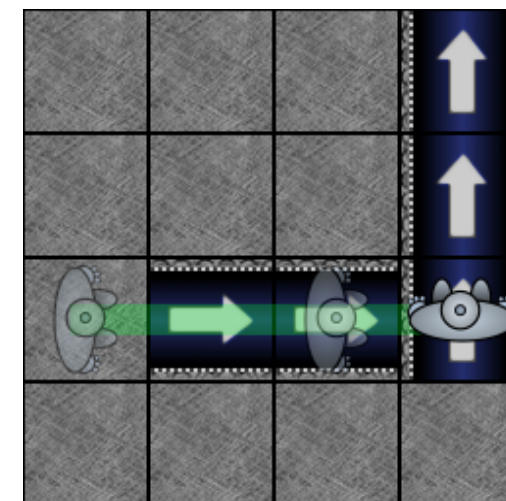
Als Beispiel: Der Roboter bewegt sich von seiner Startposition drei Schritte nach rechts und wird dann durch das Förderband nur nach oben geschoben, die Blickrichtung bleibt nach rechts.



Fall 1: Der Roboter behält die Blickrichtung bei.

Im anderen Fall bewegt sich der Roboter nur zwei Felder vor und wird dann vom Förderband nach

rechts auf das nach oben laufende Förderband geschoben. Dabei wird er auch gedreht, so dass er ebenfalls nach oben blickt.



Fall 2: Der Roboter wird gedreht.

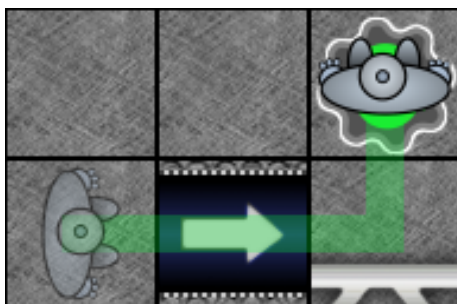
Die Elemente, die den Roboter auf dem Feld bewegen (Drehrad, Förderband, Schieber, Schrottpresse), agieren erst, wenn der Roboter nach einem Kartenzug zum Stillstand kommt. Durch eine Schrottpresse zu fahren, ist also kein Problem. Über ein Loch fahren zu wollen schon eher.

Zusätzlich agieren diese bewegenden Elemente nacheinander und nicht gleichzeitig. Die Reihenfolge ist:

#### Reihenfolge der Elementaktionen

Reihenfolge	Aktion
1	Die Förderbänder bewegen sich.
2	Der Schieber und die Schrottpresse werden aktiviert.
3	Die Drehräder drehen sich.

Es ist also möglich, dass ein Roboter, der auf einem Förderband steht, von diesem zu einem Schieber geschoben wird, der einen auf ein Drehrad-Feld schiebt und dieses dann den Roboter noch einmal dreht.



Diese Bewegung kann man mit nur einem Schritt erreichen.

### Spielumsetzung

Aufgabe der Teilnehmer ist es, eine KI zu programmieren, die aus acht Karten fünf auswählt und diese abspeichert. Als Eingabe muss die KI dafür folgende Dateien auslesen:

- **board.txt** – das Spielbrett mit ASCII-Zeichen
- **cards.txt** – acht Spielkarten zur Auswahl
- **bot.txt** – Roboterposition und -blickrichtung

Die fünf ausgewählten Karten müssen in der Datei **ccards.txt** (das doppelte „cc“ steht für „choose cards“) abgespeichert sein.

#### Die Spielbrettdatei board.txt

Die Datei **board.txt** enthält das Wichtigste überhaupt: das Spielbrett. Dessen Aufbau ist auch am kompliziertesten und muss hier genau erklärt werden.

```
12 12
V e ev^f eF
<N<<<<^ DH
b k LR c
>>>O>>>>>>>
b Rp ^ g
HL ivf^<<
i<<<L>>>>
je H^ c
<<<L bRc
b ^ ^DH<<c
^ ^ ^
i^k ^i i^Z
```

Listing 1: chop.dat

In der ersten Zeile findet man zwei durch ein Leerzeichen getrennte Ganzzahlwerte **X** und **Y**, die die Breite und die Höhe des Spielbrettes angeben. Danach folgen **Y** Zeilen mit je **X** Zeichen für das Spielbrett.

**Hinweis:** Die linke obere Ecke (in dem Fall also das **V**) hat immer den Index (0,0).

#### ASCII-Zeichen der Spielfelder

Es folgt die Bedeutung der einzelnen Zeichen.

Einfache Felder	
Zeichen	Feld
Leerzeichen	leerer Boden
H	Loch
O	Öl
S, T, U, V	Startfeld des Roboters (für die KI irrelevant, als leeres Bodenfeld behandeln)
Z	Zielfeld für den Roboter

Drehräder	
Zeichen	Feld
L	dreht den Roboter 90° gegen den Uhrzeigersinn
R	dreht den Roboter 90° im Uhrzeigersinn

Förderbänder	
Zeichen	Feld
<	Förderband nach links
>	Förderband nach rechts
^	Förderband nach oben
v	Förderband nach unten

Schieber/Presse	
Zeichen	Feld
C	schiebt den Roboter ein Feld nach links
D	schiebt den Roboter ein Feld nach rechts
E	schiebt den Roboter ein Feld nach oben
F	schiebt den Roboter ein Feld nach unten
M	Links/Rechts-Schrottpresse
N	Oben/Unten Schrottpresse

Wände	
Zeichen	Feld
a-p	definiert eine Wand/mehrere Wände pro Feld

Die Definition der Wände ist binärkodiert, dass heißt:

- Wand links = 1
- Wand rechts = 2
- Wand oben = 4
- Wand unten = 8



Beispiele:

Beispiele für Wände		
Kodierung	Zeichen	Wände
0	a	keine
1	b	links
8	i	unten
7	h	links, rechts und oben
15	p	an allen vier Seiten (entspricht einer Kiste)

### Die Kartendatei cards.txt

In jeder Zeile der Datei **cards.txt** steht eine Karte. Es gibt folgende Möglichkeiten:

- **MF 1** – bewegt den Roboter ein Feld vorwärts (vorwärts ist immer in Blickrichtung des Roboters)
- **MF 2** – bewegt den Roboter zwei Felder vorwärts
- **MF 3** – bewegt den Roboter drei Felder vorwärts
- **MB** – bewegt den Roboter ein Feld rückwärts
- **RL** – dreht den Roboter um 90 Grad nach links
- **RR** – dreht den Roboter um 90 Grad nach rechts
- **RU** – dreht den Roboter um 180 Grad

Im gleichen Format speichert man die fünf ausgewählten Karten auch in der Datei **ccards.txt** ab.

### Die Roboterdatei bot.txt

Die Datei **bot.txt** enthält nur eine Zeile mit den durch Leerzeichen getrennten Informationen:

- horizontale Position des Roboters (zwischen 0 und Breite des Spielbretts-1)
- vertikale Position des Roboters (zwischen 0 und Höhe des Spielbretts-1)
- Blickrichtung des Roboters:
  - **L** – nach links
  - **R** – nach rechts
  - **U** – nach oben („up“)
  - **D** – nach unten („down“)

Beispiel:

```
7 8 D
```

Der Roboter steht also auf Feld (7,8) und schaut (auf dem Bildschirm) nach unten.



Das Spielbrett im Großformat. 

### Weitere Dateien

#### Spielbrett in Großformat

Vor jeder Runde wird das Spielbrett noch in einem etwas schöneren ASCII-Format **ascii-board.txt** ausgegeben, das sich aber nicht zum Einlesen eignet, da es zu komplex ist. Man kann damit aber schöner nachvollziehen, wo sich der Roboter auf dem Spielbrett gerade befindet und wie die einzelnen Felder aussehen.

#### Sequenzdateien

Am Ende jeder Runde wird eine Datei **sequence.txt** erstellt, die die absoluten Bewegungen des Roboters für die fünf von der KI ausgewählten Karten auf dem Spielbrett enthält. Das heißt, es gibt folgende Einträge:

- **ML** – bewegt den Roboter um ein Feld nach links
- **MR** – bewegt den Roboter um ein Feld nach rechts
- **MU** – bewegt den Roboter um ein Feld nach oben
- **MD** – bewegt den Roboter um ein Feld nach unten
- **RL** – dreht den Roboter um 90 Grad nach links
- **RR** – dreht den Roboter um 90 Grad nach rechts
- **RU** – dreht den Roboter um 180 Grad
- **X** – Roboter wurde zerstört, Spiel ist zu Ende
- **Y** – Kartenstapel ist leer, Spiel ist zu Ende
- **Z** – Roboter hat das Ziel erreicht, Spiel ist zu Ende
- **--** – signalisieren die nächste Karte

Am Ende eines Spiels wird noch zusätzlich eine Datei **globalseq.txt** erstellt, die die gesamten Bewegungen des Roboters enthält. Auf diese Weise kann man genau nachvollziehen, wie der Roboter zum Ziel gefunden hat. Die einzelnen Runden werden dabei in der Datei durch acht Striche ----- voneinander getrennt.

## Der Wettbewerb

### Kategorien

Der Wettbewerb wird in vier Kategorien unterteilt, wobei die ersten drei eine Schwierigkeitsstufe angeben.

### Schwierigkeitsstufe leicht

In der einfachsten Stufe kann der Roboter nicht sterben, das heißt, das Spielbrett enthält keine Löcher und auch keine Schrottpresse. Auch die Ölfelder werden entfernt, da deren Auswirkungen recht komplex werden können. Zusätzlich ist das Spielbrett von einer unsichtbaren Wand umgeben, sodass der Roboter nicht herunterfällt.

### Schwierigkeitsstufe mittel

In der Standardstufe sind alle Felder wie oben beschrieben vorhanden, d.h. es gibt auch Löcher und Schrottpressen, bei denen der Roboter sterben kann. Der Roboter kann aber immer noch nicht vom Spielbrettrand fallen.

### Schwierigkeitsstufe schwer

In der schwierigsten Stufe muss der Roboter wirklich aufpassen, wo er hintritt, denn die unsichtbare Wand um das Spielbrett herum ist verschwunden. Außerhalb des Spielbretts herrscht

gähnende Leere, ein Schritt zu weit und der Roboter verschwindet auf Nimmerwiedersehen.

### Sonderpreis: 3-D-Spielbrett

Als kleine Besonderheit fänden wir es schön, wenn es ein 3-D-Spielbrett gäbe, in dem man den Roboter herumlaufen sieht. Als Anforderung muss die GUI neben der Spielbrett- und Roboteranzeige nur das komplette Sequenzskript **globalseq.txt** (siehe oben) laden und abspielen können.

**Hinweis:** 3-D schließt in diesem Fall auch die isometrische Darstellung [3] mit ein, die viele Spieler vielleicht aus Action-Rollenspielen wie Diablo oder Sacred kennen.

### Teilnahmebedingungen

Jeder Teilnehmer kann nur in einer der vier Kategorien antreten.

Zusätzlich muss es pro Kategorie (außer bei dem 3-D-Spielbrett) mindestens zwei Teilnehmer geben, damit wenigstens etwas Wettbewerb entsteht.

**freiesMagazin**-Teammitglieder sind von der Teilnahme ausgeschlossen, weil diese zum einen Zugriff auf alle andere KIs haben und zum anderen auch viel früher von diesem Wettbewerb erfahren haben.

### Qualität

Daneben gibt es noch die Anforderung, dass der Roboter in der jeweiligen Schwierigkeitsstufe, in der man teilnehmen möchte, in mindestens 75 % der Testfälle das Ziel erreichen muss. Das heißt,

eine schlechte KI, die den Roboter in jedes Loch stolpern lässt und ihn dadurch zerstört, kann nicht gewinnen. Die Einschränkung müssen wir machen, um ein gewisses KI-Niveau im Wettbewerb zu erreichen.

Das 3-D-Spielbrett sollte natürlich korrekt funktionieren und „gut aussehen“.

### Programmiersprache

Die benutzte Programmiersprache ist wie immer freigestellt. Das Projekt sollte aber auf einem „Standard-Linux-Rechner“ (d. h. mit einer aktuellen Linux-Distribution) ohne Probleme kompilieren und ausführbar sein.

Die Programme müssen im Quelltext eingereicht werden, der unter einer Open-Source-Lizenz [4] veröffentlicht werden muss. Hier helfen wir gerne bei der Auswahl einer passenden Lizenz weiter, sollten Fragen zu diesem Thema auftauchen.

Für die KIs ist es natürlich erlaubt, den Quelltext und die Bibliotheken der Referenz-Implementierung (siehe unten) unter Beachtung der Lizenz zu nutzen, um auf dieser Basis eine eigene KI zu erstellen.

Sinnvoll ist es, wenn das KI-Programm nach der Ausführung im Erfolgsfall 0 zurückliefert und bei Fehlern einen Wert ungleich 0.

### Laufzeit

Die Laufzeit der KI-Programme sollte natürlich so gering wie möglich sein. Für die Teilnahme am Wettbewerb sollten im Schnitt pro Spiel nicht mehr als maximal 6 Sekunden gebraucht werden.

Das heißt ein kompletter Testdurchgang mit 100 Kartenstapeln auf einem Spielbrett sollte maximal 10 Minuten dauern.

### Preisgelder

Die vier Kategorien sind jeweils mit einem Buchgutschein in folgender Höhe dotiert:

Preisgelder	
Stufe	Preisgeld
leicht	10 Euro
normal	20 Euro
schwer	30 Euro
3-D-Spielbrett	50 Euro

### Gewinner

Die einzelnen KIs treten auf verschiedenen (bis nach Ende des Wettbewerbs geheimen) Spielbrettern mit zuvor berechneten Kartenstapeln an. In die Bewertung fließt zunächst ein, ob der Roboter das Ziel erreicht hat oder vorher zerstört wurde. Erreicht er das Ziel, fließt noch die Anzahl der Züge bzw. Runden in die Bewertung mit ein. Es ist also sehr positiv, wenn ein Roboter in möglichst wenigen Runden das Ziel sicher findet.

Bei dem 3-D-Spielbrett ist die Wahl des Gewinners etwas schwieriger, da Aussehen immer im Auge des Betrachters liegt. Wir versuchen aber die Einsendungen objektiv zu bewerten. Der Teilnehmer mit dem schönsten 3-D-Spielbrett gewinnt.

### Wettbewerbsdauer

Der Wettbewerb begann am 27. September 2009 und endet am 13. Dezember 2009, so dass wir den Gewinnern hoffentlich noch zu

Weihnachten ein kleines Geschenk machen können. Alle Einsendungen sollten per E-Mail bis zu diesem Stichtag bei der Redaktion unter [redaktion@freiesMagazin.de](mailto:redaktion@freiesMagazin.de) eingegangen sein. Programme, die uns später erreichen, können leider nicht mehr beachtet werden.

Es ist erlaubt, dass ein Teilnehmer seine eingesandte KI mehrfach ausbessert. Wir geben gerne Hinweise auf etwaige Fehler der KI (bezogen auf den Quellcode, nicht auf falsche Entscheidungen). Informationen über die Leistung der eigenen KI im Vergleich zu anderen Teilnehmern werden aber nicht vorab veröffentlicht.

### Referenzimplementierung

Um seine eigene KI testen zu können, benötigt man die Dateien, die später den Wettbewerb verwalten werden. Das gesamte Paket kann als Tar-Archiv [5] heruntergeladen werden und muss nach dem Entpacken mit

```
$ make
```

kompiliert werden.

Das Paket setzt sich aus verschiedenen Bestandteilen zusammen, wovon ein Großteil in C++ programmiert ist. Das Paket **build-essential** und dessen Abhängigkeiten sollten auf alle Fälle installiert sein, um die Programme kompilieren zu können.

### Spiele-Engine

Die Dateien der Engine, das heißt der Verwaltung, die die KI ausführt und das Schreiben und

Lesen der Spieldateien übernimmt, nimmt den größten Teil des Archives ein.

Im Verzeichnis **robots-engine\_dir** findet man die Bibliotheken in Ordnern, die nach Basis (**libbase**), Roboter (**libbot**), Karten (**libcards**) und Spielfelder (**libtiles**) aufgeteilt sind.

Der Quellcode unterliegt der GNU Lesser General Public License (LGPL) [6], damit eine Verwendung der erstellten Bibliotheken erlaubt ist, ohne dass man den eigenen Quellcode wieder unter der LGPL veröffentlichen muss. Das heißt, eine KI, die auf **libbot** zugreift, um die Roboter-Dateien auszulesen, darf beispielsweise unter der BSD-Lizenz vertrieben werden. Würde der Quellcode unter der GPL veröffentlicht werden, hätte das Endprodukt, das die Bibliothek nutzt, auch unter der GPL veröffentlicht werden müssen, was eine zu große Einschränkung wäre.

Wer natürlich Teile des Quellcodes aus der Engine benutzen und abändern oder ergänzen möchte, kann dies auch tun, ist dann aber an eine Veröffentlichung des Gesamtwerkes unter der LGPL gebunden.

Nach der Erstellung mittels **make** findet man die Datei **robots-engine** im Hauptordner, die man wie folgt aufrufen kann:

```
$ ./robots-engine save DECKNAME
```

Dies erstellt einen zufälligen Kartenstapel und speichert diesen als Datei **DECKNAME** ab. Wer gleich mehrere Kartenstapel erstellen möchte, kann dies mit einem einfachen Skript erledigen:

(dies liegt in leicht abgewandelter Form auch als Datei **create\_decks** vor):

```
#!/bin/bash
mkdir -p decks

# create decks 0 - 99
for (( I=0; $I <= 99; I++ ))
do
    echo "./robots-engine save ↻
decks/carddeck$I.dat"
    ./robots-engine save decks/↻
carddeck$I.dat
    sleep 2
done
```

Listing 2: *create\_decks*

In dem Fall wird erst ein Verzeichnis **decks** erstellt und dann 100 Kartenstapel darin gespeichert. Die zwei Sekunden Wartezeit sind dafür da, dass die Zufallszahlen (die von der Uhrzeit abhängig sind) neu berechnet werden.

Im Ordner **gameboards** findet man das Referenz-Spielbrett **chop.dat**. Man kann nun entweder dieses Spielbrett alleine laden, sodass ein zufälliger neuer Kartenstapel erzeugt wird oder man benutzt einen der über obiges Vorgehen gespeicherten Kartenstapel:

```
$ ./robots-engine load ↻
BOARDNAME [easy|normal|hard]
$ ./robots-engine load ↻
BOARDNAME DECKNAME [easy|normal↻
|hard]
```

Als letztes Argument kann man optional (daher die eckigen Klammern) auch die Spielstufe mit angeben. Standardwert ist „normal“. Ein Beispiel:

```
$ ./robots-engine load ↻
gameboards/chop.dat decks/↻
carddeck0.dat easy
```

Wenn man das Spielbrett und Kartenstapel geladen hat, kann man eine Runde (nicht das gesamte Spiel!) per

```
$ ./robots-engine start [easy|↻
normal|hard]
```

laufen lassen. Die Spielstufe sollte dabei am besten mit der bei **load** angegebenen übereinstimmen.

Nach dem Aufruf wird intern das im Ordner liegende Skript **bot-ki** gestartet:

```
#!/bin/bash

# Reference
./KIs/robots-ki-reference $1

retVal=$?

exit $retVal
```

Listing 3: *bot-ki*

Dies ruft also einfach nur die jeweilige KI (hier die Referenz-KI) im Ordner **KIs** auf. Als Argument **\$1** wird dabei die Spielstufe übergeben. Möchte

man mit dem eigenen Bot spielen, passt man einfach die Zeile

```
./KIs/robots-ki-reference $1
```

entsprechend den eigenen Wünschen an.

Hat man eine Runde über die Option **start** ausgeführt, findet man die oben erwähnte Datei **sequence.txt** im Ordner. In dieser findet man die Bewegungssequenz wieder.

Ein komplettes Spiel kann man durchführen, wenn man die Engine solange startet, bis der Bot am Ziel ist, zerstört wurde oder keine Karten mehr auf dem Kartenstapel liegen.

### Referenz-KI

Die Referenz-KI im Ordner **KIs/robots-ki\_dir** soll nur grob veranschaulichen, wie man die Bibliotheken der Spiele-Engine nutzen kann, um mit deren Hilfe eine KI aufzubauen.

Die KI selbst ist sehr dumm, denn sie wählt immer nur die ersten fünf der acht zur Auswahl stehenden Karten aus. Das sorgt dafür, dass der Roboter nur per Zufall ins Ziel findet.

Die Referenz-KI wird unter der GNU General Public License [7] veröffentlicht.

Es ist aber möglich, innerhalb von wenigen Stunden mit Hilfe der Bibliotheken eine KI zu schreiben, die auch auf der Stufe „schwer“ den Roboter sicher ins Ziel bewegt. Diese zweite Referenz-KI von Dominik Wagenführ wird aber erst nach Beendigung des Wettbewerbs veröffentlicht, sonst wäre die Teilnahme zu einfach.



## Grafische Oberfläche

Dominik Honnef hat für den Wettbewerb eine grafische Oberfläche (GUI) in Ruby geschrieben, mit der man die Bewegungen des Roboters schöner verfolgen kann.

Aufgrund der Größe des Artikels, haben wir die Anleitung in den extra Artikel „Die GUI zum Programmierwettbewerb“ auf [Seite 51](#) ausgelagert.

## Wettbewerb starten

Wer den Wettbewerb starten will, kann das Skript `start_contest` benutzen:

```
#!/bin/bash
GAMES=$1
DECKS='find decks -name "*.dat" | \
sort '
LEVEL=$2

date
for GAME in $GAMES
do
    for DECK in $DECKS
    do
        echo "./robots-contest \
$GAME $DECK $LEVEL"
        ./robots-contest $GAME \
$DECK $LEVEL
    done
done
date

exit 0
```

Listing 4: `start_contest`

Der Aufruf ist wie folgt:

```
$ ./start_contest GAMEBOARDS [ \
easy | normal | hard]
```

**GAMEBOARDS** ist dabei eine Auflistung aller Spielbretter, die benutzt werden sollen. Als Kartenstapel werden alle Dateien im Ordner **decks** benutzt (den zum Beispiel das Skript `create_decks` zuvor angelegt hat). Ein Beispiel:

```
$ ./start_contest "gameboards/ \
chop.dat gameboards/board2.dat" \
hard
```

## Erweiterungen

Da der Wettbewerb so offen wie möglich gehalten wird, kann natürlich auch jeder Nicht-Teilnehmer etwas beitragen, wenn er möchte. Es werden hier einige Möglichkeiten vorgestellt, wie man etwas beitragen kann.

## Themen

Im Ordner **images** findet man die Themen, die für die Spielbrettanzeige genutzt werden können. Wer selbst ein neues Thema erstellen möchte, legt einfach ein neues Verzeichnis in diesem Ordner an und speichert die selbst erstellten Bilder der 31 Felder plus jeweils ein Bild für den Roboter und den Cursor dort ab. Die Felder müssen dabei eine Größe von 64x64 Pixeln haben. Der Roboter und der Cursor müssen einen transparenten Hintergrund haben.

Das neue Thema wird automatisch vom GUI-Starter erkannt und kann dort ausgewählt werden.

## Spielbretter

Es ist relativ leicht, neue Spielbretter zu erstellen. Einfach eine Textdatei anlegen, in die erste Zeile die Breite und Höhe des Spielbretts eintragen und darunter dann die einzelnen Spielfelder mit obigem ASCII-Code angeben. Bis auf die Wände sind die Codes für alle Felder ziemlich leicht zu erlernen.

Ein Spielbrett sollte (für die GUI-Anzeige) nicht größer als 12x12 Felder sein, da die GUI auf diese Größe beschränkt ist.

## Spielbrett-Editor

Wer Lust hat, kann auf Basis der Ruby-GUI oder als Eigenentwicklung einen Spielbrett-Editor schreiben, in welchem man per Drag & Drop die Felder auf ein Spielbrett setzen kann und dieser das Ergebnis dann als ASCII-Datei nach obigem Schema abspeichert.

Es gibt bereits einen Spielfeldeditor für das Originalspiel RoboRally [8], der aber für Windows geschrieben wurde. Damit lassen sich aber auch über Wine neue Spielfelder erstellen und ausdrucken.

Eine Schnittstelle für den Wettbewerb gibt es aber nicht und der Editor ist auch nicht Open Source.

## Spielfelder

Neben den obigen Spielfeldern kann man sich natürlich noch zahlreiche neue Felder ausdenken, deren Funktion man in die Engine



einprogrammieren müsste. Als Beispiele seien genannt:

- Teleporter: Teleportiert einen Roboter von einem Feld zu einem anderen.
- Fallgrube: Öffnet zu bestimmten Zeiten (z. B. jede 2 und 4 Phase einer Runde) ein Loch im Boden.
- Schiebetür: Öffnet und schließt sich nur zu bestimmten Phasen.
- Laser: Wie im Originalspiel RoboRally nimmt ein Roboter Schaden, wenn er im Strahl eines Lasers stehen bleibt.
- Reparaturkit: Wie im Originalspiel, auf dem der Roboter wieder repariert wird.
- Einbahnstraße: Der Roboter kann nur in diese Richtung ein Feld betreten und verlassen.

Der Phantasie sind keine Grenzen gesetzt.

Eine sehr gute Auflistung weiterer Spielfelder findet man auf den Seiten WarpZone [9] und RoboFactory [10]. Dort sind auch gleich zahlreiche Spielbretter abgelegt, die diese neuen Elemente nutzen.

## Schlussbemerkungen

Wir wünschen allen Teilnehmern viel Erfolg und freuen uns auf zahlreiche Einsendungen. „*Let the games begin . . .*“

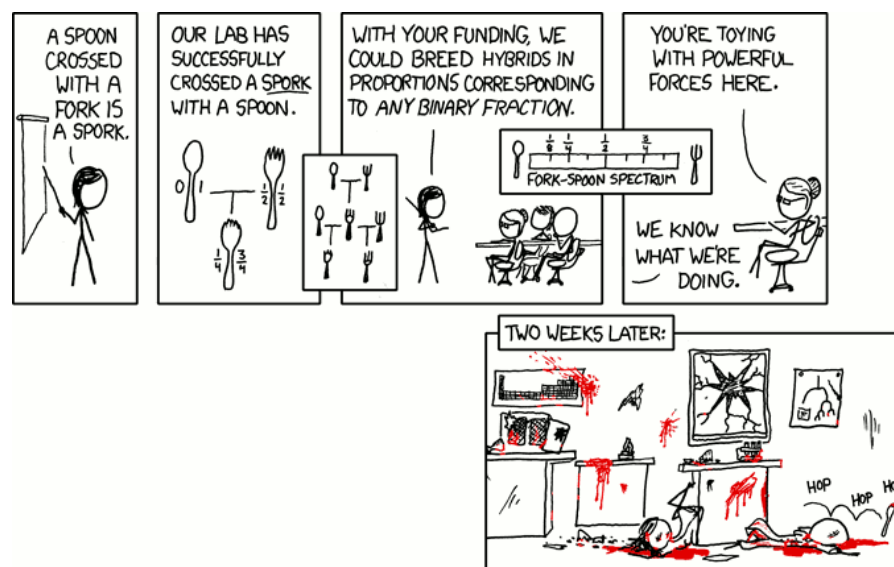
**Quellcode: per FTP oder per HTTP.**

## LINKS

- [1] <http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet>

- [2] <http://www.wizards.com/default.asp?x=ah/prod/roborally>
- [3] [http://de.wikipedia.org/wiki/Isometrische\\_Darstellung](http://de.wikipedia.org/wiki/Isometrische_Darstellung)
- [4] <http://www.opensource.org/licenses/category>
- [5] <http://freiesmagazin.de/ftp/2009/freiesMagazin-2009-10-wettbewerb.tar.gz>
- [6] <http://www.gnu.org/licenses/lgpl.html>
- [7] <http://www.gnu.org/licenses/gpl.html>
- [8] <http://www.roborally-editor.net/german/>
- [9] <http://www.randomdice.com/games/roborally/rboardelement.asp>
- [10] <http://www.robofactory.de/pages/content/rules.htm>

Diesen Artikel kommentieren



„Forks and Spoons“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/419>

## Die GUI zum Programmierwettbewerb

**D**a beim letzten Programmierwettbewerb von freiesMagazin im April dieses Jahres [1] einige Nutzer nach einer grafischen Oberfläche (GUI) fragten, um das Geschehen besser verfolgen zu können, hat Dominik Honnef diese für den anstehenden Wettbewerb (siehe „Erklärungen zum zweiten Programmierwettbewerb“ auf Seite 42) umgesetzt. Die GUI ist dabei als Programmier- und Visualisierungshilfe gedacht und wird nicht zwingend für eine Teilnahme benötigt.

### Anforderungen

Für die Umsetzung wird Ruby [2], Ruby-Gnome2 [3], RubyGems [4] und die Gosu Game Development Library [5] benutzt. Daher muss man ein paar Ruby-Pakete (und deren Abhängigkeiten) installieren:

- **ruby** bzw. **ruby1.8**
- **rubygems**
- **libgtk2-ruby** bzw. **libgtk2-ruby1.8**

Daneben benötigt man noch einige Entwickler-Bibliotheken, um die Gosu GUI kompilieren zu können:

- **g++**
- **libgl1-mesa-dev**
- **libpango1.0-dev**
- **libboost-dev**
- **libsdl-mixer1.2-dev**
- **ruby1.8-dev**

**Hinweis:** Die Paketbezeichnungen stammen von Debian/Ubuntu und können/werden unter anderen Distributionen variieren. Unter Arch Linux gibt es beispielsweise keine extra dev-Pakete. Hier hilft ein Blick in die offizielle Installationsanleitung [6] weiter.

Nach der Installation der Pakete installiert man die Gosu GUI Library mit Root-Rechten über

```
# gem install gosu
```

Falls gefragt wird, welches gosu-Paket installiert werden soll, wählt man am besten **gosu 0.7.14 (ruby)** aus. Falls eine Fehlermeldung der Art

```
ERROR: While executing gem ... ⤵
(Gem::GemNotFoundException)
  Could not find gosu (> 0) ⤵
  in any repository
```

kommt, sollte man die Installation einfach wiederholen, bis das Paket gosu gefunden wird.

Nach der Kompilierung von Gosu kann man die Entwicklerpakete wieder deinstallieren. Ruby und Co. müssen natürlich installiert bleiben.

### GUI-Start

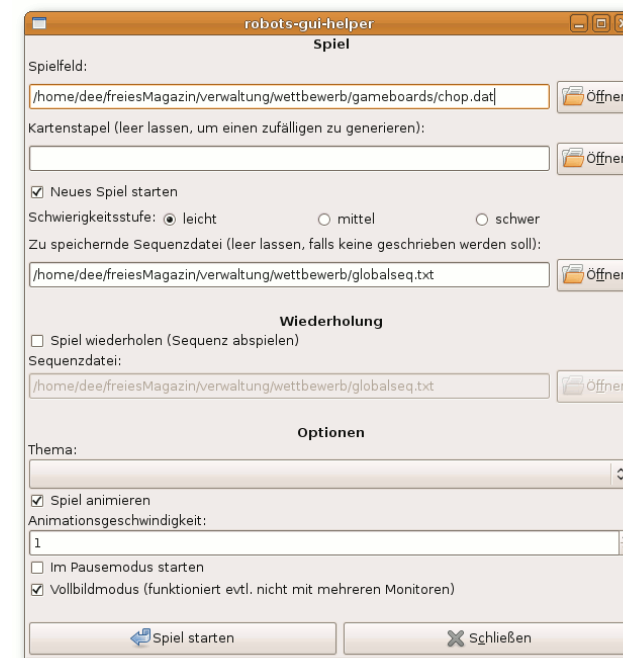
Die grafische Oberfläche besteht aus zwei Teilen: zum einen aus der Oberfläche für alle Einstellungen des Spiels „robots-gui-helper“ und zum anderen aus der Oberfläche für das Spielbrett „robots-gui“. Da die „robots-gui“ nie direkt gestartet wer-

den sollte, werden deren Optionen hier nicht näher erläutert. Den GUI-Starter ruft man per

```
$ ./robots-gui-helper
```

auf. Im Standardfall kann man nun bereits auf „Spiel starten“ klicken und die Referenz-KI sollte den Roboter bewegen.

Die Einstellungen im Starter sind fast alle selbst-erklärend. Es soll hier nur auf die Optionen eingegangen werden, die nicht offensichtlich sind.



Das Hilfsprogramm „GUI-Starter“ zur Vorbereitung der GUI. 🔍

Über „*Neues Spiel starten*“ kann man einstellen, ob der Roboter von seiner letzten Position aus der Datei `bot.txt` oder von der Startposition auf dem Spielfeld starten soll. Zusätzlich wird das Spielfeld und der Kartenstapel nicht neu eingelesen, wenn der Haken nicht gesetzt ist. Auf diese Art kann man unterbrochene Spiele fortsetzen.


Entfernt man den Haken bei „*Spiel animieren*“, läuft der Roboter nicht über das Spielfeld, sondern das Spiel wird bis zum Ende berechnet, der Roboter also das Ziel gefunden hat oder zerstört wurde. Erst dann wird die Anzeige freigeschaltet und man sieht den Roboter an der Endposition.

Mit „*Spiel starten*“ startet man das Spiel und kann dem Roboter bei der Bewegung zuschauen. Dieser hinterlässt eine transparente grüne Spur (Nein, er verliert kein Öl!), mit der man sehen kann, welche Felder er wie häufig betreten hat. Mit den Tasten `+` und `-` kann man die Bewegungsgeschwindigkeit des Roboters verändern.



Ein Testlauf mitsamt Spur des Roboters. 



Eine gute KI würde wohl eher diesen Weg wählen. 

Über einen Klick auf den Eintrag „*Pause*“ kann man das Spiel kurz unterbrechen, „*Neustart*“ startet das Spiel von vorne (bei einem zufälligem Kartenstapel wird ein neuer zufälliger ausgewählt). Mit „*Beenden*“ verlässt man die GUI und landet wieder im GUI-Starter.

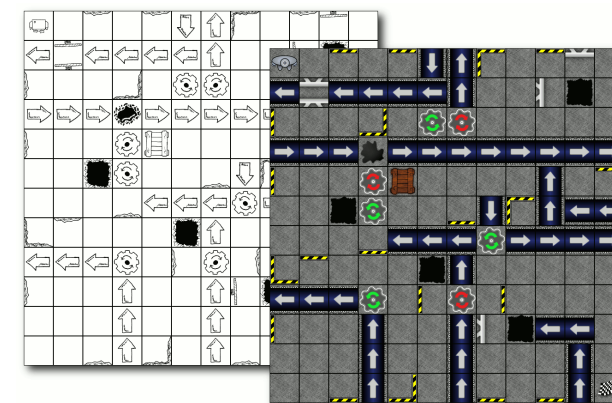
**Hinweis:** Die GUI hat eine Größe von 1024x768 Pixeln. Das heißt, um sie sinnvoll nutzen zu können, sollte die Bildschirmauflösung größer als 1024x768 sein oder man muss im GUI-Starter einstellen, dass das Programm im „*Vollbildmodus*“ starten soll. Dann werden die Grafiken auf eine kleinere Auflösung herunterskaliert.

## Die Spielbrett-Themen

In der GUI gibt es (derzeit) zwei Themen zur Auswahl.

Das Standardthema im Ordner `images/new` wurde von Arne Weinberg erstellt. Die Bilder unterliegen der GFDL [7].

Alternativ gibt es noch eine handgemalte Version im Ordner `images/pen`. Diese wurde von Dominik Wagenführ erstellt. Die Bilder unterliegen der CreativeCommons-Lizenz „*Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland*“ [8].



Die beiden vorinstallierten Themen. 

## LINKS

- [1] <http://www.freiesmagazin.de/20090401-programmierwettbewerb-gestartet>
- [2] <http://www.ruby-lang.org/de/>
- [3] <http://ruby-gnome2.sourceforge.jp/de/>
- [4] <http://gems.rubyforge.org/> 
- [5] <http://www.libgosu.org/> 
- [6] <http://ruby-gnome2.sourceforge.jp/hiki.cgi?Install+Guide> 
- [7] <http://www.gnu.org/licenses/fdl.html> 
- [8] <http://creativecommons.org/licenses/by-sa/3.0/de/>

Diesen Artikel kommentieren 



## FAQ zum Programmierwettbewerb

**D**a der zweite Programmierwettbewerb vor einer Woche, am 27. September 2009, startete [1], gab es in der Zeit schon zahlreiche Fragen, weil die Anleitung an einigen Stellen nicht detailliert genug ist. Aus dem Grund gibt es bereits jetzt eine Liste mit den Fragen und Antworten, da der Artikel im Magazin nicht so kurzfristig geändert werden konnte.

**Hinweis:** Die Webseite mit den Details zum Wettbewerb wurde bereits entsprechend geändert [2]. Es wurde aber auch eine extra Beitrag mit diesen Fragen verfasst, der ständig aktualisiert wird [3].

### Fragen zu Spielfeldern

❖ **Wie wahrscheinlich treten die einzelnen Elemente auf? Werden diese zufällig erstellt?**

❖ Die Spielbretter sind alle handgemacht, daher gibt es keine Wahrscheinlichkeit, mit der ein bestimmtes Spielfeld auftaucht.

❖ **Verhält sich ein Schieber, der in Richtung einer direkt am Schieberfeld angrenzenden Wand zeigt, wie eine Schrottpresse?**

❖ Nein, denn der Bot ist maximal ein halbes Feld breit. Ein Schieber geht genau bis zur Hälfte eines Feldes. Das heißt, wenn der Schieber komplett ausgefahren ist, dotzt der Bot leicht gegen die Wand, aber es passiert nicht Schlimmes. Wenn keine Wand da ist, reicht der Schwung aber aus, damit er auf das nächste Feld rutscht.

❖ **Kann man eine Schrottpresse oder einen Schieber von allen vier Seiten betreten?**

❖ Nein, eine Schrottpresse und auch die Schieber haben an der jeweiligen Seite, an der der Schieber steht, eine Wand und können von dort nicht befahren oder verlassen werden.

❖ **Wird man beim Rutsch durch ein Ölfeld auf das am letzten Ölfeld angrenzende Feld gesetzt oder bleibt man auf dem letzten Ölspurfeld stehen?**

❖ Man landet auf dem letzten Feld nach der Ölspur (soweit kein anderes Objekt einen vorher stoppt).

❖ **Falls zwei Förderbänder in der Richtung aufeinander zeigen und man wird von einem auf das andere Band geschoben, wird man dann um 180° gedreht?**

❖ Nein, wird man nicht. Das mit der Drehung auf den Förderbändern kommt daher, dass diese „untrennbar“ verbunden sind und eigentlich grafisch einen Bogen beschreiben sollte. Das war aber zu umständlich zu realisieren. Stellt man es sich in real vor, wird man auf einem Gepäckförderband auch gedreht, wenn es um die Kurve geht. Aber man wird sicherlich nicht gedreht, wenn man auf einem Förderband steht, was gegen ein anderes läuft. Da fällt man maximal um.

❖ **Per Förderband gelangt man auf ein Ölfeld, nach dessen Durchrutschen man auf ein wei-**

**teres, aber andere Richtung zeigendes Förderband gelangt: Wird man dann auch gedreht?**

❖ Ja, derzeit wird der Bot auch gedreht, wenn er von einem Fließband über eine Ölspur auf ein anderes Fließband geschoben wird. Das klingt etwas ungewöhnlich, aber ist so umgesetzt

❖ **Wie groß werden die Spielfelder sein? Signifikant größer als die chop.dat?**

❖ Die Wettbewerbsspielfelder werden voraussichtlich alle eine Größe von 12x12 Felder haben.

❖ **Es gibt das Leerzeichen für ein leeres Feld und es gibt das Zeichen a für ein Feld ohne Wände. Stellen diese zwei Zeichen das gleiche dar oder gibt es irgend einen Unterschied zwischen diesen Feldern?**

❖ Nein, die Felder sind identisch. Der Grund ist einfach, dass die erste (linke) Wand nicht bei **b** anfangen sollte. Wenn also ein **a** gelesen wird, ist das wie ein leeres Feld.

❖ **Wenn der Roboter das erste Feld eines langen Förderbandes betritt, wird er dann bis ganz ans Ende geschoben?**

❖ Nein, der Roboter wird pro Phase (also nach dem Ausspielen einer einzelnen Karte) nur immer ein Feld auf dem Förderband geschoben und nicht bis ans Ende. Das hat vor allem den Grund, dass es auch Förderbänder geben kann,

die im Kreis fahren. Der Roboter würde so nie mehr stillstehen bleiben.

❖ **Wenn der Roboter auf eine Wand zufährt, verfallen dann überflüssige Bewegungen oder ist der Zug dann einfach nicht möglich?**

❖ Wenn der Roboter ein Feld vor einer Wand steht und über die Karte **MF 3** drei Schritte vorwärts fahren soll, fährt er einen Schritt und die zwei anderen verfallen.

❖ **Wenn man direkt vor ein Ölspur steht und drei Felder mit MF 3 nach vorn geht, wo landet man?**

❖ Man macht einen Schritt, rutscht bis auf das erste Feld nach der Ölspur und geht dann noch zwei Schritte weiter.

## Fragen zur Schnittstelle

❖ **Ist es erlaubt, dass der Bot während des Spiel Dateien schreibt, um sich bisschen was zu merken?**

❖ An Dateien darf man abspeichern, was man möchte, natürlich.

❖ **Wenn schon nur 6 Sekunden pro Spiel zur Verfügung stehen, gibt's dann eventuell ein wenig Vorlaufzeit für jedes neue Spielfeld?**

❖ Also 6 Sekunden sind schon sehr großzügig – natürlich auf dem späteren Testrechner. Wer seine KI auf einem 486er schreibt, wird wohl auch mit 20 Sekunden Ausführungszeit am Wettbewerb teilnehmen können.

❖ **In der Datei cards.txt scheinen nicht nur die 8 Karten zu stehen, aus denen man auswählen kann, sondern der ganze restliche Stapel. Darf die KI diese Information benutzen?**

❖ Genau genommen ist es sogar die Aufgabe der KI, die Datei **cards.txt** einzulesen. Aber: Wenn die KI dran ist, stehen in der Datei nicht mehr alle Karten des Stapels, sondern nur die 8, die zur Auswahl stehen. Es gibt daher für die KI keine Möglichkeit alle Karten komplett einzulesen – zumindest wenn die Engine keine Lücke hat.

## Fragen zur 3-D-Oberfläche

❖ **Inwiefern soll das 3-D-Spielbrett wirklich 3-D sein? Ist damit nur isometrische Darstellung gedacht?**

❖ Es ist sowohl richtiges 3-D (also mit frei beweglicher Kamera), als auch 3-D in isometrischer Ansicht sowie 2-D mit isometrischer Ansicht gemeint.

❖ **Reicht eine isometrische Darstellung aus oder soll noch eine weitere (und wenn ja was für eine) Darstellung existieren?**

❖ Es wäre ggf. nicht falsch, wenn man alle vier Standardansichten der isometrischen Perspektive wählen kann oder gar frei rotieren und zoomen.

❖ **Soll die Kartengröße des 3-D-Spielbretts beliebig groß sein können?**

❖ Ja, es wäre gut, wenn zum Beispiel auch ein 100x100-Spielbrett dargestellt werden könnte.

❖ **Darf die 3-D-Darstellung voraussetzen, dass bestimmte Pakete installiert sind, wie z. B. die wxWidgets-Bibliothek oder verschiedene OpenGL-Bibliotheken?**

❖ Natürlich wird man für die grafische Darstellung auf vorhandene Bibliotheken zurückgreifen müssen. Die Mesa-Bibliotheken, wxWidgets 2.8 und andere unter neuen Linux-Distributionen vorhandenen Bibliotheken sind dabei kein Problem.

## Allgemeine Fragen

❖ **Ist der Wettbewerb nur für natürliche Personen oder kann man auch als Verein daran teilnehmen?**

❖ Es ist durchaus möglich, auch als Verein teilzunehmen, jedoch muss man sich den Preis dann natürlich untereinander aufteilen.

❖ **Wie kann man auf die Bibliotheken (also z. B. die libcards.a) in C zugreifen?**

❖ In C gar nicht, nur in C++, da die Library Klassen enthält und keine Funktionen. Am besten schaut man sich für eine Verwendung das Makefile der Referenz-KI im Ordner **KIs/robots-ki\_dir** an. Theoretisch muss man nur die Header-Dateien per **#include "NAME.hh"** in die jeweilige C++-Datei einbinden und dann beim Kompilieren die jeweilige Library mit angeben über

```
$ g++ -I./libbase -I./libcards -c main.cpp
$ g++ -o ki.bin main.o -L./libbase -L./libcards -lcards -lbase
```

❖ **Darf man außer Konkurrenz auch an den anderen Wettbewerbskategorien teilnehmen?**

❖ Da der Wettbewerb offen geführt und die spätere Wertung auch nicht geheim gehalten wird, kann jeder daheim bei sich selbst nachvollziehen, ob seine KI in einer anderen Kategorie besser

oder schlechter abschneidet. Eine Einbeziehung aller KIs, die nur außer Konkurrenz teilnehmen, würde den Testaufwand auf unserer Seite wahrscheinlich zu sehr erhöhen.

Wer seine 3-D-Oberfläche außer Konkurrenz einreichen will, kann dies aber gerne tun. Wir stellen diese dann vor, sie kann aber nicht gewinnen, wenn der Teilnehmer bereits in einer anderen Kategorie vertreten ist.

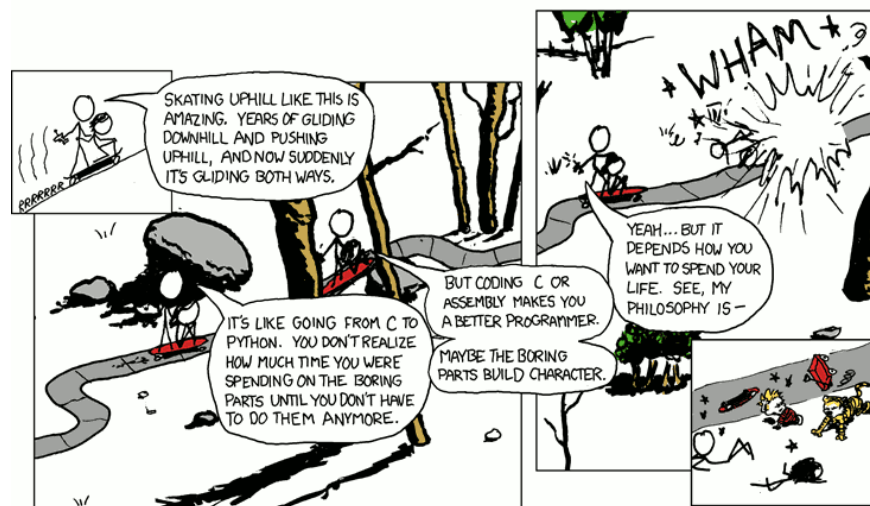
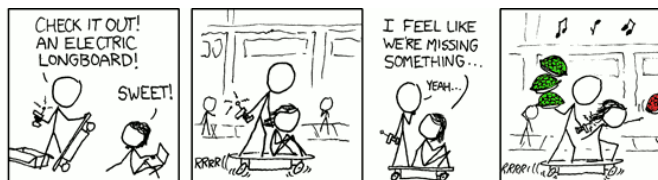
❖ **Muss das Ziel nur „überrollt“ werden oder muss die Bewegung direkt auf dem Ziel enden?**

❖ „Überrollen“ reicht. Sobald das Zielfeld einmal befahren wurde, endet das Spiel.

❖ **Beim Starten der GUI bekomme ich einen Fehler der Art „invalid multibyte char“. Was soll ich tun?**

❖ Die GUI setzt Ruby in der Version 1.8.x voraus, mindestens jedoch 1.8.6. – Mit Ruby 1.9 funktioniert die Anwendung nicht.

Diesen Artikel kommentieren 



„Electric Skateboard (Double Comic)“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/409>



## Leserbriefe

Für Leserbriefe steht unsere E-Mailadresse [redaktion@freiesMagazin.de](mailto: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

#### Frau gegen Mann

❖ Meine Meinung – als Mann – zur Frage der Geschlechtsneutralität ist, dass es ein nicht zu rechtfertigender Aufwand ist die LeserINNEN separat aufzuführen, wo doch die Intention so klar ist. Meiner Meinung nach ist dies auch grammatikalisch zu rechtfertigen, denn die männliche Form eines Nomens ist imho geschlechtsneutral deklariert.

**Martin Weinelt**

❖ In der aktuellen Ausgabe stößt es mir schon beim Lesen des Editorials etwas sauer auf. Ihr fragt, ob Ihr Eure Artikel geschlechtsneutral schreiben sollt.

Nun stelle ich mir die Frage, wo das denn bisher nicht eingehalten wurde.

Da ich fachlich sicher nicht ausreichend Wissen habe, um dies genau zu erklären bzw. zu begründen, verweise ich nur auf einen Artikel von Arthur Brühlmeier mit dem Titel „Sprachfeminismus in

der Sackgasse“ [1]. Herr Brühlmeier stellt darin sehr gut klar, wo der Unterschied zwischen biologischem Geschlecht und grammatikalischem Genus liegt.

So sind mit „Liebe Leser“ sowohl männliche als auch weibliche Leser korrekt angesprochen, da das Wort Leser zwar grammatikalische männlich ist, dies aber nicht auf das biologische Geschlecht des Lesers hinweist.

Ich empfehle die Lektüre des Artikels jedem, dem der Erhalt der deutschen Sprache am Herzen liegt und hoffe, ich konnte etwas zur Lösung der (meiner Meinung nach aus Fehlinformationen aufgestellten) Frage nach der Geschlechtsneutralität beitragen.

**Gordon Schultz**

❖ Ich bitte Sie hiermit, auch in Zukunft keine „geschlechterneutrale Sprache“ zu verwenden und verweise auf den Unterschied zwischen Genus, dem grammatischen Geschlecht, und Sexus, dem biologischen Geschlecht.

**Mathias Linkerhand**

❖ Ihr verwendet die Formulierung Leser allgemein und meint damit auch die Leserinnen. Wie wäre es, stattdessen stets die Formulierung Leserinnen zu verwenden und damit auch die männlichen Leser zu meinen? Es ist doch kein Geheimnis, dass zu wenig Frauen sich für Linux-Themen interessieren. Damit würde klargestellt, dass Frauen auf jeden Fall angesprochen wer-

den sollen. Der Gedanke scheint absurd? Dann lest mal die Satzung des Versorgungswerks der Rechtsanwälte in Berlin, denn dort wird es so gehandhabt, vgl. §4 „Die Bezeichnungen der Organe können auch in männlicher Form geführt werden“ [2].

**M. Schinagl**

❖ Ich möchte darauf hinweisen, dass die meisten deutschen Begriffe wie Leser, Fußgänger, Lehrer etc. androgyn sind. Sollten Sie Ihre Einstellung, die Sie zur Zeit vertreten, ändern, fügen Sie sich einem öffentlichen Druck, der gar nicht existiert und machen sich, ich möchte fast sagen ... lächerlich.

Genaugenommen ist „Leser“ geschlechterneutral, während „Leserinnen und Leser“ eine Wertung enthält. Die Person, von der die Beschwerde kam, meinte es zwar gut, kannte aber wohl diesen wundervollen Artikel nicht: „Sprachfeminismus in der Sackgasse“ [1].

**Frank**

❖ In Ihrer Ausgabe 09/2009 stellen Sie die Frage, ob ein Verzicht auf geschlechterneutrale Sprache Ihrem weiblichen Publikum zuzumuten sei. Als Teil dieser weiblichen Leserschaft lautet meine Antwort: Ja, unbedingt!

Und als Studentin der Germanistik und großer Liebhaberin der deutschen Sprache möchte ich dazu noch anmerken, dass diese geschlechterneutralen Formulierungen für mich ganz eindeutig in die Kategorie „Vergewaltigung der deut-





schen Sprache“ fallen. Nicht nur, dass es aus rein ästhetischer Sicht eine Zumutung ist, ein „LeserInnen“ im Fließtext zu betrachten oder gar visuell über ein unnötig angehängtes „und Leserinnen“ zu stolpern – ich halte diese Hervorhebung der weiblichen Form schlicht für unnötig. Nicht nur im Deutschen ist es Usus, bei der Ansprache einer geschlechtlich heterogenen Masse die männliche Form zu gebrauchen. Wenn der Feminismus auch in heutiger Zeit ernst genommen werden möchte, sollte er künftig von solchen albernen Forderungen absehen. **Franziska Pfeiffer**

❖ Ich will nur zwei „Dankeschöns“ (und Dankeschöninnen?) loswerden: Das erste betrifft Euer Magazin im allgemeinen und das zweite eure Entscheidung zu dem strittigen Thema. Endlich Menschen, die vernünftig mit der deutschen Sprache umgehen ...

Das erinnert mich an einen einfach (mal wieder) herrlichen Monolog meiner Deutschlehrerin. Wir hatten in unserer Klasse mal wieder ein paar Gastschüler aus Frankreich. Unsere Lehrerin betrat den Raum und meinte:

*„Guten Morgen, allerseits ... Oder sollte es heute nicht eher heißen: „Guten Morgen, liebe Schülerinnen und Schüler!“? Egal ... Ah, guten Morgen auch an euch ... Oder besser „... liebe Gästinnen und Gäste!“ Sagt mal, findet ihr es nicht auch zum (tschuldigung) Kotzen, diese ständige verbale feminine Emanzipation? Das geht zumindest mir gehörig auf die Nerven ...“*

**Sebastian Billaudelle**

❖ Mein erster Gedanke dazu war: „Ach nee, nicht Ihr auch noch“.

Ich finde, dies ist in Wirklichkeit nicht die Frage „Frau gegen Mann“ sondern „Leserlichkeit kontra femifaschistische Propaganda“. Umsonst werden solche Dinge wie z. B. [dieses Firefox-Plugin] [3], um Texte wieder leserlich zu machen, wohl kaum angeboten.

Außerdem kennt man ja mittlerweile die Auswüchse des Genderwahns und ich bin es schlicht leid, Dinge wie z. B. [dieses Bild] [4] ertragen zu müssen. Und das Binnen-I war der erste Schritt dazu. Principiis obsta!

Ich ersuche daher, im Sinne der Lesefreundlichkeit zu entscheiden. Zumindest drei Leser (mich und zwei Bekannte, die genauso denken) würdet Ihr im Falle einer gegenläufigen Entscheidung verlieren. **Christian Unterholzner**

❖ Gerade im Technikbereich fallen Frauen oft genug einfach unter den Tisch. Ich selbst erwartete die weibliche Form oft schon gar nicht mehr, ich werde sowieso übersehen. Als Magazin, das sich mit Freier Software beschäftigt, fände ich es umso schöner, wenn ein gesellschaftlicher Blick über den üblichen angeblich genderneutralen Ansatz im EDV-Bereich hinaus geworfen werden würde. Wenn Ihr Euch für eine Veränderung entscheiden würdet, würde ich mich sehr freuen!

**Michaela Blomberg**

❖ Bitte, bitte tut mir den Gefallen und lasst dieses gräßliche „Leser/-innen“ und so einen Krem-

pel weg. Es geht mir als Frau entsetzlich auf die Nerven.

Frauen sind zum größten Teil gleichberechtigt heutzutage und z. T. werden sie sogar bevorzugt behandelt. Wenn jemand dann auch noch auf so einen Schrott bestehen muss, frage ich mich wirklich, ob da überhaupt ein vernünftiges Selbstbewußtsein vorhanden ist.

Und ja, man hat mir schon mehrmals vorgeworfen, ich würde die Emanzipation kaputt machen und diejenigen haben das tatsächlich ernst gemeint! Ich muss sagen, wenn die Emanzipation von sowas abhängt, dann ist sie keine. Ich arbeite in einer reinen Männergruppe in der IT und bin dort genauso akzeptiert wie jeder von den Jungs. Das nenne ich Emanzipation, und zwar von beiden Geschlechtern. **Jessica Borchert**

❖ Die geschlechterneutrale Sprache nimmt mittlerweile derart seltsame Formen an, dass z. B. das US-amerikanische Militär bei Drohnen, also „unbemannten Flugkörpern“ nicht mehr wie ursprünglich von „Unmanned Aerial Vehicles“ spricht, sondern mittlerweile von „Unpopulated Aerial Vehicles“, denn es könnte ja ebensogut auch keine Frau darin sitzen ...

Ich persönlich halte Eure Herangehensweise für eine vernünftige – letztlich soll ja niemand willentlich und wissentlich verletzt werden. Aber wie viele der heute lebenden Frauen fühlen sich denn beispielsweise durch Literatur von Schiller, Goethe, Lessing o. Ä. verletzt – nur weil die im Traum



nicht daran gedacht haben, geschlechtsneutral zu schreiben?

Ich bin sicherlich nicht alleine mit der Meinung, dass es noch nie geschadet hat, die Kirche im Dorf zu lassen ... **Hans-Stefan Suhle**

❖ Als Mann fühle ich mich beim Lesen von Artikeln, deren Autoren unter Weglassung aller sprachlicher Harmonie so krampfhaft um Geschlechtsneutralität bemüht sind, oftmals abgegrenzt. Wenn also ein Artikel mit „Liebe Leser und Leserinnen“ beginnt, interpretiere ich das als eine Ansprache, die sich vorrangig an Frauen richtet, da außer der geschlechtsneutralen Anrede, nur die lesenden Frauen begrüßt werden. Mann fühlt sich unerwünscht und fehl am Platze.

Als Frau fühlte ich mich bei modernen, nach politischer Korrektheit strebenden Ansprachen allerdings vor den Kopf gestoßen: Bei Berufsbezeichnungen, die in alten Zeiten den Männern vorbehalten waren (Schreiner, Holzfäller, Schlosser etc.), ist der maskuline Genus ja sehr häufig zu finden. Würde ich als weiblicher Schreiner mit „Liebe Schreiner und Schreinerinnen“ angesprochen, klänge das in meinen Ohren so, als wolle man explizit darauf hinweisen, dass sich unter den Angesprochenen außer den „normalen“ Schreibern auch einige Frauen befinden – so, als wäre es etwas besonderes als Frau des Schreiners mächtig zu sein.

Beim Worte „Leserin“ hätte ich den Eindruck, da wolle jemand darauf hinweisen, ich sei, obwohl weiblich, doch lesekundig. **Ralf (Kommentar)**

❖ Ich möchte Sie zum Verzicht auf eine „geschlechtsneutrale“ bzw. genderkorrekte Textgestaltung beglückwünschen. Selbst journalistisch tätig empfinde ich Unsitten wie etwa das Binnen-I als Verunstaltung der deutschen Sprache die das Lesen von Texten unnötig erschwert. Ich kann nicht erkennen, dass die berechtigten Anliegen von Frauen damit nennenswert zu unterstützen wären. Trotzdem grassiert zur Zeit das Gender-Fieber. Letztens konnte ich einem Printmedium sogar schon gegenderte Adjektive lesen wie etwa „kundInnenfreundlich“!

Es bleibt zu hoffen, dass auch diese Mode vom Zahn der Zeit zermahlen werden wird und Sie bis [dahin] standhaft bleiben. **Harald Hackenberg**

❖ „Der Leser“ ist „geschlechtsneutral“, da es in der deutschen Sprache einen Unterschied zwischen Genus (grammatikalisches Geschlecht) und Sexus (biologisches Geschlecht) gibt. „Leser“ ist ein Maskulinum und daher ist der dazugehörige Artikel eben „der“. Das ist so, das war so, und das wird in absehbarer Zeit auch so bleiben, da können Feministen noch so sehr auf die Barrikaden gehen! :)

Die Redaktion macht mit ihrer Entscheidung meiner Meinung und der Grammatik nach alles richtig! **Dirk Sohler (Kommentar)**

❖ Ich würde es sehr begrüßen, wenn Ihr in Eurem Magazin beide Geschlechter (also die Leserinnen und Leser) berücksichtigen würdet. Meine Freundin zum Beispiel hat sich auch schon dar-

über aufgeregt, respektive sich ausgeschlossen gefühlt, als nur von den männlichen Personen die Rede war. **Sebastian Liechi**

❖ Ich kann Sie nur in aller Form bitten, diesen Unsinn nicht mitzumachen. Das ist einfach nur lächerlich und macht einen Text komplett unlesbar. In meinen Augen ist das eine Vergewaltigung der deutschen Sprache. **Frank Hemmerling**

❖ Vielen Dank für die zahlreichen Leserbriefe. Wie man sieht, sorgt das Thema für viel Diskussionsstoff, wobei die Meinung fast durchgehend (sowohl von unseren männlichen als auch weiblichen Lesern) in die gleiche Richtung geht. Eine Verunglimpfung durch die deutsche Sprache durch Binnen-Is wird es bei **freiesMagazin** in absehbarer Zeit nicht geben, ebensowenig wie die Einführung einer „geschlechterneutraleren“ Sprache, die Frauen und Männer extra erwähnt. Der Hinweis auf die Unterscheidung zwischen grammatikalischer und biologischer Bezeichnung gab es oft genug.

Davon unberührt bleibt natürlich die Verbindung zwischen der Bezeichnung und einer realen Person. Das heißt, wir werden natürlich auch in Zukunft von der Leserin Eva Mustermann sprechen, wenn diese direkt gemeint ist.

Interessant ist im Übrigen das Wort „Person“, das grammatikalisch weiblich ist, aber natürlich auch Männer mit einschließt. Darüber hat sich aber wohl noch kein Mensch beschwert. ;)

**Dominik Wagenführ**



## Duden Korrektor

❖ [Ich] kann bestätigen, dass der Duden Korrektor sehr nützlich ist und viel mehr Fehler erkennt als die integrierte Lösung. Ab Version 6.0 ist auch das Problem passé, dass der Computer bei der Korrektur enorm ausgebremst wird. Also auf jeden Fall zu dieser Version greifen, denn im Handel geistern auch noch viele 5.0-Versionen herum.

**Flo** (Kommentar)

## Ratespiel

„Kleiner, blecherner Schrotthaufen sucht in Großbaustelle Fahne fürs Leben. Was bin ich?“ [5]

❖ Ein alter Computer, ausgemistet, weil Win-Vista nicht mehr darauf laufen wollte, sucht am großen Schrottplatz nach einer Flasche Alkohol, um die Erinnerungen an die Jahre der Windows-Prostitution wegzuwaschen.

**Setsuna** (Kommentar)

❖ Könnte eine Bierdose sein? Fahne = Alkohol-fahne, kleiner Schrotthaufen = zerdrückte Bierdose.

**Daniela Wegert** (Kommentar)

❖ Bin ein bald krepierender alter Rechner von freiesMagazin, der zwischen den Druckfahnen von freiesMagazin nach dem ein oder anderen Fähnchen sucht – mit €-Zeichen drauf, um nicht bald doch zu verrecken ...

**christoph** (Kommentar)

❖ Wall-E [6]. :-) [Link von der Redaktion eingefügt].

**Mario** (Kommentar)

❖ Ein alkoholabhängiger Autofahrer.

**Jörg Zweier** (Kommentar)

❖ Der Roboter aus dem Film „Nr. 5 lebt“ [7] [Link von der Redaktion eingefügt] bei seiner Suche nach einem kleinen blechernen Freund auf einer Großbaustelle.

**sklicek** (Kommentar)

❖ Vielen Dank für die vielen Antworten. Interessant ist, dass der Alkohol eine große Rolle bei den meisten Lösungen spielt. Wir interpretieren hier aber nichts hinein. ;) Die beiden Beiträge mit „Wall-E“ bzw. „Nummer 5“ trafen dabei den Nagel fast auf den Kopf, denn es war natürlich der Protagonist des neuen freiesMagazin-Programmierwettbewerbs gemeint (siehe „Erklärungen zum zweiten Programmierwettbewerb“ auf Seite 42).

**Dominik Wagenführ**

## PovRay-Raytracing

❖ Beim Betrachten des Bildes von Gilles Tran könnte einem glatt der Kitt aus der Brille fliegen, wenn man denn eine tragen würde. (Wie viel Arbeitszeit muss wohl ein versierter PovRay-Kenner investieren, um ein annähernd vergleichbares Ergebnis zu erzielen?) Marcel Jakobs, herzlichen Dank für den interessanten Bericht, der auch ein Beispiel beinhaltet, das zu allem Überfluss auch noch auf Anhieb funktioniert und so in der Lage ist, dem Anfänger ein kleines Erfolgserlebnis zu liefern.

**Stefan** (Kommentar)

❖ Erstmal vielen Dank für das dicke Lob. Ich finde das Bild von Gilles Tran auch fantastisch. Unglaublich aber wahr: fast alle Objekte auf dem

Bild können auf seiner Seite [8] herunter geladen und in eigenen (privaten und auch kommerziellen) Bildern verwendet werden.

Generell finde ich es sehr praktisch, wenn man sich die Objekte, die man erstellt hat, in einer Datei speichert und nur noch in die Szenen einbindet (so wie man es im Beispiel mit der Datei colors.inc mit den vordefinierten Farben sehen kann). Dadurch kann man einmal definierte Objekte sehr leicht wieder verwenden. Man braucht sie lediglich noch skalieren, drehen und verschieben.

**Marcel Jakobs**

## LINKS

- [1] <http://www.bruehlmeier.info/sprachfeminismus.htm>
- [2] <http://www.b-rav.de/satzung.htm#s1>
- [3] <https://addons.mozilla.org/de/firefox/addon/6822>
- [4] [http://4.bp.blogspot.com/\\_-4sOUbNZmaw/SWTcGVBAfOI/AAAAAAAAAug/cZDRxiSjBk0/s1600-h/mannumtauschenwerbung.jpg](http://4.bp.blogspot.com/_-4sOUbNZmaw/SWTcGVBAfOI/AAAAAAAAAug/cZDRxiSjBk0/s1600-h/mannumtauschenwerbung.jpg)
- [5] <http://www.freiesmagazin.de/2009-09-05-ratespiel>
- [6] <http://www.imdb.com/title/tt0091949/> 
- [7] <http://www.imdb.com/title/tt0910970/> 
- [8] <http://www.oyonale.com/modeles.php?lang=en&page=40> 

Die Redaktion behält sich vor, Leserbriefe gegebenenfalls zu kürzen.

Die Leserbriefe kommentieren 

## Veranstaltungskalender

### Messen



Veranstaltung	Ort	Datum	Eintritt	Link
Linux Info Tag	Landau	10.10.2009	frei	<a href="http://infotag.lug-ld.de">http://infotag.lug-ld.de</a>
Ubucon	Göttingen	16.10.-18.10.09	15 EUR	<a href="http://www.ubucon.de">http://www.ubucon.de</a>
Linux-Kongress	Dresden	27.10.-30.10.09	-	<a href="http://www.linux-kongress.org/2009/">http://www.linux-kongress.org/2009/</a>
The OpenSolaris Developer Conference	Dresden	28.10.-30.10.09	-	<a href="http://www.osdevcon.org/2009">http://www.osdevcon.org/2009</a>
Linux Solution Day	Osnabrück	29.10.2009	frei	<a href="http://www.linux-solution-day.de">http://www.linux-solution-day.de</a>
androidcamp	Berlin	03.11.2009	frei	<a href="http://androidcamp-berlin.mixxt.de/">http://androidcamp-berlin.mixxt.de/</a>
World Usability Day	Weltweit	05.11.2009	-	<a href="http://www.worldusabilityday.de">http://www.worldusabilityday.de</a>
OpenRheinRuhr	Bottrop	07.11.-08.11.09	frei	<a href="http://openrheinruhr.de">http://openrheinruhr.de</a>
Brandenburger Linux-Infotag	Potsdam	21.11.2009	frei	<a href="http://www.blit.org/2009">http://www.blit.org/2009</a>

(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](mailto:redaktion@freiesMagazin.de).

## Konventionen

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

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

## Vorschau

freiesMagazin erscheint immer am ersten Sonntag eines Monats. Die November-Ausgabe wird voraussichtlich am 1. November unter anderem mit folgenden Themen veröffentlicht:

- Hardwarekompatibilität unter Linux
- Testen mit Perl

Es kann leider vorkommen, dass wir aus internen Gründen angekündigte Artikel verschieben müssen. Wir bitten dafür um Verständnis.



## Impressum

freiesMagazin erscheint als PDF und HTML einmal monatlich.  
Redaktionsschluss für die November-Ausgabe: 18. Oktober 2009

### Kontakt

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

### Autoren dieser Ausgabe

Raoul Falk **S.13**  
Florian E.J. Fruth **S.4**  
Martin Gräßlin **S.9**  
Steve Klicek **S.26**  
Rainer König **S.18**  
Marcel Jakobs **S.30**  
Mathias Menzer **S.11**  
Dominik Wagenführ **S.39**  
Philipp Walther **S.28**

### ISSN 1867-7991

Erscheinungsdatum: 4. Oktober 2009  
Erstelldatum: 6. Oktober 2009

### Redaktion

Dominik Honnef Thorsten Schmidt  
Dominik Wagenführ (Verantwortlicher Redakteur)

### Satz und Layout

Ralf Damaschke Yannic Hauptenthal  
Marcus Nelle Markus Schaub  
Sebastian Schlawtow

### Korrektur

Daniel Braun Raoul Falk  
Stefan Fangmeier Lukas Martini  
Johannes Mitlmeier Karsten Schuldt

### Veranstaltungen

Ronny Fischer

### Logo-Design

Arne Weinberg ([GNU FDL](#))

Dieses Magazin wurde mit  $\text{\LaTeX}$  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 und Beiträge in freiesMagazin unter der [GNU-Lizenz für freie Dokumentation \(FDL\)](#). Das Copyright liegt beim jeweiligen Autor. freiesMagazin unterliegt als Gesamtwerk ebenso der [GNU-Lizenz für freie Dokumentation \(FDL\)](#) mit Ausnahme von Beiträgen, die unter einer anderen Lizenz hierin veröffentlicht werden. Das Copyright liegt bei Dominik Wagenführ. Es wird die Erlaubnis gewährt, das Werk/die Werke (ohne unveränderliche Abschnitte, ohne vordere und ohne hintere Umschlagtexte) unter den Bestimmungen der GNU Free Documentation License, Version 1.2 oder jeder späteren Version, veröffentlicht von der Free Software Foundation, zu kopieren, zu verteilen und/oder zu modifizieren. Die xkcd-Comics stehen separat unter der [Creative Commons-Lizenz CC-BY-NC 2.5](#). Das Copyright liegt bei [Randall Munroe](#).