

Topthemen dieser Ausgabe

Kurze Vorstellung der zwölften Fedora-Ausgabe

Fedora 12 „Constantine“ ist am 17. November 2009 erschienen und wartet mit aktualisierten Desktops und einer Vielzahl von neuen Funktionen auf. Die Veröffentlichung wurde zweimal um eine Woche verschoben, um kurzfristig entdeckte Probleme zu beseitigen. Dieser Artikel versucht einen Eindruck von der Distribution mit Schwerpunkt auf den Neuerungen zu geben. ([weiterlesen](#))

Seite 3



Geheimniskrämerei

Jeder hat seine kleinen und größeren Geheimnisse. Seien es Kontoauszüge, persönliche Texte oder die Bilder der ersten Liebe – nicht alles ist für fremde Augen bestimmt. Der Artikel befasst sich mit den Möglichkeiten, diese Geheimnisse unter Linux vor ungewollten Blicken zu schützen. Dabei werden gute und weniger geeignete Verfahren betrachtet und zum Schluss eine praxistaugliche Lösung vorgestellt. ([weiterlesen](#))

Seite 17



GIMP in 90 Minuten (kennenlernen)

Das GNU-Bildbearbeitungsprogramm GIMP gilt bei Anwendern, die nicht regelmäßig damit arbeiten, als kompliziert, erschlagend umfangreich und vielfach unverständlich. Da mutet der Versuch, die Grundlagen von GIMP in 90 Minuten zu vermitteln, schon fast verwegen an. Dass das aber möglich ist, zeigt dieser Workshop, der auf der Ubucon 2009 gehalten wurde. ([weiterlesen](#))

Seite 23



Editorial

Auf in die nächste Runde

Pünktlich zum Weihnachtsfest gab es eine kleine Überraschung für das **freiesMagazin**-Team. Dominik Wagenführ wurde für sein Engagement bei **freiesMagazin** vom ubuntuusers.de-Team geehrt [1]. Da er aber natürlich nur der Kopf des Magazins ist und es sich ohne Körper so schlecht leben lässt, gebührt diese Ehrung dem ganzen Team, die teilweise versteckt im Hintergrund arbeiten. Und natürlich müssen wir auch den ganzen fleißigen Autoren danken, die Monat für Monat für Artikel sorgen, um **freiesMagazin** zu füllen. Danke also auch von unserer Seite.

Und was hat das Ganze gebracht? Werfen wir einen kurzen Blick zurück auf das Jahr 2009. Insgesamt wurden in den vergangenen zwölf Monaten 119 Artikel für das Magazin geschrieben (Magazinartikel der Redaktion nicht mitgezählt). Das macht also – wer keinen Taschenrechner bei der Hand hat – zehn Artikel pro Ausgabe. Ein sehr imposante Zahl für ein „kleines“ Freiwilligen-Projekt. Zu dieser Leistung beigetragen haben insgesamt 48 Autoren. Aber auch die Downloadzahlen können sich sehen lassen. Über das Jahr hinweg schwankten diese zwischen 10.000 und 12.000 Downloads.

Damit **freiesMagazin** sich auch in diesem Jahr steigern und besser werden kann, sind wir weiterhin auf der Suche nach neuen Autoren, denn ohne sie fehlt schlicht das Salz in der Maga-

zinsuppe. Gerade die verschiedenen Schreibstile und Artikelniveaus machen einen großen Reiz aus, denn so gibt es für jeden etwas im Magazin zu lesen. Daher freuen wir uns über jeden Artikelvorschlag, der von unseren Lesern bzw. Neu-Autoren kommt. Wenn Sie also eine Idee – oder keine Idee, aber Lust am Schreiben – haben, nehmen Sie einfach über redaktion@freiesMagazin.de Kontakt mit uns auf.

Auch unser kleiner, blecherner Freund des im Oktober gestarteten Programmierwettbewerbs hat seine Runden endlich zu Ende gedreht. Zwölf Teilnehmer schickten ihn durch eine lebensgefährliche Fabrikhalle. In den meisten Fällen hat der Roboter das Ziel dennoch erreicht. Wer den Blechkameraden am häufigsten ins Ziel schicken konnte, können Sie im Artikel „Abschluss des zweiten Programmierwettbewerbs“ auf [Seite 39](#) nachlesen. Dort finden Sie die Gewinner des Wettbewerbs und eine ausführliche Beschreibung, wie dieser durchgeführt wurde.

Wir wünschen Ihnen abschließend gutes Gelingen und viel Glück in allen Dingen im Jahr 2010
Ihre **freiesMagazin**-Redaktion

LINKS

- [1] <http://ikhaya.ubuntuusers.de/2009/12/24/weihnachtsgeschenke-vom-team-an-fleissigenutzer/>

Inhalt

Linux allgemein

Kurze Vorstellung der zwölften Fedora-Ausgabe	S. 3
Raus aus der Ubuntu-Paketabhängigkeits-hölle	S. 10
Der Dezember im Kernel-Rückblick	S. 15

Anleitungen

Geheimniskrämerei	S. 17
Secure Copy (SCP) per Mausclick in KDE	S. 20

Software

GIMP in 90 min (kennenlernen)	S. 23
X2goserver-home unter Ubuntu	S. 33
ASCIIPortal – The GUI is a lie!	S. 36

Magazin

Editorial	S. 2
Abschluss des zweiten Programmierwettbewerbs	S. 39
Leserbriefe	S. 49
Veranstaltungen	S. 51
Vorschau	S. 51
Konventionen	S. 51
freiesMagazin-Index 2009	S. 52
Impressum	S. 58

Das Editorial kommentieren 

Kurze Vorstellung der zwölften Fedora-Ausgabe von Hans-Joachim Baader

Fedora 12 „Constantine“ ist am 17. November 2009 erschienen und wartet mit aktualisierten Desktops und einer Vielzahl von neuen Funktionen auf. Die Veröffentlichung wurde zweimal um eine Woche verschoben, um kurzfristig entdeckte Probleme zu beseitigen. Dieser Artikel versucht, einen Eindruck von der Distribution mit Schwerpunkt auf den Neuerungen zu geben.

Redaktioneller Hinweis: Der Artikel „Fedora 12“ erschien erstmals bei Pro-Linux [1] unter der GNU Free Documentation License [2].

Überblick über die Neuerungen

Fedora 12 bringt umfassend aktualisierte Software und zahlreiche Neuerungen. Alle Neuheiten und Aktualisierungen sind in den Release Notes [3] mit Verweisen auf detailliertere Angaben festgehalten. Eine kurze Übersicht und Screenshots findet man in den One Page Release Notes [4]. In diesem Artikel sollen vor allem die Neuerungen genauer untersucht werden, sofern das möglich ist.

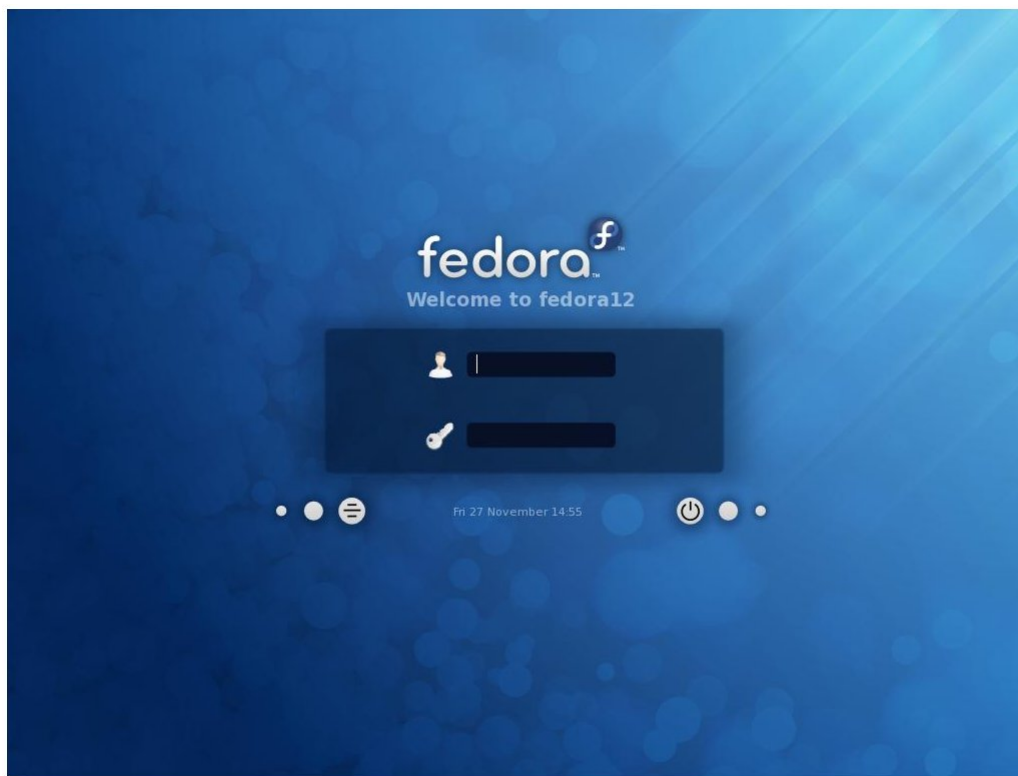
Die neue Version läuft nicht mehr auf älteren x86-Prozessoren bis

hin zum Pentium. Die Distribution wurde für Pentium II und neuere CPUs generiert und auf Intel-Atom-Prozessoren optimiert. Dies macht die Software durch die besseren Optimierungsmöglichkeiten etwas schneller.

Die Unterstützung für Webcams wurde erweitert, womit nicht nur zahlreiche neue Modelle funktionieren, sondern auch die Bildqualität,

insbesondere bei billigen Modellen, besser sein soll. GNOME 2.28.1 (mit einer optionalen Vorschau auf die GNOME-Shell) und KDE 4.3.2 stehen neben anderen als Desktop-Umgebungen zur Verfügung. Beide Oberflächen wurden weiter poliert. GNOME ersetzt das Notizzettel-Programm Tomboy durch Gnote, das weniger Ressourcenverbrauch bei gleicher Funktionalität verspricht. Empathy wurde aufgrund der besseren Integration in GNOME zum Standard-Client für Instant Messaging. In KDE sind ein aktualisiertes „Air“-Theme, konfigurierbare Tastenkürzel in Plasma, höhere Geschwindigkeit und neue Effekte des Window-Managers, ein neues Tool zum Melden von Fehlern und ein Konfigurationsprogramm für LIRC zu verzeichnen.

Der NetworkManager enthält nun eine Datenbank mit vielen Breitband-Internet-Zugangsanbietern für Mobilgeräte weltweit. Außerdem kann er jetzt mit IPv6 umgehen. Die freie Broadcom-Firmware openfwf soll dafür sorgen, dass diverse Broadcom-WLAN-Chips ohne Weiteres funktionieren. Der Video-Kodierer „Thusnelda“ für Ogg Theora wurde in der neuen Version 1.1 in-



Fedora 12 mit Display-Manager kdm. 

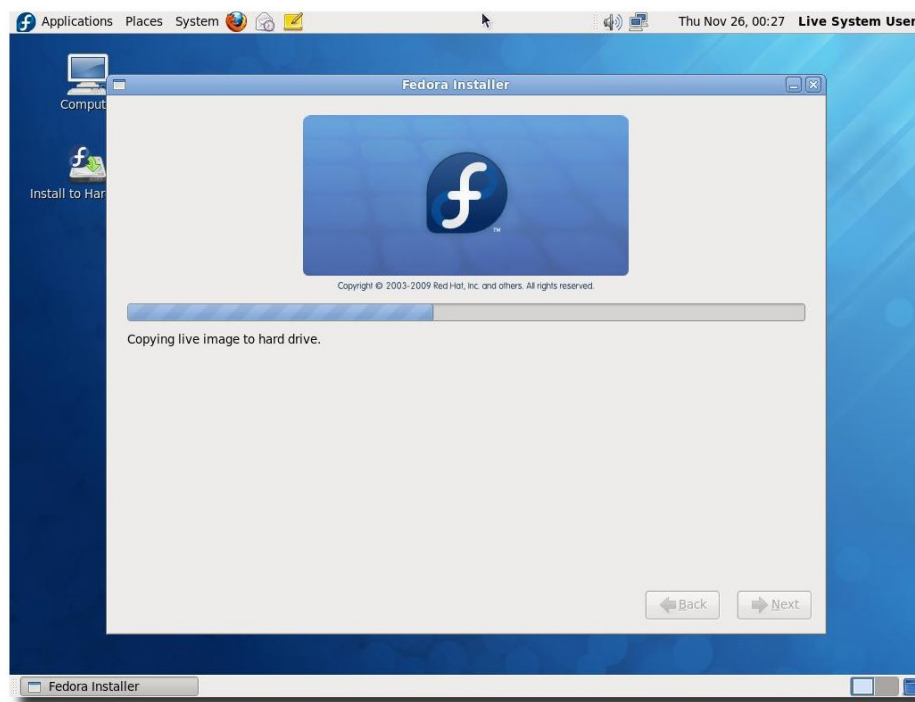
tegiert, die eine ähnliche Qualität wie das proprietäre H.264 erreichen soll. Die Lautstärkeregelung von PulseAudio unterstützt nun Profile, Wechsel des Eingabekanals und eine einfachere Einrichtung von Lautsprechern. Ferner ist eine Integration in den UPnP-Medienserver Rygel vorhanden. Auch die Energieverwaltung wurde weiter verbessert.

Für Entwickler sind unter anderem NetBeans 6.7, PHP 5.3, Perl 6, Eclipse 3.5 und das aktualisierte SystemTap von Interesse. Das „Automatic Bug Reporting Tool“ ABRT kann automatisch einen Bericht an Fedora senden, wenn ein Programm abgestürzt ist. Das erspart den Benutzern, sich spezielle Kenntnisse zur Abgabe von Fehlermeldungen aneignen zu müssen.

Das Paketformat RPM kann nun auch mit xz komprimierte Software enthalten, was es ermöglicht, mehr Software auf die Medien zu packen. Dadurch und durch das aktivierte Yum-Plug-in yum-presto wird die Downloadgröße von Paketen, insbesondere bei Aktualisierungen, kleiner. Der Bootloader GRUB unterstützt nun ext4, was Bootpartitionen mit diesem Dateisystem ermöglicht. Der Bluetooth-Dienst wird nun nur noch gestartet, wenn er benötigt wird, und beendet sich selbst, wenn 30 Sekunden kein Gerät genutzt wurde. Auch die Virtualisierung mit dem im Kernel integrierten Hypervisor KVM wurde verbessert. Sie soll schneller sein und weniger Speicher benötigen sowie Hotplug von Netzwerkkarten, Booten von PXE und flexiblere Netzwerkkonfigurationen beherrschen.

Die Grafikerunterstützung wurde weiter verbessert. Experimentelle 3D-Unterstützung für ATI-Grafikkarten mit Radeon HD 2400 und neuer durch den freien Treiber kann aktiviert werden. Die Modus-Umschaltung im Kernel soll nun auch mit NVIDIA-Karten funktionieren.

Wenn mehrere Monitore vorhanden sind, funktioniert das Booten nun besser, und der Desktop erstreckt sich standardmäßig über alle Monitore. Multi-Pointer-X ist enthalten. Die grafische Oberfläche von Moblin ist als Paket verfügbar, außerdem soll es einen „Moblin Fedora Remix“ als installierbare Live-CD geben.



Installation von der Live-CD. 🔍

Die Initial Ramdisk wird nun mit Dracut generiert, was flexibler und distributionsunabhängig sein soll. Ein PackageKit-Plug-in kann automatisch nach einem fehlenden Paket suchen, wenn der Benutzer versucht, ein nicht vorhandenes Programm zu starten. Die Sicherheit wurde weiter verbessert, indem die Privilegien von Prozessen, insbesondere systemnahen Prozessen, weiter eingeschränkt wurden. Prozesse können außerdem in einer „SELinux-Sandbox“ laufen.

Trotz der vielen Änderungen und des verkürzten Entwicklungszyklus wurde nach Angaben der Entwickler dieses Mal besonders viel Wert auf die Stabilisierung und auf Detailverbesserungen gelegt.

Das System beruht auf dem Linux-Kernel 2.6.31.5. Einige der aktualisierten Bestandteile von Fedora 12 sind GNOME 2.28.1, KDE 4.3.2, Xfce 4.6.1, LXDE 0.4.2, OpenOffice.org 3.1.1, Firefox 3.5.5, Samba 3.4.2 und Python 2.6.2.

Installation

Fedora kann auf verschiedene Arten installiert werden. Von einem Live-System aus, das als CD verfügbar ist, ist eine Möglichkeit. Ein direktes Star-

ten eines Installationssystems ist mit einer DVD oder dem äquivalenten Satz von CDs möglich. Auch eine Installation von einem minimalen Image (**boot.iso**) ist möglich, das dann den Großteil der benötigten Pakete von einem Server herunterlädt. Ein etwas größeres, aber dennoch stark reduziertes ISO-Image ist **netinst.iso**. Zudem kann man mit einem einzelnen Kommando (**livecd-iso-to-disk** unter Linux, **liveusb-creator** unter Windows) einen bootfähigen USB-Stick mit Fedora erzeugen. Dies ist in der Installationsanleitung beschrieben.

Von Fedora 12 stehen DVDs für die Architekturen x86, x86_64 und PPC sowie Live-CDs mit GNOME und KDE für x86 und x86_64 zur Verfügung. Bei den Live-CDs ist der Umfang der Software natürlich beschränkt; das ist der Grund dafür, dass es verschiedene Varianten gibt, darunter eine mit KDE, eine mit GNOME und eine mit LXDE als Desktop. Letztere wurde aufgrund von Fehlern zunächst zurückgezogen, ist aber mittlerweile wieder verfügbar. Mehr Software kann man von der DVD oder alternativ von einem Satz aus sechs CDs installieren. Die DVD bietet auch ein Rettungssystem sowie eine Textmodus-Installation.

Die Installation ist so einfach, dass sie von allen Anwendern meist ohne Probleme bewältigt werden kann, und das ist ja nach wie vor wichtig, solange nur wenige Rechner mit vorinstalliertem Linux ausgeliefert werden.

Die grafische Installation benötigt 192 MB RAM, die textbasierte kommt mit 64 MB aus. Fedora empfiehlt, die grafische Installation

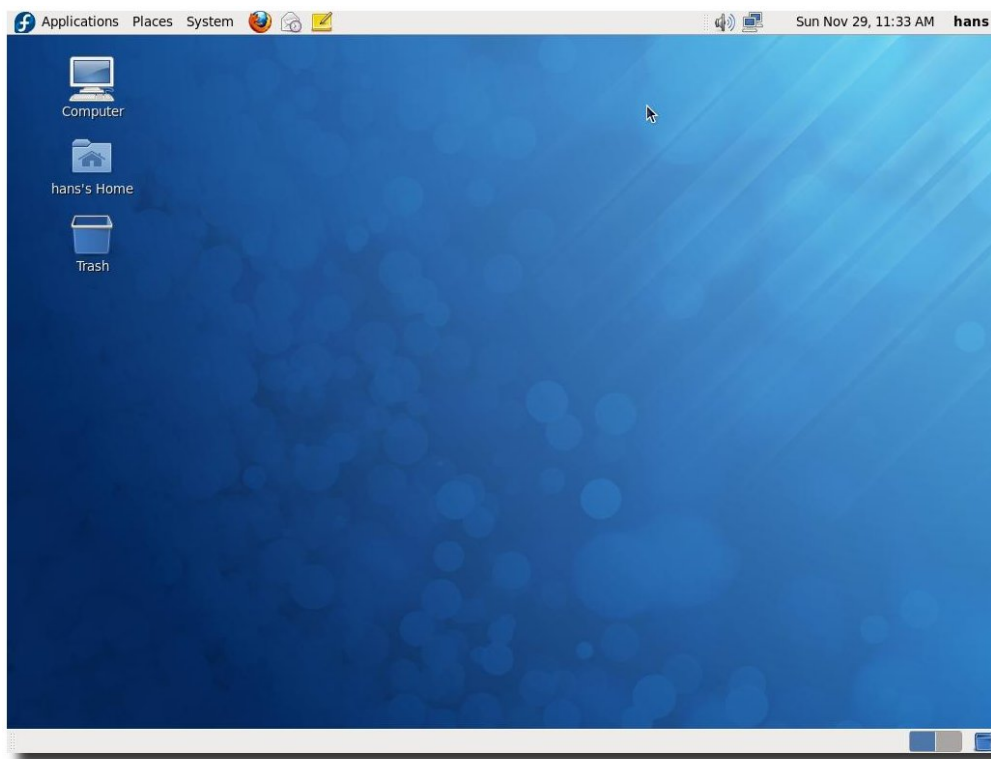
zu verwenden. Ist das nicht möglich, sollte man die grafische Installation per VNC von einem anderen Rechner aus steuern oder die automatische Installation mit Kickstart wählen. Wenn nur der Textmodus bleibt, kann man nur ein System mit einer festen Auswahl von Basispaketen und einer Standard-Partitionierung aufsetzen.

Die grafische Installation ist gegenüber Fedora 11 nahezu unverändert (siehe „Kurze Vorstellung der elften Fedora-Ausgabe“, [freiesMagazin 08/2009 \[5\]](#)). Die wesentliche Neuerung ist, dass der Bootloader GRUB jetzt ext4 unterstützt, was Bootpartitionen mit diesem Dateisystem ermöglicht. Man benötigt keine separate Bootpartition mit einem anderen Dateisystem wie ext2 oder ext3 mehr.

Laufender Betrieb

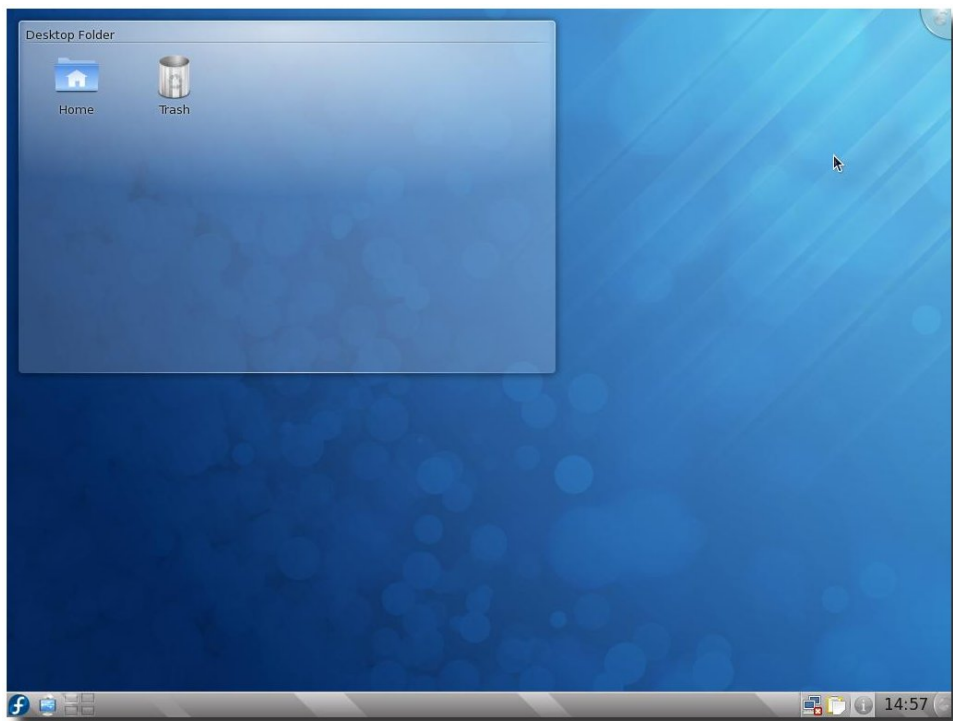
Fedora 12 soll nach Angaben der Entwickler nochmals etwas schneller starten, nachdem Fedora 11 diesbezüglich einige Erfolge feierte. Tatsächlich wurde im Test eine weitere geringfügige Beschleunigung festgestellt. Die Startzeit bis zum Login-Bildschirm verringerte sich von 35 auf etwa 25 Sekunden.

GNOME benötigt in Fedora 12 direkt nach dem Start mit einem geöffneten Terminal-Fenster 205 MB, KDE 260 MB und LXDE 145 MB. Bei allen erhöht sich der Speicherbedarf nach einigen Stunden Laufzeit nochmals um ca. 50 MB. Immer wieder faszinierend ist dabei, dass selbst vermeintlich simple Programme unter top einen virtuellen



GNOME-Desktop beim Start. 

Speicherbedarf von über 500 MB für sich reklamieren. Zum Glück belegt keines der Programme des Standard-Desktops mehr als 16 MB im realen RAM und davon ist auch noch ein guter Teil Platz, der von den dynamischen Bibliotheken benötigt und von allen Programmen gemeinsam genutzt wird.



KDE-Desktop beim Start. 🔍

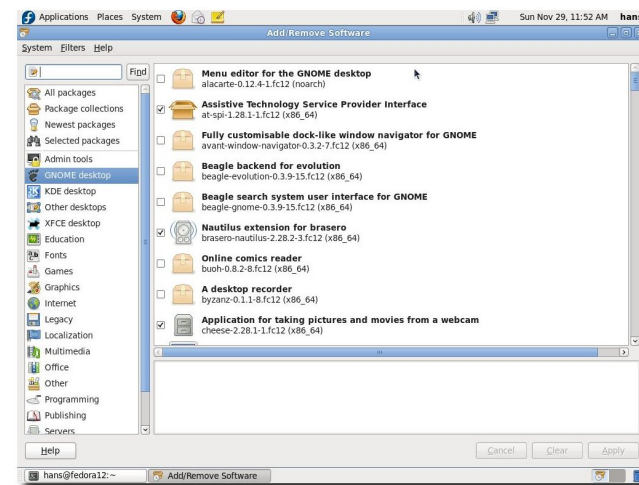
Alle drei Varianten hatten das Problem, dass der NetworkManager nach dem Start keine Netzwerkverbindung herstellen konnte. Erst nach einem Aus- und Wiedereinschalten des Netzwerks funktionierte es. Der Grund kann eigentlich nur in

irgendwelchen Einstellungen liegen, da es mit anderen Distributionen diese Probleme nicht gab.

Als Office-Suite ist in der Live-Installation nicht OpenOffice.org, sondern unter GNOME Abiword und sonst kein weiteres Büroprogramm enthalten. KDE bringt einen Teil der KOffice-Suite (die Textverarbeitung KWord, die Tabellenkalkulation KCalc und das Präsentationsprogramm KPresenter) mit, allerdings in der alten Version 1.6.3. LXDE bietet ebenfalls nur Abiword an und hat generell eine ähnliche Software-Auswahl wie GNOME. Als Browser ist in GNOME und LXDE Firefox 3.5.5 installiert, in KDE Konqueror.

Das Grafikprogramm GIMP ist in keiner Fedora-Standardinstallation mehr vorhanden. Unter KDE ist KolourPaint als einfacher Ersatz vorhanden, unter den anderen Varianten muss man GIMP oder ein anderes Programm bei Bedarf von Hand nachinstallieren. Die Paketverwaltung erfolgt über PackageKit, daneben steht yum auf der Kommandozeile zur Ver-

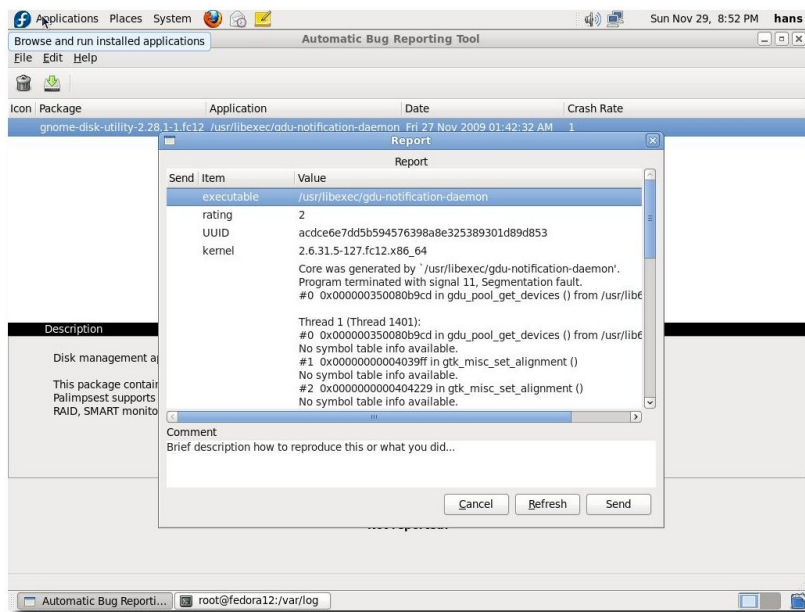
fügung. Das GNOME-Frontend von PackageKit heißt gpk-application. Dessen Bedienung ist intuitiv und übersichtlich. Das einzige unter Fedora 11 festgestellte Problem, dass im 64-Bit-System sowohl i586- als auch amd64-Pakete angezeigt wurden, wurde in Fedora 12 behoben. Ganz ohne Kritik kann das Programm jedoch leider nicht bleiben: Es gibt keine Möglichkeit, die Software-Repositoryys zu ändern oder neue hinzuzufügen. Man muss entweder die yum-Konfigurationsdateien direkt editieren oder sie über den Webbrowser importieren. Unter KDE wird KPackageKit verwendet, das im Test so weit einwandfrei funktionierte. Nur bei der Installation von neuen Repositoryys zeigte es Mängel, aber dazu im Punkt „Multimedia“ mehr.



Paketverwaltung mit gpk-application. 🔍

Die GNOME-Benutzer werden begrüßen, dass GNOME in der neuen Version wiederum nur wenige größere Änderungen aufweist. Die größ-

te optische Änderung ist natürlich, dass Icons in Menüs und Buttons nicht mehr angezeigt werden. Das alte Verhalten ist aber wieder herstellbar. Standardmäßig werden nur noch zwei Workspaces angezeigt. Das Notizblock-Programm Tomboy, das Mono benötigte, wurde durch Gnote ersetzt. In bestehenden Installationen wird Tomboy, das weiterhin in den Repositories enthalten ist, aber nicht ersetzt. Empathy wurde aufgrund der besseren Integration in GNOME zum Standard-Client für Instant Messaging anstelle von Pidgin. Empathy besitzt aber zwei wesentliche Nachteile gegenüber Pidgin: Es unterstützt keine Proxy-Server und ist nicht über Plugins erweiterbar. Dies dürfte viele Benutzer dazu veranlassen, zu Pidgin zurückzukehren.



Ein aufgezeichnetes Problem in ABRT. 🔍

Neu ist das „Automatic Bug Reporting Tool“ ABRT, eine Entwicklung von Red Hat, die nicht so simpel ist, wie es sich anhört. Das Programm bekommt Abstürze von Programmen mit und hält diese fest. Öffnet man ABRT, so zeigt es eine Liste der verzeichneten Abstürze an. Der Benutzer hat nun die Möglichkeit, Einträge zu löschen oder sich genauer anzusehen. Den Bericht, den ABRT erzeugt, kann man optional an den zuständigen Bugtracker senden, wobei man noch auswählen kann, welche Informationen man mitsendet. Es wird nichts automatisch versandt.

ABRT erspart den Benutzern, sich spezielle Kenntnisse zur Abgabe von Fehlermeldungen aneignen zu müssen. Es ist in Python geschrieben und verfügt über Plug-ins. Diese enthalten das Wissen, an welche Stelle welcher Bericht gesendet werden muss. Man kann davon ausgehen, dass im Laufe der Zeit noch einige Plug-ins hinzukommen werden.

KDE 4.3.2 bringt viele Verbesserungen und Erweiterungen gegenüber Version 4.2. Wie schon in der Betrachtung von Ubuntu 9.10 dargestellt (siehe „Ubuntu 9.10 – Die neue Version „Karmic Koala“ vorgestellt“, [freiesMagazin 12/2009 \[6\]](#)), bietet die neue Version kaum noch Anlass zur Kritik; der erheblich höhere Speicherbedarf gegenüber KDE3 muss jedoch weiterhin ge-

nannt werden. Außerdem kam es zu einem Absturz des Plasma-Desktops, der allerdings nicht näher analysiert wurde. Die Absturzbehandlung des Systems fängt diesen ab und ermöglicht es dem Benutzer, das Programm neu zu starten, so dass die Auswirkungen gering sind.

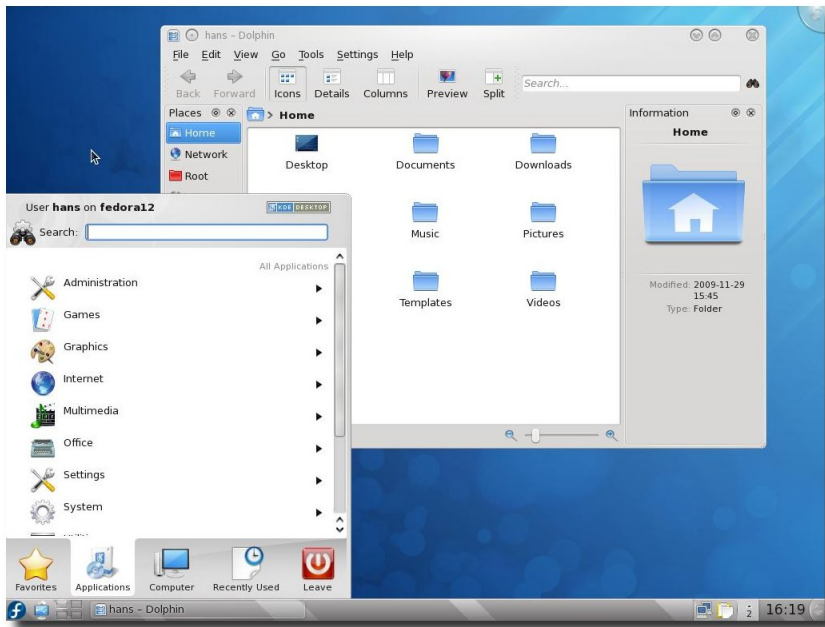
KDE nutzt weiterhin das „nm-applet“ von „NetworkManager-gnome“, wodurch auch die Passwortverwaltung für WLAN-Zugänge usw. über ein GNOME-Programm läuft. Unter GNOME läuft wie schon in früheren Versionen gnome-screensaver unnötigerweise auch dann, wenn er gar nicht benötigt wird, weil der Screensaver abgeschaltet wurde.

Neben den bekannten Desktops bietet Fedora 12 auch die Möglichkeit, den Moblin-Desktop zu installieren, der für kleine Bildschirmauflösungen Vorteile bieten könnte. Die Installation erfolgt mit **yum install @moblin-desktop**. Leider scheint dies Compositing oder etwas anderes zu benötigen, was in der Testumgebung nicht zur Verfügung stand.

Multimedia

Aus den allgemein bekannten Gründen kann Fedora, ebenso wie die meisten anderen Distributionen, nur wenige Medienformate abspielen, da es viele benötigte Codecs nicht mitliefern kann. Wenn man versucht, ein Video abzuspielen, erhält man mit dem KDE-Videoplayer „DragonPlayer“ nicht einmal eine Meldung, dass etwas nicht geht. Nur Totem bietet auch die Option an, über die Paketverwaltung nach passen-

den Plug-ins zu suchen. Dazu muss man aber vorher in der Paketverwaltung die zusätzlichen Repositorys eintragen. Wenn man weiß, wie es geht, ist es im Prinzip ganz einfach. Über die Webseite von RPM Fusion [7] kann man Pakete installieren, die die Repositorys hinzufügen. Allerdings funktionierte die direkte Installation über den Webbrowser nicht. Unter GNOME mit Firefox tat sich nach „Öffnen mit ...“ gar nichts, obwohl das Paket heruntergeladen wurde. Unter KDE und Konqueror wurde zwar KPackageKit geöffnet, produzierte jedoch nur einen „Unknown Error“.



KDE mit Dateimanager Dolphin.

Das war nicht das einzige Problem von KPackageKit. Beim Hinzufügen der RPM-Fusion-

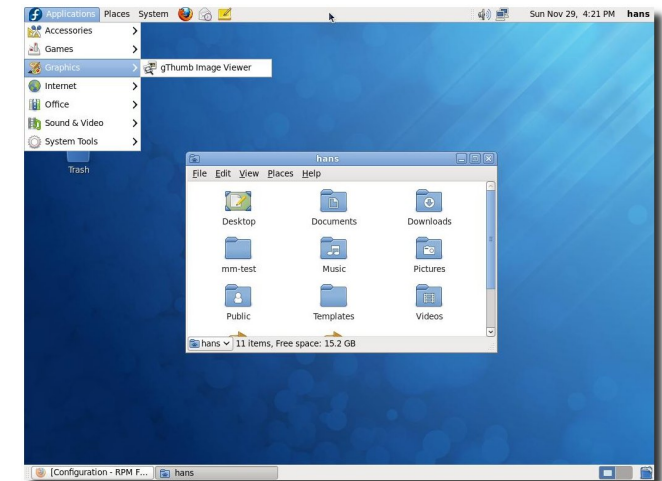
Repositorys geriet KPackageKit in eine Endlosschleife, in der es immer wieder die Schlüssel des Repositorys hinzufügen wollte. Im Endeffekt ist es – egal ob GNOME oder KDE – nötig, nach dem Herunterladen ein Terminal zu öffnen und die Pakete mit `rpm -i` zu installieren.

Nun sind zwar die Repositorys von RPM Fusion eingerichtet, aber die benötigten Plug-ins muss man von Hand suchen und installieren. Unter GNOME dürften **gstreamer-plugins-good** und **gnash** ein guter Einstieg sein, unter KDE **xine-lib-extras-freeworld** und **gnash**, denn KDE verwendet in der Standard-einstellung das Xine-Backend.

Unter GNOME sollte man beim Hinzufügen von GStreamer-Plug-ins nicht zu übereifrig sein (das Xine-Backend von Totem wurde entfernt, GStreamer ist nun die einzige Option). Im Test kam es dann nämlich dazu, dass Totem und sogar Nautilus gar nicht mehr starten wollten – sie hingen einfach. Die Deinstallation von **gstreamer-plugins-bad** und **gstreamer-plugins-ugly** rettete Totem. Doch Nautilus wollte immer noch nicht starten. Glücklicherweise ließ sich noch **strace** nachinstallieren. Dieses offenbarte, dass das Programm auf eine

Antwort vom policykit-Daemon über D-Bus wartete. Dieser war offensichtlich einfach abgestürzt.

Der Grund blieb letztlich unklar – ein Neustart des Desktops oder des ganzen Rechners löste dieses Problem. Die Installation der Plug-ins hatte vielleicht gar nichts damit zu tun.



GNOME mit Dateimanager Nautilus.

Nach der Installation der Plug-ins funktionierten die Medienplayer erwartungsgemäß. Enttäuschend war aber die mangelnde Hilfestellung für die Benutzer bei der Installation der zusätzlichen Plug-ins. Es war einiger Aufwand nötig, alles zum Laufen zu bekommen, doch glücklicherweise muss man ihn nur ein einziges Mal betreiben.

Eine erfreuliche Neuerung ist dagegen der Videokodierer „Thusnelda“ für Ogg Theora, der in der neuen Version 1.1 integriert wurde. Die neue Version soll eine ähnliche Qualität wie das proprietäre H.264 erreichen, ist aber vollständig freie Software. Das Angebot an Medien mit diesem Format dürfte zwar noch schmal sein, aber mit

„Thusnelda“ können die Benutzer nun Videos in besserer Qualität erzeugen, was einen wichtigen Schritt zur Demokratisierung des Webs darstellt.

Bemerkenswert ist noch, dass im Test die gleichen Probleme auftraten wie unter Ubuntu 9.10: Unter GNOME gab es ein Video, das von Totem nicht abgespielt werden konnte, obwohl das Video durchaus dekodiert werden konnte. Unter KDE dagegen wurden JPEG-Dateien von der HTML-Komponente nicht richtig dargestellt.

Weitere Neuerungen

Die Rechte von diversen Daemonen wurden eingeschränkt, um sie für Angriffe uninteressant zu machen. Eine SELinux-Sandbox ermöglicht es nun, GUI-Programme mit stark eingeschränkten Rechten zu starten. Es ist jedoch, wie die Fedora-Entwickler eingestehen, kaum möglich, diese Funktion mit den Anwendungen zu nutzen, die es am meisten nötig hätten. Da die Sandbox Netzverbindungen unterbindet und Dateizugriffe auf eine nur temporär existierende Verzeichnis-Hierarchie beschränkt, ist sie fast nur für Anwendungen nutzbar, die aus dem Browser heraus gestartet werden, beispielsweise Programme, die Word- oder PDF-Dateien anzeigen. Damit das funktioniert, muss man das Paket **policycoreutils-sandbox** nachinstallieren.

Fazit

Fedora hat unter anderem das Ziel, die neuesten Entwicklungen aus der Open-Source-Welt schnell zu den Anwendern zu bringen. Der Preis

für die große Aktualität ist jedoch, dass eine Vielzahl von Funktionen unfertig ausgeliefert wird. Bereits kurz nach der Veröffentlichung von Fedora 12 standen über 100 Updates bereit und das nur für eine Standardinstallation. Dies könnte auf der einen Seite als mangelnde Qualität ausgelegt werden. Andererseits passt es eher zu den Prinzipien von Freier Software, häufig neue Versionen bereitzustellen und von den Benutzern testen zu lassen. Dass verschiedene Zielgruppen andere Anforderungen haben, ist klar. Auf der anderen Seite ist aber auch zu sehen, dass die Fehler korrigiert werden - und das auch noch schnell.

Trotz einiger bereits erfolgter Updates zeigte Fedora 12 eine Reihe von Problemen. Glücklicherweise kam man mit Geduld und einigen Workarounds meist doch zum Ziel. Dennoch kann man Fedora Heimanwendern und Anfängern kaum empfehlen. In anderen Distributionen laufen die meisten Dinge etwas runder und es wird mehr Hilfestellung geboten, wenn etwas schief geht. Die Zielgruppe von Fedora dürfte auch eher bei bereits in Linux erfahrenen Anwendern und Entwicklern liegen. Für diese dürften die Aktualität und die zahlreichen Innovationen ein gewichtiges Argument sein. Auch als Desktopsystem in Firmen und Organisationen ist Fedora 12 durchaus geeignet, allerdings nur, wenn die Wartung von erfahrenen Administratoren übernommen wird.

Um das Ganze etwas positiver abzuschließen, sei jedoch daran erinnert, dass in Fedora auch

eine riesige Menge Technologie steckt, die einfach funktioniert, aber hier nicht weiter erwähnt werden kann. In den Bereichen Virtualisierung, Tracing, SELinux und vielem mehr nimmt Fedora eine Vorreiterrolle ein, nicht zuletzt durch die zahlreichen Red-Hat-Mitarbeiter, die hier mitwirken. Für die Zielgruppe ist das ein Grund mehr, Fedora einzusetzen.

LINKS

- [1] <http://www.pro-linux.de/berichte/fedora12.html>
- [2] <http://www.gnu.org/copyleft/fdl.html> 
- [3] <http://docs.fedoraproject.org/> 
- [4] http://fedoraproject.org/wiki/Fedora_12_tour 
- [5] <http://www.freiesmagazin.de/freiesMagazin-2009-08>
- [6] <http://www.freiesmagazin.de/freiesMagazin-2009-12>
- [7] <http://rpmfusion.org/> 
- [8] <http://www.fedoraforum.de/>

Autoreninformation

Hans-Joachim Baader befasst sich seit 1993 mit Linux. 1994 schloss er sein Informatikstudium erfolgreich ab und machte die Softwareentwicklung zum Beruf. Seit 1995 arbeitet er freiberuflich. Daneben ist er einer der Betreiber von Pro-Linux.de.

Diesen Artikel kommentieren 



Raus aus der Ubuntu-Paketabhängigkeitshölle von Dominik Wagenführ

Wer ein „normales“ Ubuntu-System, d. h. eines mit Internetanschluss besitzt, wird das Problem nicht kennen. Alle anderen Nutzer sind aber sicherlich schon des Öfteren verzweifelt, als sie ein Paket installieren wollten, aber keinen direkten Internetzugang hatten. Das Paket herunterladen ist schön und gut, aber es fehlen immer die ganzen Abhängigkeiten.

Die native Lösung, bei der man ein Paket nach dem anderen per USB-Stick von einem PC mit Internetanschluss zu dem ohne trägt, ist alles andere als komfortabel. Die zweite direkte Lösung wäre die Installation der Pakete auf dem ersten Ubuntu-System, um diese dann per USB-Stick zu System B zu tragen. Der Haken an der Sache: Welche Pakete waren schon auf System 1 installiert, welche sind wirklich notwendig? Wenn man zwei ähnliche Systeme betreibt, ist das meist noch ein lösbares Problem, aber wenn diese sich total unterscheiden, z. B. ein Ubuntu-System als Basis und ein Kubuntu-System als Ziel, dann ist unklar welche GTK- oder GNOME-Abhängigkeiten nachinstalliert werden müssen.

Aus diesem Grund wäre es gut, wenn man alle Pakete für eine Offline-Installation herunterladen kann, die man als Abhängigkeiten für ein zu installierendes Paket benötigt. Das klingt nach Rekursion und genau darin liegt die Lösung.

Hinweis: Die hier vorgestellte Lösung sollte auf allen Systemen funktionieren, die APT für die

Paketverwaltung einsetzen (z. B. Debian, alle Ubuntu-Derivate etc.).

Zielsetzung

Das Ziel ist es also, ein kleines Bash-Skript zu schreiben, welches einem zu einer vordefinierten Paketliste A alle abhängigen Pakete rekursiv heraus sucht. Rekursiv bedeutet, dass zuerst die normalen Abhängigkeiten der ersten Pakete A gesucht werden, die eine Paketliste B ergeben. Zu der Vereinigung von A und B sucht man dann wieder alle Abhängigkeiten C und so weiter. Das macht man solange, bis man keine neuen Abhängigkeiten dazu gewinnt.

Für die Umsetzung wird der Befehl **apt-cache depends -i** genutzt, der einem zu einer Liste von Paketen die wichtigen Abhängigkeiten auflistet, d. h. ohne empfohlene und vorgeschlagene Pakete.

Abhängigkeiten suchen

Die erste Version des Skriptes soll nur die Abhängigkeiten der, in einer Datei **packages** durch Zeilenumbrüche getrennte, übergebenen Pakete filtern. Der Befehl dazu lautet:

```
$ xargs -a packages apt-cache \
depends -i | nawk '{ print $NF }' | \
sort | uniq
```

xargs liest dabei die Argumente aus der Datei **packages** aus und übergibt diese an **apt-cache**. Da bei diesem Aufruf aber Zeilen wie

```
texlive-common
Haengt ab: tex-common
```

entstehen, filtert man mit **nawk** noch das letzte Wort jeder Zeile über **\$NF** heraus (siehe „Keine Angst vor awk“, freiesMagazin 07/2009 [1]). Zum Schluss wird die Liste noch sortiert (**sort**) und doppelte Einträge entfernt (**uniq**).

Im Skript steht aber noch mehr: am Anfang gibt es Abfragen, ob die Anzahl der Argumente stimmt oder die übergebene Paketdatei überhaupt existiert. Darüber hinaus werden noch ein paar Variablen deklariert, die später etwas sinnvoller genutzt werden.

```
#!/bin/bash

#####
# Argumentpruefung
#####

if [ $# -eq 0 ]
then
    echo "Es wurden keine Parameter \
uebergeben."
    exit 1
fi

if [ $# -gt 1 ]
then
    echo "Es wurden mehr als ein \
Parameter uebergeben."
    exit 1
```



```

fi

if !([ -f "$1" ])
then
    echo "Datei $1 existiert nicht~
    ."
    exit 1
fi

#####
# "Globale" Variablen
# (Zum Veraendern!)
#####

# Zusatz beim Speichern der ~
# Paketliste
SUFFIX="all"

# Verzeichnis fuer temporaere ~
# Dateien
TEMP="/tmp"

#####
# "Lokale" Variablen
#####

# Quelldatei mit den Paketen, die ~
# installiert
# werden sollen.
ORIGIN="$1"

#####
# Implementierung
#####

# Temporaere Datei

```

```

TEMPFILE="$TEMP/$ORIGIN-tmp"

# Pakete extrahieren
xargs -a "$ORIGIN" apt-cache ~
depends -i | \
    nawk '{ print $NF }' | sort | ~
    uniq \
    > "$TEMPFILE" | exit 1

# Kopiere komplette Paketliste
cp "$TEMPFILE" "$ORIGIN-$SUFFIX" | ~
exit 1

echo
echo "Paketliste mit kopiert nach ~
\"$ORIGIN-$SUFFIX\"."

```

Listing 1: *apt-depends-pre.sh*

Nachdem man die Datei per

```
$ chmod +x apt-depends-pre.sh
```

ausführbar gemacht hat (dies wird in Zukunft nicht mehr extra erwähnt), kann man die Abhängigkeitsliste per

```
$ ./apt-depends-pre.sh packages
```

erstellen. Das Ergebnis ist in der Datei **packages-all** zu finden.

Rekursion einbauen

Das Skript muss nun noch so erweitert werden, dass es rekursiv arbeitet.

Die Idee wurde oben bereits beschrieben. Realisiert wird das Ganze, indem man die Anzahl

der Gesamtpakete nach jeder Abhängigkeitenbestimmung ausgewertet und mit dem vorherigen Schritt vergleicht. Ist sie identisch, sind keine neuen Pakete dazu gekommen und das Skript kann beendet werden.

Zusätzlich muss man auf einige Ausgaben von **apt-cache** aufpassen. Es gibt beispielsweise Ausgaben wie

```

x11-common
|Haengt ab: debconf
  Haengt ab: <debconf-2.0>
    cdebconf
    debconf

```

Hier arbeitet **apt-cache** leider etwas missverständlich. Obige Ausgabe bedeutet, dass das Paket **x11-common** von **debconf** oder (!) von **<debconf-2.0>** abhängt. Das heißt, das Oder steht immer vor dem Paket, welches „verodert“ wird. **<debconf-2.0>** selbst ist dabei kein echtes, sondern ein virtuelles Paket. Das bedeutet, dieses Paket existiert nicht wirklich, sondern stellt nur zwei weitere Pakete **cdebconf** und **debconf** zur Verfügung.

Da die Veroderung (vorerst) zu kompliziert zu filtern ist, macht man es sich einfach und entfernt einfach nur die virtuellen Pakete mittels **egrep -v "<.*>"** aus der Liste (die Option **-v** negiert den Suchparameter und findet somit alles, außer dem Suchbegriff).

```
#!/bin/bash
```



```
#####
# Argumentpruefung
#####

if [ $# -eq 0 ]
then
    echo "Es wurden keine Parameter
    uebergeben."
    exit 1
fi

if [ $# -gt 1 ]
then
    echo "Es wurden mehr als ein
    Parameter uebergeben."
    exit 1
fi

if !([ -f "$1" ])
then
    echo "Datei $1 existiert nicht
    ."
    exit 1
fi

#####
# "Globale" Variablen
# (Zum Veraendern!)
#####

# Zusatz beim Speichern der
Paketliste
SUFFIX="all"

# Verzeichnis fuer temporaere
Dateien
```

```
TEMP="/tmp"

#####
# "Lokale" Variablen
#####

# Quelldatei mit den Paketen, die
installiert
# werden sollen.
ORIGIN="$1"

# Name der letzten Paketdatei
LAST="$ORIGIN"

# Name der neuen Paketdatei
NEW="$ORIGIN"

# Anzahl der Pakete in der
Quelldatei
NUMBER='wc -l "$ORIGIN" | awk '{
print $1 }''

# Alte Anzahl Pakete zum Vergleich
OLDNUMBER="0"

# Zaehler fuer die Iteration
COUNTER=0

#####
# Implementierung
#####

# Solange iterieren, bis keine
neuen Pakete
# dazu gekommen sind.
while [ $OLDNUMBER != $NUMBER ]
```

```
do
    # Neue Iteration beginnen.
    # Neuen Dateinamen setzen.
    COUNTER=$(( $COUNTER + 1 ))
    NEW="$TEMP/$ORIGIN$COUNTER"
    TEMPFILE="$NEW-tmp"

    # Pakete extrahieren
    xargs -a "$LAST" apt-cache
depends -i | \
    awk '{ print $NF }' | sort
    | uniq | \
    egrep -v "<.*>" \
    > "$TEMPFILE" | exit 1

    # Alte Pakete hinzufuegen
    cat "$LAST" >> "$TEMPFILE" |
    exit 1

    # Paketliste sortieren
    cat "$TEMPFILE" | sort | uniq >
"$NEW" | exit 1

    # Alte Daten sichern
    LAST="$NEW"
    OLDNUMBER=$NUMBER

    # Neue Anzahl Pakete berechnen
    NUMBER='wc -l "$NEW" | awk '{
print $1 }''
done

# Kopiere komplette Paketliste
cp "$NEW" "$ORIGIN-$SUFFIX" | exit
1
echo
```




```
echo "Paketliste mit $NUMBER
Paketen kopiert nach \"$ORIGIN-
$SUFFIX\"."
echo
echo "Zum Download der Pakete:"
echo "xargs -a \"$ORIGIN-$SUFFIX\"
sudo apt-get install --reinstall -d
"
echo
exit 0
```

Listing 2: *apt-depends.sh*

Weitere Erläuterungen zum Skript

Die einzelnen Zeilen des Skripts sollen noch kurz erläutert werden.

Für den Vergleich, ob man keine weiteren neuen Pakete gefunden hat, wird in jeder Iteration die Anzahl der Pakete (= Anzahl Zeilen) mit dem Befehl `wc -l` gesucht. Da dieser aber immer auch noch den Dateinamen ausgibt, benötigt man nur die erste Ausgabe, welche wieder mit `nawk` gefiltert wird:

```
NUMBER='wc -l "$ORIGIN" | nawk '{
print $1 }'
```

Die Iteration findet dann über eine `while`-Schleife statt und vergleicht die alte Anzahl der Pakete mit der neuen:

```
while [ $OLDNUMBER != $NUMBER ]
do
    ...
done
```

Damit die temporären Dateien einen eindeutigen Namen bei jeder Iteration bekommen, wird ein Zähler (**COUNTER**) eingeführt, der bei jedem Schritt hochgezählt und in den neuen Dateinamen integriert wird:

```
COUNTER=$(( COUNTER + 1 ))
NEW="$TEMP/$ORIGIN$COUNTER"
```

Zur Sicherheit werden bei jeder Iteration neben den Abhängigkeiten auch die alten Pakete, die als Basis für die Abhängigkeitssuche benutzt wurden, in das Endergebnis geschrieben:

```
cat "$LAST" >> "$TEMPFILE" |
exit 1
```

Das `| exit 1` bedeutet im Übrigen, dass im Fehlerfall das Skript mit einem Fehlerstatus 1 verlassen wird. Im Erfolgsfall ist der Rückgabewert 0 (**exit 0** in der letzten Zeile).

Herunterladen der gefilterten Pakete

Ruft man das Skript auf, erhält man eine Ausgabe ähnlich der folgenden:

```
$ ./apt-depends.sh packages

Paketliste mit 132 Paketen kopiert
nach packages-all.

Zum Download der Pakete:
xargs -a "packages-all" apt-get
install --reinstall -d
```

Die letzte Zeile gibt bereits an, wie man mittels `apt-get` die gefilterten Pakete herunterla-

den kann (durch die Option `-d` wird nichts installiert). Die Option `--reinstall` sorgt dafür, dass ein Paket wirklich neu heruntergeladen wird, auch wenn es bereits im System installiert war.

Nach der Ausführung der Zeile (mit Root-Rechten) findet man alle heruntergeladenen Pakete im Verzeichnis `/var/cache/apt/archives`:

```
# xargs -a "packages-all" apt-get
install --reinstall -d
...
Es wurden 116MB in 2min38s geholt
(730kB/s)
Herunterladen abgeschlossen und im
Nur-Herunterladen-Modus
```

Hinweis: Das Verzeichnis sollte man vor der Ausführung mittels `apt-get clean` komplett leeren, damit man nur die Pakete am Ende dort liegen hat, die man wirklich benötigt.

Lokale Paketquelle erstellen

Die Pakete kann man nach dem Herunterladen auf einen USB-Stick kopieren, zum Zielrechner tragen und dort mittels

```
# dpkg -i *.deb
```

installieren.

Etwas praktikabler ist es aber vielleicht, wenn man eine lokale Paketquelle eröffnet.

Dazu führt man im Verzeichnis `/var/cache/apt/archives/` folgenden Befehl aus:



```
# dpkg-scanpackages ./ /dev/null | \
gzip > Packages.gz
```

Dies erstellt eine Paketliste der vorhandenen Pakete und packt diese in eine neue Datei **Packages.gz**.

Diese Datei kopiert man dann zusammen mit den Paketen auf den USB-Stick – z. B. in ein Unterverzeichnis **karmic**, wenn man diese Ubuntu-Version nutzt.

Auf dem Zielrechner fügt man danach in der Paketverwaltung (**Synaptic** etc.) oder direkt in der Datei **/etc/apt/sources.list** die folgende neue Paketquelle ein:

```
deb file:/media/USB-STICK/karmic ./
```

Dabei ist **USB-STICK** natürlich der Einhängpunkt des USB-Sticks.

Nach dem Neueinlesen der Paketquellen, z. B. über

```
# apt-get update
```

stehen die neuen Pakete vom USB-Stick in der Paketverwaltung zur Verfügung und man kann ganz normal die Pakete installieren, die man ganz am Anfang in die Datei **packages** geschrieben hatte. Die Abhängigkeiten werden automatisch aufgelöst.

Achtung: Man sollte natürlich nicht verschiedene Ubuntu-Versionen mischen. Pakete von

Ubuntu 9.04 „Jaunty Jackalope“ haben nichts auf Ubuntu 9.10 „Karmic Koala“ verloren und umgekehrt.

Schlussbemerkung

Noch ein wichtiger Hinweis zum Schluss: Das hier vorgestellte Skript arbeitet fern von perfekt und erkennt viele Dinge nicht. So wird weder auf Veroderungen noch auf Konflikte eingegangen.

Dies kann in manchen Fällen dazu führen, dass die Installation bzw. das Herunterladen der gefilterten Pakete nicht ohne Weiteres möglich ist:

```
Die folgenden Pakete haben nicht \
erfüllte Abhaengigkeiten:
debconf-english: Kollidiert: \
debconf-i18n aber 1.5.23ubuntu2 \
soll installiert werden
debconf-i18n: Kollidiert: debconf\
-english aber 1.5.23ubuntu2 soll \
installiert werden
E: Kaputte Pakete
```

Hier muss man die Paketliste **packages-all** am Ende noch einmal manuell editieren, bevor man die Pakete installieren kann, da im obigen Beispiel die beiden Pakete **debconf-english** und **debconf-i18n** in Konflikt stehen und nur eines der beiden installiert werden darf.

Eine Verbesserung des Skriptes bleibt dem geneigten Leser zur Übung überlassen – oder ist vielleicht irgendwann Teil eines neuen Artikels.

apt-cache depends hat im Übrigen bereits einen Parameter **--recurse**, dessen Verwendung diesem Artikel aber jede Grundlage entzogen hätte. ;)

```
$ xargs -a packages apt-cache \
depends -i --recurse | \
nawk '{ print $NF }' | sort | \
uniq > packages-all
```

Die Paketanzahl unterscheidet sich auch von der des Skriptes und es ist unklar, wieso einige Abhängigkeiten bei der **apt-cache**-eigenen Methode nicht gelistet werden.

LINKS

[1] <http://www.freiesmagazin.de/freiesMagazin-2009-06>

Autoreninformation

Dominik Wagenführ besitzt auch Rechner ohne Netzwerkanschluss, wozu so ein kleines Skript ganz hilfreich ist, wenn man Pakete installieren will, die nicht auf der Installations-CD vorliegen. Zum Großteil entstand das Skript aber aus Spaß an der Freude.

Diesen Artikel kommentieren





Der Dezember im Kernelrückblick von Mathias Menzer

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

2.6.32 in aller Kürze

Bereits am 2. Dezember veröffentlichte Torvalds den Kernel 2.6.32 [1] nach weniger als drei Monaten Entwicklungszeit. Die Highlights sind diesmal eher unter der Haube zu finden. So sorgt „Per-backend-device based writeback“ für eine bessere Skalierung bei steigender Auslastung des Systems, wenn es um das Schreiben auf blockorientierte Speichergeräte wie zum Beispiel Festplatten geht. Mit „Kernel Samepage Merging“ kann im Bereich der Virtualisierung Speicherplatz eingespart werden. Hostsysteme sind nun in der Lage, Speicherbereiche gleichen Inhalts verschiedener Gäste zusammenzulegen und dadurch Arbeitsspeicher frei zu bekommen, der dann anderweitig genutzt werden kann.

„Timechart“ liefert eine grafische Darstellung des Verlaufs des Startvorgangs als SVG-Datei und soll Entwicklern und interessierten Anwendern bei der Optimierung des Startvorgangs behilflich sein. Weiterhin wurden Verbesserungen unter anderem an den Dateisystemen Ext4 und btrfs vorgenommen. Eine vollständige Übersicht ist auf kernelnewbies.org zu finden [2].

DRBD

Der aktuellste Kernel kam ohne neues Dateisystem daher – das zumindest kann 2.6.33 nicht von sich behaupten. DRBD (Distributed Replicated Block Device) [3] wird schon seit Langem entwickelt, mit der Aufnahme in die Entwicklungsversion des Linux-Kernels ist nun ein Höhepunkt für das Projekt erreicht. Exakt 10 Jahre nachdem der Entwickler Philipp Reisner mit der Programmierung begann, akzeptierte Torvalds den Quellcode für den Kernel [4].

DRBD ist ein verteiltes Dateisystem, das abgelegte Daten über einen Verbund von Rechnern verteilt um den Ausfall einzelner Maschinen zu kompensieren. Es kann als Netzwerk-basiertes Raid-1-System betrachtet werden. Die Aufnahme in den Linux-Kernel bedeutet für Administratoren, dass sie künftig die für die Nutzung von DRBD benötigten Kernel-Module nun nicht mehr selbst erstellen müssen.

Nouveau

Auch an der Grafikfront gab es Bewegung, als der freie Grafiktreiber Nouveau, der NVIDIA-Chips von NV4 bis NV50 unterstützt, aufgenommen wurde. Vorangegangen war eine Diskussion auf der Linux-Kernel-Mailingliste, als sich Torvalds darüber wunderte, dass Red Hat zwar den Nouveau-Treiber standardmäßig mit Fedora ausliefere, diesen jedoch aus den Patches für den Mainline-Kernel entferne [5]. Ein wenig waren die Nouveau-Entwickler von diesem Schritt wohl

überrumpelt, der zu diesem Zeitpunkt noch nicht geplant war, da er nach ihrer Ansicht noch nicht bereit dafür sei. Einer der Gründe liegt wohl darin, dass für einige Grafikkarten noch proprietäre Bestandteile enthalten sind. Vorerst werden diese nun aus dem Treiber selbst entfernt und müssen von externen Quellen nachgeladen werden. Ziel ist jedoch die vollständige Ablösung dieser binären Teile, wobei der Entwickler Ben Skeggs nun einen Erfolg verbuchen konnte. Er veröffentlichte einen selbstentwickelten Ersatz, der auch bereits mit einigen Karten erfolgreich getestet wurde [6].

Ceph

Als zweites neues Dateisystem im Linux-Kernel ist Ceph [7] auf Hochverfügbarkeit ausgerichtet, indem die gespeicherten Daten über mehrere Server verteilt werden. Ceph stellt dabei in erster Linie die Logik der Verteilung der Daten über die in das Cluster eingebundenen Server und nutzt zur Datenspeicherung entweder das eigene EBOFS (Extent-Based Object Filesystem) oder andere auf den jeweiligen Servern verfügbare Dateisysteme. Bislang existierten auf FUSE (Filesystem in Userspace) basierende Clients, diese sollen nun durch einen Kernel-Client ersetzt werden [8].

GKH räumt staging auf

Greg Kroah-Hartman lässt kaum eine Gelegenheit aus, um darauf hinzuweisen, dass der von



ihm gepflegte staging-Bereich des Kernels keine Ablage für schlechte und vernachlässigte Treiber ist [9]. Ein solches Negativbeispiel stellen die Android-spezifischen Treiber dar, die von Google für 2.6.29 eingereicht, jedoch nicht mehr gepflegt werden. Da sich auch sonst niemand darum kümmern möchte, wurden sie nun entfernt.

Das gleiche Schicksal könnte auch dem Treiber für Microsofts Virtualisierungslösung Hyper-V blühen. Dieser wurde erst vergangenen Juli von Microsoft unter die GPL gestellt (siehe „Der Juli im Kernel-Rückblick“, [freiesMagazin](#) 08/2009 [10]) und bereits im September zum ersten Mal von Kroah-Hartman für den baldigen Rauswurf nominiert (siehe „Der September im Kernelrückblick“, [freiesMagazin](#) 10/2009 [11]).

Torvalds wird 40

Den Abschluss des Dezembers bildete dann Torvalds selbst – er wurde am 28. Dezember 40 Jahre alt [12]. Herzlichen Glückwunsch!

LINKS

- [1] <http://lkml.org/lkml/2009/12/3/11>
- [2] http://kernelnewbies.org/Linux_2_6_32
- [3] <http://www.drbd.org/>
- [4] <http://www.pro-linux.de/NB3/news/1/15059/linus-torvalds-nimmt-drbd-in-den-linux-kernel-auf.html>
- [5] <http://www.pro-linux.de/NB3/news/1/15074/freier-vidia-treiber-nouveau-in-den-kernel-2633-aufgenommen.html>
- [6] <http://www.pro-linux.de/NB3/news/1/15084/nouveau-bald-ohne-proprietäre-firmware.html>
- [7] <http://ceph.newdream.net/>
- [8] <http://www.pro-linux.de/NB3/news/1/15067/verteiltes-dateisystem-ceph-vor-der-aufnahme-in-den-standard-kernel.html>
- [9] <http://www.pro-linux.de/NB3/news/1/15118/linux-2633-ohne-android-treiber.html>
- [10] <http://www.freiesmagazin.de/freiesMagazin-2009-08>

- [11] <http://www.freiesmagazin.de/freiesMagazin-2009-10>
- [12] <http://www.linux-magazin.de/NEWS/Linus-Torvalds-feiert-40.-Geburtstag>

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



„Mission“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/452>

Geheimniskrämerei von Ralf Hersel

Jeder hat seine kleinen und größeren Geheimnisse. Seien es Kontoauszüge, persönliche Texte oder die Bilder der ersten Liebe – nicht alles ist für fremde Augen bestimmt. Dieser Artikel befasst sich mit den Möglichkeiten, diese Geheimnisse unter Linux vor ungewollten Blicken zu schützen. Dabei werden gute und weniger geeignete Verfahren betrachtet und zum Schluss eine praxistaugliche Lösung vorgestellt.

Als Anordnung für diesen Test dient ein Ubuntu 9.10 mit ein paar Beispieldateien in einem Verzeichnis mit dem Namen **tresor**. Diese sollen ohne viel Aufwand verschlüsselt und vor fremden Augen versteckt werden.

Verschlüsseln per Kontextmenü

Als erste Maßnahme kommt einem der Befehl „*Verschlüsseln*“ aus dem Kontextmenü eines Verzeichnisses in den Sinn. Damit kann man ganze

Verzeichnisse oder einzelne Dateien mittels GnuPG verschlüsseln. Voraussetzung ist, dass man bereits einen GnuPG-Schlüssel erzeugt hat. Leser, die auf Ubuntu 9.10 aktualisiert haben, wundern sich vielleicht darüber, dass der Befehl „*Verschlüsseln*“ im Datei-Browser nicht mehr vorhanden ist. Um diesen Befehl wieder hervorzuholen, muss das Paket **seahorse-plugins** installiert werden. Mehr Informationen findet man im Wiki von ubuntuusers.de [1].

Wendet man nun den Befehl „*Verschlüsseln*“ auf das Verzeichnis **tresor** an, so muss zuerst der Schlüssel gewählt und entschieden werden, ob jede Datei einzeln verschlüsselt wird oder ein verschlüsseltes Paket erstellt wird.

Während der Verschlüsselung muss das Passwort für den GnuPG-Schlüssel für jede Datei einzeln eingegeben werden. Die Originaldateien werden nach dem Verschlüsseln nicht gelöscht,

das muss man selbst erledigen. Fügt man dem Verzeichnis eine neue Datei hinzu, so muss diese auch wieder verschlüsselt werden; dies geschieht nicht automatisch.

Man erkennt hier schnell, dass dieses Verfahren nicht den gewünschten Zweck erfüllt; zu viele Schritte sind nötig, um zum Ziel zu kommen. Das Verschlüsseln aus dem

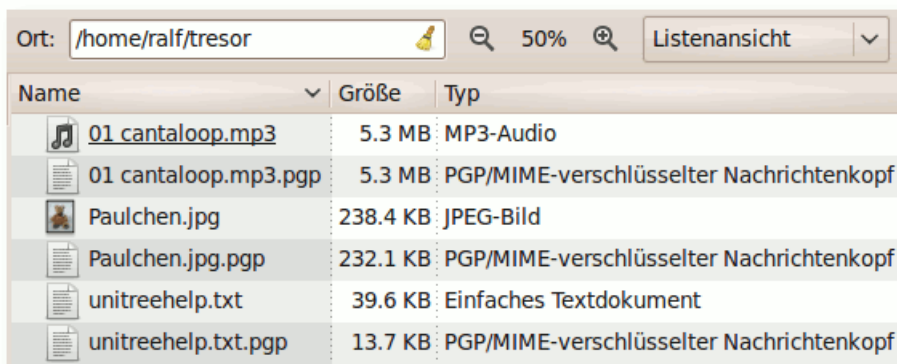
Kontextmenü einer Datei oder eines Verzeichnisses ist zum einmaligen Chiffrieren gedacht – und nicht um ein transparent verschlüsseltes Verzeichnis zu erzeugen.







Verschlüsseln beim Komprimieren

Da man aus Fehlern lernen soll, greift der zweite Versuch eine Erfahrung aus dem ersten auf. Beim Verschlüsseln wurde angeboten, die Dateien in ein Paket (Archiv) zu stecken. Diese Idee wird nun getestet.

Wenn man aus dem Kontextmenü eines Verzeichnisses den Befehl „*Komprimieren*“ wählt, erscheint ein Dialog. Dort gibt es die Möglichkeit, ein Passwort für die Verschlüsselung einzugeben; man beachte die angekreuzte Option „*Dateiliste ebenfalls verschlüsseln*“. Als Ergebnis erhält man ein ZIP-Archiv, das die Dateien des Verzeichnisses **tresor** in verschlüsselter Form enthält.

So weit, so gut. Was geschieht aber, wenn dem Archiv weitere Dateien hinzugefügt werden sollen? Zieht man eine neue Datei mit der Maus in das Archiv („*Datei zum Archiv hinzufügen*“), so wird die Datei unverschlüsselt in das Archiv gepackt. Um zusätzliche Dateien beim Einfügen verschlüsseln zu lassen, muss zuerst das Archiv geöffnet werden. Dort gibt man im Menü „*Bearbeiten*“ das Passwort für das Archiv ein. Erst dann werden Neuzugänge im Archiv verschlüsselt.



Name	Größe	Typ
 01_cantaloop.mp3	5.3 MB	MP3-Audio
 01_cantaloop.mp3.gpg	5.3 MB	PGP/MIME-verschlüsselter Nachrichtenkopf
 Paulchen.jpg	238.4 KB	JPEG-Bild
 Paulchen.jpg.gpg	232.1 KB	PGP/MIME-verschlüsselter Nachrichtenkopf
 unitreehelp.txt	39.6 KB	Einfaches Textdokument
 unitreehelp.txt.gpg	13.7 KB	PGP/MIME-verschlüsselter Nachrichtenkopf

Inhalt des verschlüsselten Archivs. 



Ein verschlüsseltes Archiv wird erstellt. 🔍

Wie beim ersten Versuch ist auch hier die gewünschte Unterstützung noch nicht optimal. Man darf nicht vergessen, die Originaldateien zu löschen, nachdem sie in das Archiv gelegt wurden. Außerdem muss der Archivmanager als zusätzliches Werkzeug bedient werden. Entgegen der Vermutung wird die Archivliste nicht verschlüsselt; dadurch kann jeder die Dateinamen im Archiv lesen. Einen Vorteil hat die Archivmethode gegenüber dem ersten Verfahren: Die Dateien werden nicht nur verschlüsselt, sondern auch komprimiert.

Verschlüsseln per EncFS

Auf der Suche nach der besten Methode zur Verschlüsselung von Verzeichnissen fand sich schließlich EncFS. Das „Encrypted File System“ [2] ist eine Verschlüsselungserweiterung für unixartige Dateisysteme. Mit EncFS kann ein Benutzer innerhalb kürzester Zeit und ohne großen Aufwand verschlüsselte Verzeichnisse erstellen.

Verschlüsselte Bereiche werden mit Passwörtern geschützt und in versteckten Ordnern abgelegt.

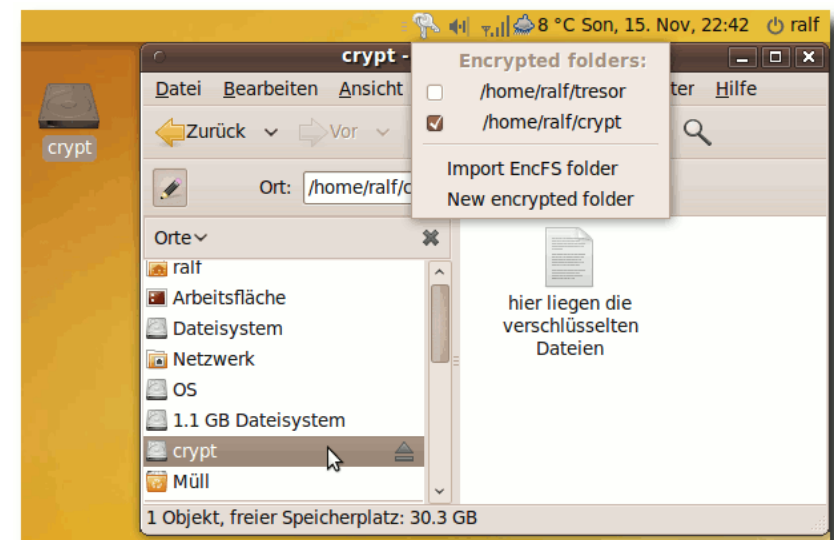
EncFS kann aus der Paketverwaltung installiert werden, dazu werden die Pakete **encfs** und evtl. **cryptkeeper** benötigt. Nach der Installation muss man überlegen, ob man das verschlüsselte Verzeichnis bei jeder Sitzung oder nur gelegentlich braucht. Im ersten Fall empfiehlt es sich, Cryptkeeper in die Startprogramme mit aufzunehmen, ansonsten kann man sich im Menü oder Panel einen Starter erzeugen, der das Programm **cryptkeeper** ausführt. Dadurch erhält man im Panel ein Schlüsselsymbol, das das Öffnen und Schließen eines EncFS-Verzeichnisses ermöglicht.

Die Steuerung der EncFS-Verzeichnisse erfolgt mittels Cryptkeeper. Hier können nun die verschlüsselten Verzeichnisse geöffnet bzw. geschlossen werden, man kann bestehende EncFS-Verzeichnisse importieren oder neue erstellen. Im Beispiel wurde das bestehende Verzeichnis **tresor** importiert.

Wird ein neues EncFS-Verzeichnis erstellt („*New encrypted folder*“), gibt man als Erstes einen Namen für das Verzeichnis ein, klickt auf „*Forward*“, gibt dann zweimal ein Passwort ein und sieht anschließend das geöffnete Verzeichnis in der Dateiverwaltung.

Cryptkeeper zeigt durch den Haken vor dem Verzeichnispfad an, dass **crypt** nun geöffnet ist. Das Schließen bzw. Aushängen des Verzeichnisses muss über **cryptkeeper** geschehen (Haken vor dem Verzeichniseintrag abwählen), da ein Aushängen in der Dateiverwaltung aufgrund fehlender Root-Rechte scheitert.

Aber was geschieht hier im Hintergrund? EncFS erzeugt in Wirklichkeit zwei Verzeichnisse: **crypt** und **.crypt_encfs**, also ein sichtbares und ein verstecktes Verzeichnis. Im versteckten Verzeichnis befinden sich die verschlüsselten Dateien, während im sichtbaren Verzeichnis die entschlüsselten Dateien zu sehen sind, wenn **.crypt_encfs** eingehängt ist. **cryptkeeper** entfernt das Verzeichnis, wenn **.crypt_encfs** ausgehängt wird.



Ein neues EncFS-Verzeichnis ist eingehangen. 🔍



Für Übereifrige oder Vergessliche hält **cryptkeeper** noch ein besonderes Schmankerl parat: Es kann eingestellt werden, nach wie vielen untätigen Minuten das Verzeichnis automatisch ausgehängt wird. Ob das leere Verzeichnis danach gelöscht wird oder nicht, kann man auch hier angeben.

Individualisten oder Konsolenfreunde können auf **cryptkeeper** getrost verzichten. Wer nur das Paket **encfs** installiert, kann die Geheimniskrämerei auch über die Konsole oder über ein kleines Shellskript steuern, so wie es im folgenden Beispiel gezeigt wird.

```
#!/bin/bash
# mount or unmount the encfs
# folder .tresor to tresor
#
# check if it is already
# mounted or not
a=$(mount | grep tresor | wc -l)
)
if [ $a -eq 0 ]; then
    echo "mounting tresor,
    enter: y, y, p, password"
    encfs ~/.tresor ~/tresor
else
    echo "unmounting tresor"
    fusermount -u ~/tresor
fi
```

Listing 1: *Encfs-mount.sh*

Das Skript ermittelt in Zeile 5, wie viele Verzeichnisse mit dem Namen „tresor“ eingehängt sind. In Zeile 6 wird geprüft, ob überhaupt ein solches Verzeichnis eingehängt ist. Falls noch nichts eingehängt wurde, wird in Zeile 8 das EncFS-Verzeichnis **.tresor** in **tresor** eingehängt. Falls dies bereits der Fall ist, wird in Zeile 11 das Verzeichnis ausgehängt.



Cryptkeeper-Einstellungen. 🔍

Verbindet man dieses Skript mit einem Starter, kann mit einem Mausklick der Tresor geöffnet bzw. geschlossen werden.

Der Name „tresor“ steht natürlich stellvertretend für einen beliebigen Verzeichnisnamen. Im Gegensatz zu **cryptkeeper** wird das ausgehängte Verzeichnis nicht entfernt, es ist einfach leer. In dieses leere Verzeichnis darf man jedoch keine Dateien hinein kopieren, sonst weigert sich EncFS, das versteckte Verzeichnis einzuhängen. Wer außerdem das automatische Aushängen ein-

schalten möchte, benutzt die Option **--idle=n** in Zeile 8, wobei **n** für die Minuten steht.

Fazit

Der dritte Versuch zeigt ein Verfahren auf, mit dem Verzeichnisse transparent verschlüsselt werden können. Die Verwendung des Encrypted File Systems in Verbindung mit Cryptkeeper oder einem einfachen Shellskript erlaubt das bequeme Öffnen und Schließen eines verschlüsselten Verzeichnisses. Weder müssen die Originaldateien nachträglich gelöscht werden noch muss man für jede einzelne Datei ein Passwort eingeben. Die Methode folgt der simplen Metapher „Tresor auf, Dateien rein, Tresor zu“. Damit ist es für den alltäglichen Umgang mit sensiblen Daten besonders geeignet.

LINKS

- [1] <http://wiki.ubuntuusers.de/Seahorse>
- [2] <http://wiki.ubuntuusers.de/EncFS>

Autoreninformation

Ralf Hersel war seit 2007 Autor des Yalm-Magazins. Seine Artikel richten sich in erster Linie an Ein- und Umsteiger auf Linux. Dadurch möchte er seinen Beitrag zur Verbreitung von offener Software leisten.

Diesen Artikel kommentieren ☁️



Secure Copy (SCP) per Mausklick in KDE von Markus Schulze

Egal ob Wohngemeinschaft, Familie oder Single: In immer mehr Haushalten existieren zwei PCs oder gar mehr. Doch was tun, wenn man lediglich Dateien von Gerät A nach Gerät B kopieren will? Und was, wenn man das nach Möglichkeit komfortabel per Mausklick und ohne Passwordeingabe machen will? Vielleicht sogar mit sicherer, also verschlüsselter Übertragung?

Heißt das Betriebssystem Linux und nutzt man als Desktopumgebung KDE, so kann man mit Hilfe dieser Anleitung ein Servicemenü für Konqueror & Dolphin erzeugen, mit dem man zukünftig einfach per Mausklick eine Datei an einen festgelegten Ort eines anderen Rechners kopiert.

Public-Key-Authentifizierung

Um nicht permanent die Anmeldedaten des Zielrechners eingeben zu müssen, muss man in einem ersten Schritt die SSH-Authentifizierung ohne Passwordeingabe konfigurieren. Erst wenn dies geschehen ist, kann nach Lust und Laune mittels SCP (Secure Copy) zwischen den Rechnern kopiert werden. Damit ist die Voraussetzung geschaffen, den Kopiervorgang zu skripten. Dies ist umso mehr nötig, da das Servicemenü in KDE3 genau wie in KDE4 auf ein Programm (z. B. ein Shell-Skript) verweisen muss, welches die gewünschten Aktivitäten ausführt.

Bei einer normalen SSH-Verbindung wird zur Authentifizierung ein Passwort abgefragt. Um dies

umgehen zu können, authentifiziert man sich nicht mehr per Passwort, sondern mittels eines Schlüsselpaares. Dazu wird an Rechner A ein Schlüsselpaar erzeugt und der öffentliche Schlüssel dem Rechner B übergeben. Mit entsprechenden Einstellungen in der `ssh.config` kann jetzt eine sichere Verbindung ohne lästige Passwordeingabe aufgebaut werden.

Zunächst wechselt man in der Konsole am Rechner A in das versteckte Verzeichnis `.ssh` des persönlichen Ordners. Dort tippt man anschließend:

```
$ ssh-keygen -b 2048 -t rsa
```

womit das Schlüsselpaar erzeugt wird (während der Erstellung solange `Enter` drücken bis man wieder beim Eingabeprompt landet). Im Ordner `.ssh` gibt es nun eine Datei `id_rsa` (enthält den privaten Schlüssel) und eine `id_rsa.pub` (enthält den öffentlichen Schlüssel). Letztere muss noch auf Rechner B kopiert werden, etwa so:

```
$ scp ~/.ssh/id_rsa.pub <user_b>@<rechner_b>:~/home/<user_b>/.ssh/
```

Nun wechselt man zu Rechner B und importiert den Inhalt von `id_rsa.pub` in die Datei `authorized_keys` mittels

```
$ cat id_rsa.pub >> authorized_keys
```

und löscht anschließend die Datei `id_rsa.pub` wieder.

Außerdem muss man als root noch einmal überprüfen, ob die Anmeldung via Public Key in der Konfiguration überhaupt erlaubt ist (bei openSUSE ist dies beispielsweise standardmäßig nicht der Fall). Dazu dürfen folgende Zeilen in der `/etc/ssh/sshd_config` nicht auskommentiert sein:

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
```

Hat man all diese Punkte beachtet, sollte man sich jetzt von Rechner A auf Rechner B ohne Passwordeingabe anmelden können und somit auch Dateien ohne Passwortabfrage sicher kopieren dürfen. Eine kurze Überprüfung mittels

```
$ ssh <user_b>@<rechner_b>
```

schafft Gewissheit.

Will man Dateien in beide Richtungen ohne Anmeldung kopieren, muss man die oben genannten Befehle entsprechend noch einmal umgekehrt ausführen. Weitere Hinweise zur Public-Key-Authentifizierung gibt es bei Linupedia [1].

SCP-Skripte

Jetzt benötigt man noch ein Skript, welches beim Klick auf den Servicemenüeintrag ausgeführt wird. Als Beispiel soll ein einfacher SCP-Aufruf dienen:



```
#!/bin/bash
scp "${1}" "${2}"
```

Die beiden Eingabeparameter `${1}` und `${2}` repräsentieren die Quelle und das Ziel und werden von Konqueror und Dolphin an das Skript übergeben.

Für umfangreichere Skripte, welche durch Gebrauch von KDialog auch aussagefähiger werden (z.B. mit Fortschrittsbalken, Anzeige von Quelle und Ziel usw.), sei an dieser Stelle auf die beiden Skripte **SecureFileCopy.sh** und **SecureFolderCopy.sh** verwiesen. Es gibt sie für KDE3 [2] und KDE4 [3].

Servicemenüeintrag in KDE3

Was jetzt noch fehlt, ist der Eintrag im Servicemenü, welcher auf das Beispielskript verweist und beim Mausklick ausgeführt wird.

```
[Desktop Entry]
Encoding=UTF-8
X-KDE-Submenu=Datei an anderen PC senden
ServiceTypes=all/allfiles
Actions=MoveToHomeShare1;
ExcludeServiceTypes=kdedevice/*

[Desktop Action MoveToHomeShare1]
Name=openSuse11.2@SamsungQ45 (~/HomeShare)
Exec='/usr/share/apps/konqueror/servicemenus/scripts/SecureFileCopy.sh' %u 'MyUser1@111.222.333.1:/home/MyUser1/HomeShare'
Icon=folder_html
```

Listing 1: SCP.desktop

Zum grundlegenden Aufbau eines Servicemenüs findet man auf der Website von KDE im Entwicklerbereich eine ausführliche Dokumentation [4]. Auch ohne diese Einführung sollte jedoch für die meisten Leser der Code der **SCP.desktop** verständlich sein.

Das Entscheidende steht hier in der vorletzten Zeile: Das Skript **SecureFileCopy.sh** wird ausgeführt und die gewählte Datei (`%u`) und der Zielort (`'MyUser1@...'`) werden an das Skript übergeben. Das sind genau die beiden Eingabeparameter, von denen im vorletzten Absatz im Abschnitt „SCP-Skripte“ gesprochen wurde.

Abschließend verschiebt man die Datei **SCP.desktop** nach `/usr/share/apps/konqueror/servicemenus` und achtet darauf, dass alle Rechte richtig gesetzt sind (**SCP.desktop**: 644, Skripte: 755). Außerdem müssen alle relevanten Dateien dem User und der Gruppe **root** angehören.

Servicemenüeintrag in KDE4

Bei KDE4 verhalten sich die Dinge (warum auch immer) ein klein wenig anders. Zunächst muss die Zeile **ServiceTypes** in **SCP.desktop** geändert werden und eine zusätzliche Zeile mit **Service** eingefügt werden:

```
ServiceTypes=KonqPopupMenu/Plugin,~
all/allfiles
Type=Service
```

Da man bei KDE4 die **SCP.desktop** nach `/usr/share/kde4/services/ServiceMenus` kopieren muss, bedarf es auch einer Anpassung der Zeile **Exec**:

```
Exec='/usr/share/kde4/services/~
ServiceMenus/~
SecureFileCopyAcerAspire3000.sh' %u
```

Aus irgendeinem Grund gelingt es nicht, in KDE4 einen zweiten Parameter an das Skript zu übergeben. Die zugegebenermaßen nicht sehr elegante Lösung besteht darin, auf unterschiedliche Skripte zu verweisen und den Zielrechner fest im Skript zu hinterlegen. Man braucht also pro Rechner ein Skript. **SecureFileCopyAcerAspire3000.sh** sieht demnach so aus:

```
#!/bin/bash
T0='MyUser2@333.222.111.1:/home/~
MyUser2/HomeShare'
scp "${1}" $T0
```

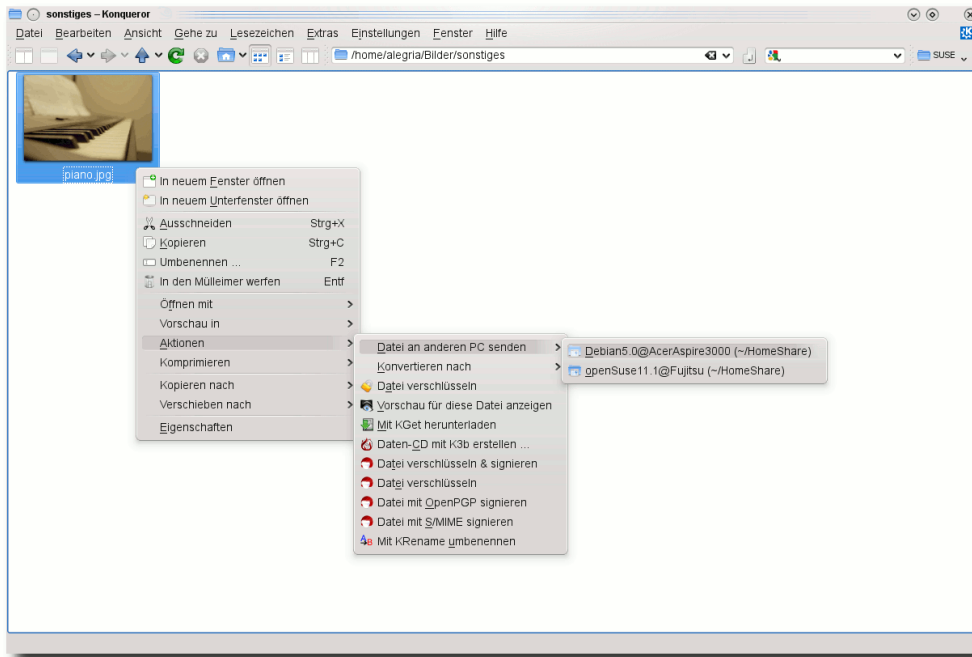
Fazit

Hat man dies alles berücksichtigt, erscheint ab sofort in Konqueror und Dolphin beim Rechtsklick auf eine beliebige Datei der gewünschte Servicemenüeintrag „Datei an anderen PC senden“, in einer Ebene darunter kann nun der entsprechenden Eintrag zum Senden der Datei auf den gewünschten Rechner ausgewählt werden.



Im diesem Beispiel ist das für KDE3 der Menüeintrag „*openSuse11.2@SamsungQ45 (~/HomeShare)*“, der bei einem Mausklick das Skript **SecureFileCopy.sh** mit den richtigen Parametern aufruft und die Datei in den Ordner **HomeShare** des Users **MyUser1** auf dem Rechner **111.222.333.1** kopiert.

In KDE4 wird hingegen nach einem Klick auf den Menüeintrag „*Debian5.0@AcerAspire3000 (~/HomeShare)*“ das, für diesen Rechner angepasste, Skript **SecureFileCopyAcerAspire3000.sh** ausgeführt und die Datei in den Ordner **HomeShare** des Users **MyUser1** auf dem Rechner **333.222.111.1** kopiert.



Servicemenüeintrag in KDE4. 🔍

Ausblick

Für den ambitionierten Leser noch einige Anregungen für mögliche Weiterentwicklungen:

Was passiert wenn man Dateien und Ordner zusammen markiert und kopieren will bzw. wie setzt man diese Konstellation technisch um?

Momentan ist der Zielort fix. Wie könnte man eine grafisch ansprechende Auswahl des Zielortes erreichen ohne bei der Performance Einbußen zu haben?

Die beiden als Download zur Verfügung gestellten Skripte verwenden die Progressbar rein zeitgesteuert und nicht mengengesteuert. Wie könnte man hier Verbesserungen erzeugen?

Denkbar wäre auch ein einfaches Installationsprogramm und/oder Paket, welches einige Variablen entgegen nimmt und anschließend die Skripte dynamisch erzeugt, ausführt und die Dateien entsprechend in die Ordner kopiert. Mit welchem Tool und welcher Programmiersprache erreicht man

dies am besten, oder macht ggf. ein einfaches Konqueror-Plugin mehr Sinn?

Ideen oder Lösungsvorschläge nehme ich gerne unter **marcos.i-networx[AT]gmx[DOT]de** entgegen.

Diese Funktionen lassen sich unter GNOME mit den „Nautilus-Skripts“ realisieren („Nautilus-Scripts“, **freiesMagazin** 11/2009 [5]).

LINKS

- [1] http://www.linupedia.org/opensuse/Einrichten_von_public_keys_mit_ssh#Publickey-Authentifizierung
- [2] <http://marcos.i-networx.de/freiesMagazin/KDE3/>
- [3] <http://marcos.i-networx.de/freiesMagazin/KDE4/>
- [4] <http://developer.kde.org/documentation/tutorials/dot/servicemenus.html> 
- [5] <http://www.freiesmagazin.de/freiesMagazin-2009-11>

Autoreninformation



Markus Schulze ist seit 2006 von Linux und freier Software angetan und arbeitet mit openSuse, Debian und Kubuntu. Als großer Fan von KDE versucht er auch ohne Kenntnisse in C++ die eine oder andere sinnvolle Erweiterung am System umzusetzen.

[Diesen Artikel kommentieren](#) 

GIMP in 90 Minuten (kennenlernen) von Karsten Günther

Das GNU-Bildbearbeitungsprogramm *GIMP* [1] gilt bei Anwendern, die nicht regelmäßig damit arbeiten, als kompliziert, erschlagend umfangreich und vielfach unverständlich. Da mutet der Versuch, die Grundlagen von *GIMP* in 90 Minuten zu vermitteln, schon fast verwegend an. Dass das aber möglich ist, zeigt dieser Workshop, der auf der Ubucon 2009 gehalten wurde (siehe „Rückblick: Ubuntu-Treffen auf der Ubucon 2009“, freiesMagazin 11/2009 [2]).

Redaktioneller Hinweis: Das Beispielbild in diesem Artikel heißt „WildFlower Girl“ [3] und wurde von Pink Sherbet Photography [4] unter der Creative-Commons-Attribution-Lizenz [5] veröffentlicht. Die anderen Bilder stammen vom Autor Karsten Günther.

GIMP starten

Die drei wichtigsten Methoden, *GIMP* zu starten, sind:

- Aus dem „Anwendungen“-Menü des Desktops (unter „Grafik“). *GIMP* startet ohne Bild.
- Aus dem Dateimanager. Dort bietet das Kontextmenü (über die rechte Maustaste) bei Bildern meistens eine Option der Form „Bild mit *GIMP* laden“ an. Vorteil: Zunächst kann das richtige Bild in der Symbolansicht gefunden werden.
- Von der Befehlszeile (mit Optionen) über die Eingabe von `gimp BILDDATEINAME`. Vorteil:

GIMP zeigt hier, ob es beim Laden von Plugins oder Bildern Probleme gibt.

Bereits zuvor bearbeitete Bilder verwaltet *GIMP* in einer History, dem Dokumentenindex, den man unter „Datei → Zuletzt geöffnet“ findet. Die Bilder lassen sich von dort direkt oder per Tastenkombination (`Strg` + `1` bis `Strg` + `0`) laden.

Eine Anmerkung zum Speichern: Nur das *GIMP*-Format „xcf(.bz2)“ speichert alle Bildinformationen. Das Speichern in anderen Formaten „exportiert“ die momentan sichtbaren Bildteile,

erhält dabei aber besondere Funktionen wie Ebenen, Auswahlen usw. nicht. Das Originalbild verändert *GIMP* bei einem Export nicht.

Und noch etwas ist gut zu wissen: `Strg` + `Z` (Rückgängig/Undo) hebt den letzten Bearbeitungsschritt auf. Voreingestellt geht das mindestens fünfmal, mehr lässt sich im Menü „Bearbeiten“ unter „Einstellungen“ festlegen. Einen unbeabsichtigt zuviel rückgängig gemachten Schritt stellt `Strg` + `Y` wieder her.

Die Fenster im Überblick

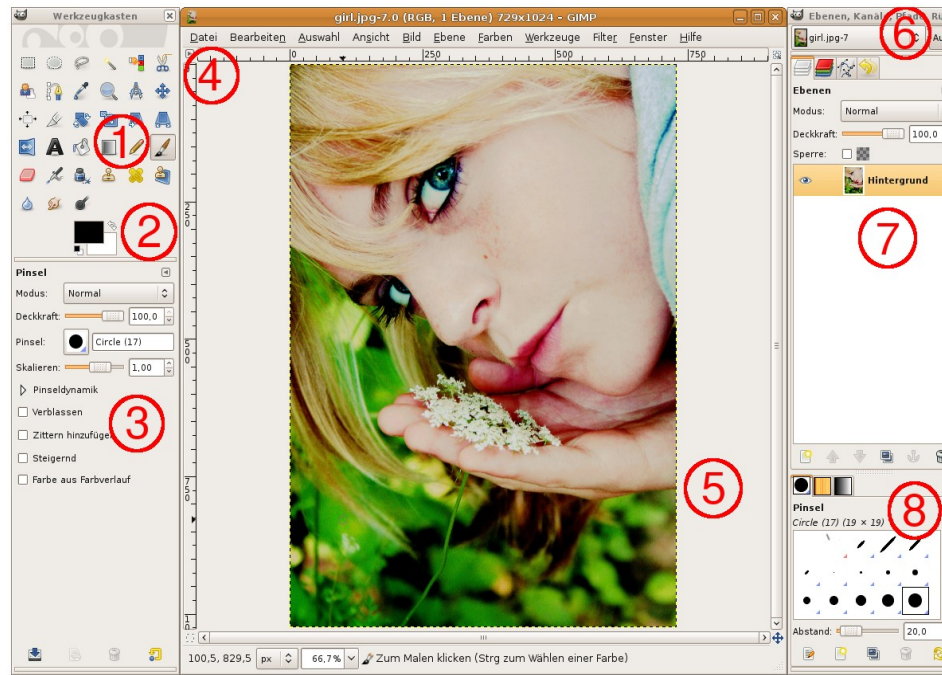
*GIMP*s Fenster sind wie folgt aufgebaut:

1. Werkzeuge
2. Einstellungen
3. Werkzeugoptionen
4. Lineale
5. Haupt(Bild-)fenster
6. Bildauswahlfenster
7. Oberes Dockfenster
8. Unteres Dockfenster

Alle Fenster weisen bestimmte Besonderheiten auf.

Der Werkzeugkasten

Der Werkzeugkasten enthält die von den Entwicklern als am wichtig-



*GIMP*s Fenster im Überblick. 🔍



sten angesehen Funktionen in Form von „Werkzeugen“. Die Werkzeuge lassen sich in mehrere Gruppen zusammenfassen:

- Die „Auswahlwerkzeuge“ markieren bestimmte Bildbereiche zur weiteren Bearbeitung.
- Das „Pfadwerkzeug“ erstellt gerichtete Linien (mit einem Anfangs- und Endpunkt).
- Die „Messwerkzeuge“ sind beim Abmessen von Strecken oder Winkeln im Bild hilfreich.
- Mit den „Translationswerkzeugen“ wird bei der Bearbeitung der (restliche) Bildinhalt ohne Qualitätsverlust erzeugt.
- Im Gegensatz dazu stehen die „Transformationswerkzeuge“, die bei einem einmaligen Einsatz meisten keinen (sichtbaren) Qualitätsverlust erzeugen, wohl aber bei mehrfacher Anwendung. Diese Werkzeuge sollten daher immer nur einmal angewendet werden: Anstatt also ein Bild erst um 2,5 und dann um weitere 3,7 Grad zu drehen, sollte es besser nur einmal um 6,2 Grad gedreht werden.
- Das „Textwerkzeug“ ist für das Einfügen von Textboxen zuständig.
- Die „Malwerkzeuge“ malen mit ganz unterschiedlichen Methoden im Bild und verändern es dabei.

Jedes dieser Werkzeuge verfügt über Optionen – also Schalter – die es erlauben, bestimmte Eigenschaften fein abzustimmen.

Im unteren Teil des Werkzeugkastens gibt es noch die Möglichkeit, zwei Farben (Vordergrund-

und Hintergrundfarbe) einzustellen, sowie sie schnell zu vertauschen bzw. auf die Voreinstellungen zurückzusetzen.

Achtung: Die Optionen von Werkzeugen müssen vor ihrem Einsatz aktiviert werden, um die gewünschte Wirkung zu erzielen.

Das Bildfenster

Sofern im obersten Dockfenster der Schalter „Auto“ aktiviert ist, zeigt das Bildfenster stets das aktuell bearbeitete Bild. Es dient der Bearbeitung und sollte möglichst groß sein. Mit der **Tab**-Taste lassen sich Werkzeugkasten und Dock jederzeit schließen bzw. wieder öffnen.

Die Darstellungsgröße im Bildfenster lässt sich schnell mit den Tasten **+** und **-** umschalten. Ein Mausrad macht das zusammen mit der **Strg**-Taste. Die gedrückte mittlere Maustaste erlaubt es, den sichtbaren Ausschnitt im Bildfenster zu verschieben. Mit gedrückter Leertaste und der Mausbewegung erhält man das gleiche Ergebnis.

Die Titelleiste enthält den Dateinamen, Größe, Farbmodus usw. Ein Sternchen davor weist auf ungesicherte Änderungen hin.

Zwei Lineale am linken bzw. oberen Bildrand enthalten Hilfslinien, die sich mit der Maus herausziehen lassen. Hilfslinien verschwinden, wenn sie außerhalb des Bildes geschoben werden.

Die Statuszeile am unteren Bildrand enthält Informationen über die aktuelle Aktion und über die aktuelle Ebene.

In den vier Ecken des Fensters befinden sich Buttons. Sie öffnen das Menü, Skalieren die Ansicht, aktivieren die Schnellmaske bzw. zeigen den Bildnavigator.

Innerhalb des Bildfensters gibt es ein Kontextmenü; die dort enthaltenen (Unter-)Menüs können auch „abgehängt“ werden.

Das Dockfenster

Im Dockfenster auf der rechten Seite lassen sich spezielle, kleine Infofenster andocken. Es enthält voreingestellt drei Fenster:

1. Ein Bildauswahlfenster, welches festlegt, für welches Bild die anderen Fensterinformationen zeigen.
2. Das obere Dockfenster. Es ist einstellbar und enthält Dialoge, oft Ebenen, Kanäle, Pfade und das Journal.
3. Das untere Dockfenster enthält ebenfalls einstellbare Dialoge, oft Pinsel, Muster, Farbverläufe etc.

Die meisten Dialogfenster haben am unteren Rand Buttons für die wichtigsten und Kontextmenüs mit den anderen relevanten Funktionen.

Es ist ganz einfach, zusätzliche Dialoge in ein Dockfenster aufzunehmen: Ein Mausklick in das kleine Dreieck rechts oben neben dem Titel öffnet ein Menü. Damit lassen sich über „Reiter hinzufügen“ zusätzliche Dialoge ins Dock aufnehmen, etwa den Auswahleditor, die Zeigerinformationen, den Dokumentenindex, die Fehlermeldungen, das Histogramm, die Ablagen, die Schriften



usw. Einige besonders oft verwendete Dialoge lassen sich in beiden Dockfenstern vorhalten.

Auswahlen

Auswahlen markieren Bildbereiche zur späteren Bearbeitung. Wie dies aussieht, hängt von den dann eingesetzten Werkzeugen (oder Filtern) ab. *GIMP* bietet im Werkzeugkasten sieben Auswahlwerkzeuge mit unterschiedlichen Eigenschaften. Alle Werkzeuge lassen sich durch Optionen im Detail steuern. In den meisten Fällen werden Auswahlen nicht in einem Schritt, sondern in mehreren Stufen erstellt. Eine Grobauswahl wird verfeinert, ergänzt oder es wird Überflüssiges entfernt.

Alle Auswahlwerkzeuge haben besondere Eigenschaften und Einsatzzwecke:

- Die Rechteckauswahl und die Ellipsenauswahl sind einfach, elegant und erzeugen schnell geometrische Formen.
- Die Freihandauswahl ist flexibel, nutzt das Auge zur Kantenfindung und ist daher gut für „logische“ Auswahlen geeignet. Sie hat zwei Modi: freihändig (bei gedrückter linker Maustaste) und den Linienmodus durch (linke) Mausklicks. Dieser Modus ist besonders schnell und oft sehr effektiv.
- Der Zauberstab (die lokale Farbauswahl) ist sehr gut zur Verbesserung von bestehende Auswahlen geeignet. Die Schwelle steuert, wie weit die in die Auswahl aufgenommenen Farben von der am Mausklickpunkt abweichen dürfen. Sie lässt sich interaktiv mit der linken

Maustaste anpassen, indem man diese festhält und die Maus nach rechts bzw. links bewegt. Der Zauberstab wählt immer zusammenhängende Bildbereiche aus.

- Anders wirkt die globale Farbauswahl. Unabhängig vom Klickpunkt der Maus wählt sie farblich ähnliche (gesteuert durch die Schwelle, die sich auch interaktiv anpassen lässt) Bereiche im gesamten Bild aus. Das ist allerdings nur selten sinnvoll, denn sie erzeugt eine weiche Auswahl, oft mit Löchern.
- Die magnetische (oder „intelligente“) Schere ermittelt Kanten automatisch, oft sehr gut, manchmal auch gar nicht. Die mit ihr erzeugte Linie ist zunächst noch keine Auswahl; es lassen sich zusätzliche Stützpunkte anbringen (oder Stützpunkte verschieben), bis die Linie den letzten mit dem ersten Punkt verbindet und sie mit `Return` abgeschlossen wird.
- Die Vordergrundauswahl ist ein modernes Grobauswahlverfahren. Sie kombiniert Freihandauswahl mit der (globalen) Farbauswahl. Noch ist das Werkzeug nicht wirklich ausgereift, denn es erfordert noch sehr viel Nachbearbeitung.

Achtung: Vor dem Auswählen sollten immer die Optionen der Werkzeuge kontrolliert werden, um auch wirklich die gewünschten Ergebnisse zu erzielen. So ist die voreingestellte Option „*Kanten glätten*“ wohl in den meisten Fällen nicht sinnvoll. Viele der Optionen sind in bestimmten Situationen nützlich einsetzbar, das Buch „*GIMP – kurz & gut*“ beschreibt sie im Detail.

Eine Besonderheit bei Auswahlwerkzeugen ist die Option „*Kanten ausblenden*“. Sie erlaubt, Auswahlen zu erzeugen, die kontinuierlich von 100% bis 0% abnehmen, also nicht unmittelbar abbrechen. Bei diesen als „weich“ oder „unscharf“ bezeichnete Auswahlen zeigt die Ameisenlinie an, wo Pixel zu genau 50% ausgewählt sind. Die eine Hälfte der Auswahl liegt dann innerhalb, die andere außerhalb der Linie.



Die weiche Auswahl. 🔍



Neben den sieben im Werkzeugkasten vorhandenen Auswahlwerkzeugen gibt es noch die Schnellmaske: Sie erzeugt Auswahlen durch Malwerkzeuge und erlaubt so oft, besondere Effekte zu erzielen. Auf diese Funktion, die man über „Auswahl → Schnellmaske umschalten“ oder **Umschalt** + **Q** erreicht, wird hier aber nicht näher eingegangen.

Auswahlen kombinieren

Alle Auswahlwerkzeuge haben vier Modi. Diese legen fest, was geschieht, wenn das Werkzeug angewendet wird, obwohl schon eine Auswahl vorhanden ist:

- Voreingestellt ist „Aktuelle Auswahl ersetzen“, also eine neue Auswahl erzeugen. Die vorige verschwindet, **Strg** + **Z** erlaubt sie zu restaurieren.
- Der Modus „Auswahl ergänzen“ bewirkt, dass GIMP die alte Auswahl um die neue, jetzt erzeugte ergänzt. Die **Umschalt**-Taste aktiviert diesen Modus.
- Genau umgekehrt (entfernt also die neue von der bestehenden Auswahl) wirkt der Modus „von Auswahl abziehen“; **Strg** aktiviert den Modus.
- Ein Spezialfall stellt „Schnittmenge der Auswahlen bilden“ dar. Nur die in der bestehenden und der neuen Auswahl vorhandenen Teile bleiben ausgewählt.

Manchmal ist es einfacher, das zu markieren, was nicht ausgewählt werden soll, statt eine Auswahl auf den gewünschten Bereich zu tätigen.

Dann wird so vorgegangen: Die unerwünschten Bereichen werden ausgewählt, anschließend wird die Auswahl durch **Strg** + **I** invertiert. Sie



Nach der Invertierung der Maske und **Entf** erhält man den ausgewählten Bereich. 

enthält nun die zuvor nicht ausgewählten Bereiche. Und falls die Ameisenlinie stört: **Strg** + **T** lässt sie verschwinden, ohne die Auswahl zu löschen. **Strg** + **A** wählt alles aus, **Strg** + **Umschalt** + **A** löscht die Auswahl.

Auswahlen testen

Wann immer etwas ausgewählt wurde, ist es wichtig zu wissen, was nun genau ausgewählt ist, bzw. wie es aussieht. **Entf** löscht die aktuelle Auswahl aus dem Bild und zeigt so, ob eventuell etwas vergessen wurde. Umgekehrt zeigt **Strg** + **I** gefolgt von **Entf** das „freigestellte“ Objekt. Wie immer machen **Strg** + **Z** (und eventuell **Strg** + **Y**) die Löschaktionen rückgängig.

Auswahlen modifizieren

Das „Auswahl“-Menü enthält eine Reihe interessanter Funktionen, um bestehende Auswahlen zu verändern. Sie lassen sich

- vergrößern und verkleinern (beispielsweise um Löcher zu schließen).
- verweichen („Ausblenden“) und schärfen
- verzerren (beispielsweise für Rahmen) und vieles mehr.

Hier sind dem Spieltrieb und der Fantasie keine Grenzen gesetzt.



Auswahlen lassen sich nachträglich verzerren oder ausblenden. 🔍

Auswahlen weiterverarbeiten

Was geschieht nun mit den Auswahlen? Ausgewählte Bereiche können kopiert, eingefügt, verschoben, mit Filtern bearbeitet werden usw.

Die Auswahlen selbst lassen sich:

- als Kanal speichern, mit weichen Kanten
- als Pfad speichern, zum Nachziehen, mit harten Kanten
- direkt nachziehen (mit Malwerkzeug) über „Bearbeiten → Auswahl nachziehen“
- mit Farbe, einem Farbverlauf oder Muster füllen

Auch wirken Translations- und Transformationswerkzeuge auf Auswahlen, sofern sie vorhanden sind – ansonsten bearbeitet *GIMP* das gesamte Bild.

Ebenen

Ebenen sind Verwaltungsstrukturen, um bestimmte Bildbereiche separat verwenden zu können. Sie entstehen beispielsweise beim Einfügen kopierter Teile, beim Duplizieren, durch eingefügten Text usw. Oft lassen sie sich als separate Folien vorstellen, was aber, wie jeder Vergleich, etwas hinkt. Sie dienen auch als Backup und zu Vergleichszwecken, beispielsweise um verschiedene Filter(einstellungen) zu testen.

Der Ebenendialog im rechten Fenster zeigt und verwaltet die Ebenen. Dort kennzeichnet ein Auge die momentan sichtbaren Ebenen. Ein Farbbalken markiert die aktuelle Ebene. Wichtig dabei: Nur auf dieser arbeitet *GIMP* normalerweise.

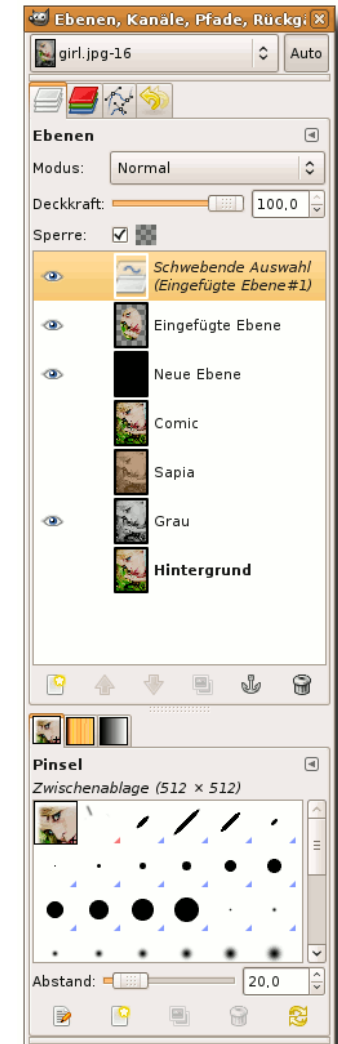
GIMP zeigt alle als sichtbar markierten Ebenen an. Nichttransparente Ebenen verdecken dabei die darunter liegenden, sofern sie (sichtbar) aktiviert sind. Es ist ein häufiger Fehler von *GIMP*-Anfängern, eine andere als die gewünschte Ebene

zu aktivieren und sich dann zu wundern, warum sich vermeintlich nichts tut. Tatsächlich arbeitet *GIMP* auf der aktiven, verdeckten Ebene.

Achtung: Nur wenige Dateiformate sind in der Lage, Ebenen zu speichern. In Animationen nehmen Ebenen die Einzelbilder auf.

Im Ebenendialog stehen über das Kontextmenü viele spezielle Funktionen für die gezielte Manipulation von Ebenen zur Verfügung. „*GIMP* – kurz & gut“ beschreibt sie im Detail.

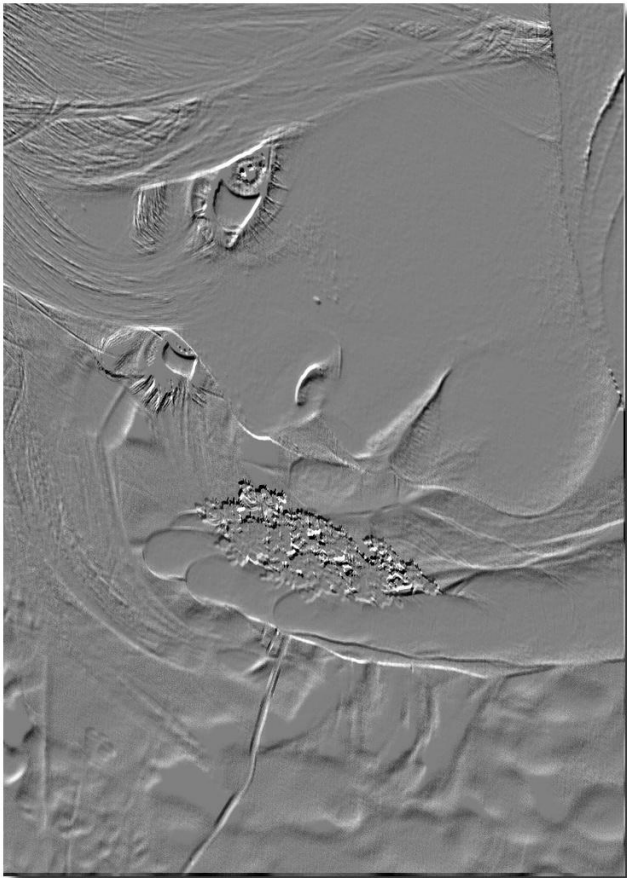
Ein Beispiel für das Erzeugen und Verwenden einer Ebene: Eine Auswahl markiert einen Bildbereich, **Strg** + **C** kopiert ihn in die Zwischenablage (im Pinseldock erscheint er als erster Pinsel). Mit **Strg** + **V** kopiert



Der Ebenendialog. 🔍



GIMP den Inhalt der Zwischenablage an die gleiche Stelle im Bild, von der er stammt. Allerdings befindet sich die Kopie jetzt auf einer temporären Ebene, der so genannten „schwebenden Auswahl“ – so heißt sie auch im Ebenendialog.



Über „Faser extrahieren“ und „Farbe → Entsättigen“ kann man schöne Relief-Bilder erzeugen. 🔍

Ein Klick auf den ersten Button unter der Ebenenliste (oder **Strg** + **Umschalt** + **N**) wandelt die temporäre Ebene in eine normale um.

Transparenz, Alphakanal und Deckkraft

Ebenen sind entweder undurchsichtig („opak“) oder (teil-)transparent. Der Wert für die Transparenz steuert die Deckkraft der Pixel im Alphakanal. Hierbei sind Werte im Bereich von 0 bis 255 möglich. Im Ebenendialog lässt sich für jede Ebene die Deckkraft variieren.

Bei opaken Ebenen (also ohne Transparenz) erscheint durch Anwendung des Radierers die (aktuelle) Hintergrundfarbe, bei Ebenen mit Alphakanal entsteht Transparenz.

Ebenenmodi

Die Pixel(werte) zweier übereinanderliegender sichtbarer Ebenen lassen sich auf unterschiedliche Weise kombinieren („verrechnen“), was zu besonderen Effekten führt. Im Ebenendialog steuert dies der Button „Modus“. Besonders wichtige Modi:

- „Normal“: Die Pixel der oberen Ebene ersetzen (verdecken) die der darunterliegenden Ebenen.
- „Vernichten“ erzeugt „raue“ Effekte. Dieser Modus wird gern mit einfarbigen Ebenen geringer Deckkraft verwendet.
- „Multiplikation“ verstärkt Farbwirkungen und Kontraste. Dieser Modus lässt sich oft bei Ebenen mit (teilweise) identischem Inhalt nutzen, wie sie beispielsweise durch das Kopieren und Einfügen erzeugt werden.

- „Überlagern“ wirkt entgegengesetzt und erzeugt schnell sehr helle („überbelichtet“) wirkende Bilder. Das kann beispielsweise beim Aufhellen von Schatten nützlich sein.
- „Faser extrahieren“ erzeugt bei identischen Ebenen reliefartige Strukturen, wenn beide um wenige Pixel verschoben werden.
- „Unterschied“ und „Abziehen“ werden für „Motion-Detection“ verwendet. Die Modi lassen alle identischen Teile der Ebenen verschwinden und zeigen dadurch die Unterschiede zwischen den Ebenen.

Farben

Das „Farben“-Menü enthält die wesentlichen Funktionen zum Farbgleich. Unter „Automatisch“ stehen sechs Funktionen zur Verfügung, die oft schon deutliche Verbesserungen erlauben.

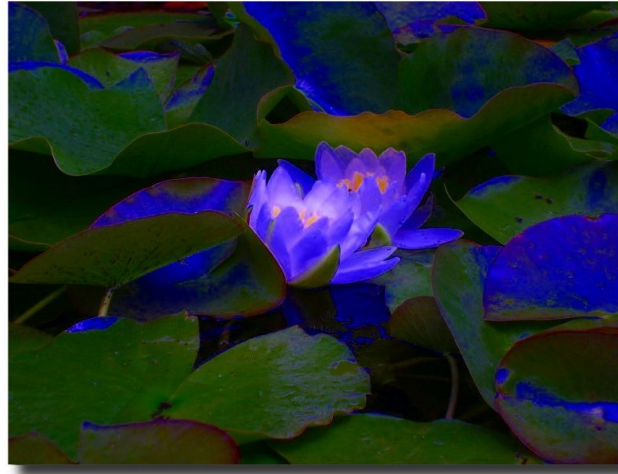
Mit „Abgleichen“ werden die Helligkeitwerte der aktuellen Ebene gleichmäßig über das Histogramm gestreckt. Das Ergebnis ist eine Kontrasterhöhung der angewählten Ebene, das entweder sehr gut ausfällt oder sehr schlecht (wie im Beispiel auf der nächsten Seite).

Der „Weißabgleich“ stellt die Farben neu ein und dient zur Korrektur von farbstichigen Bildern. Er kann aber nur bei RGB-Bildern verwendet werden.

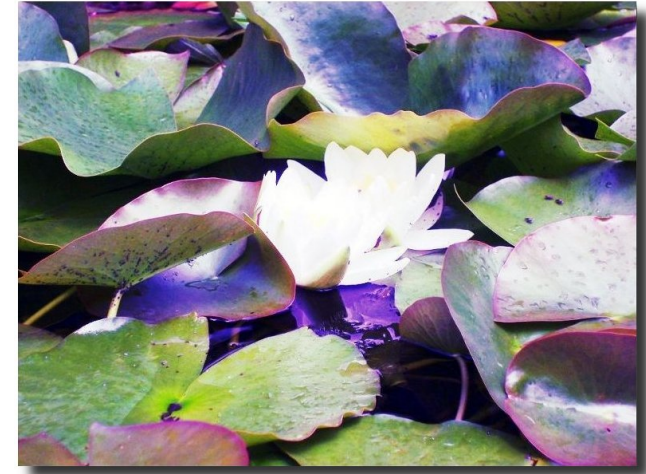
Die „Farbverbesserung“ erhöht die Farbsättigung der aktuellen Ebene und arbeitet dabei ähnlich wie die „Kontrastspreizung“.



Das Originalbild. 🔍



Farbverbesserung. 🔍



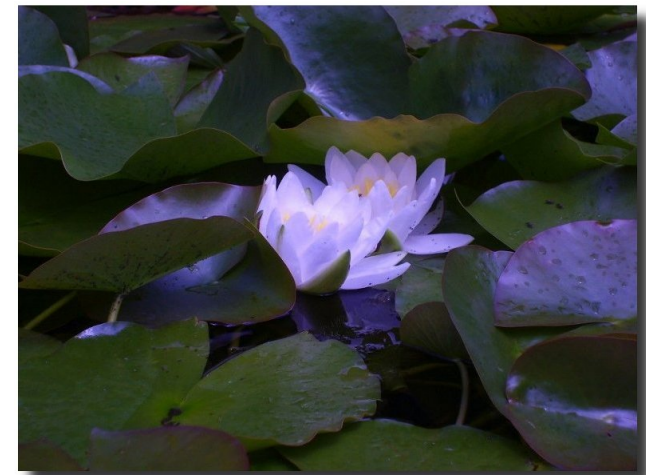
Abgleichen. 🔍



Weißabgleich. 🔍



Kontrastspreizung. 🔍



Normalisieren. 🔍

Der Menüpunkt „*HSV strecken*“ arbeitet wie der Name schon sagt im HSV-Farbraum. Seine Funktion ist ähnlich wie die der „*Farbverbesserung*“ und kann, wie auch „*Farbverbesserung*“, nicht auf Graustufenbilder angewandt werden.

Mit „*Kontrastspreizung*“ werden die Kanäle Rot, Grün und Blau gleichmäßig im Histogramm gestreckt. Dadurch werden helle Farben heller und dunkle Farben dunkler, es entsteht eine Kontrasterhöhung, mit leichten Farbverschiebungen.

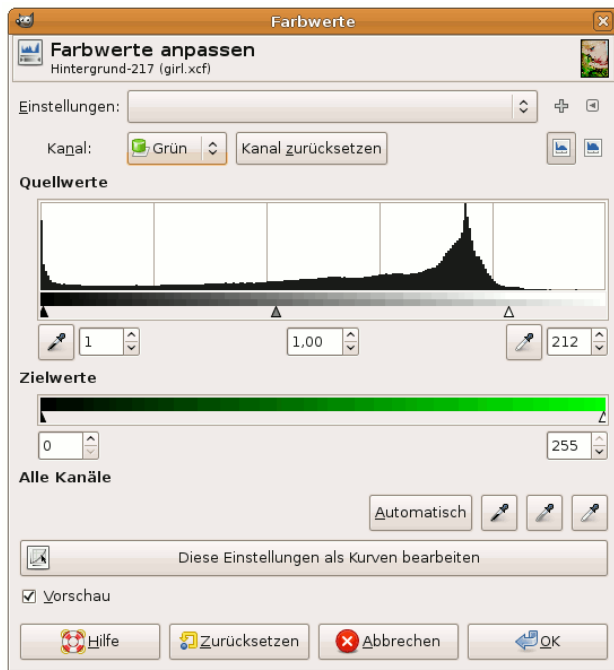
Beim „*Normalisierung*“ wird die Helligkeit der aktiven Ebene so verteilt dass der hellste Punkt fast weiß ist und der dunkelste annähernd schwarz ist. Das Kommando wirkt ähnlich wie die „*Kontrastspreizung*“ allerdings nur auf den Helligkeit-



kanal. Diese Filter verbessern flau und kontrastarme Bilder oft deutlich.

Für alle weitergehenden Aufgaben sind manuelle Lösungen erforderlich. Mindestens die Funktionen „Werte“ und (Gradations) „Kurven“ sollte man kennen.

Vorgehensweise: Zunächst die Farbkanäle einzeln so einstellen, dass die Wertebereiche vollständig ausgenutzt werden, dann eventuell noch die Helligkeiten („Werte“) korrigieren. Die aktuellen Farbwerte im Histogramm zeigt die Farbpipette im Bildfenster.



Im Dialog „Werte“ kann man den Wertebereich für jede Farbe einstellen. 🔍

Filter

Viele weitere sinnvolle und wünschenswerte Funktionen sind bei GIMP in Form von „Filtern“ (oft als Plug-ins realisiert) implementiert. Einige arbeiten weitgehend automatisch, wie bei den Farbkorrekturen gezeigt, aber die weitaus meisten lassen sich in unterschiedlicher Weise manuell steuern. Hier hilft nur Experimentierfreude und Neugier weiter, um die besten Ergebnisse zu erzielen.

Typische Aufgaben für Filter ist das Weichzeichnen, Verfremden, Dekorieren oder Schärfen von Bildern.

Schärfen

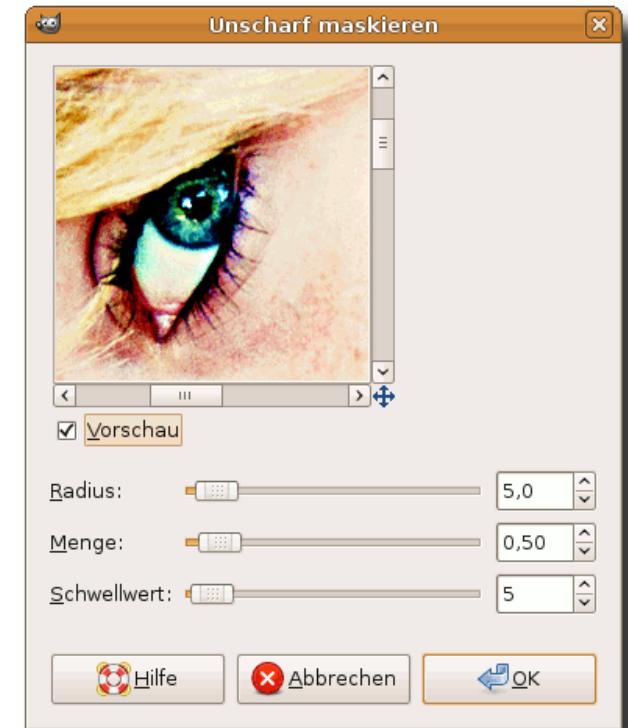
Im „Filter“-Menü finden sich unter „Verbessern“ mehrere Schärfungfilter. Alle Schärfungsmethoden arbeiten ähnlich: Sie erhöhen den Kontrast an Farbgrenzen, was bewirkt, dass das Auge dies als „scharfes“ Bild interpretiert.

Unschärf maskieren

Einer der wichtigsten Schärfungfilter heißt „Unschärf maskieren“, was auf eine im Fotolabor angewendete Methode zurückgeht. Dort wurde das unscharfe Negativ mit einem zweiten, noch etwas unschärferen überlagert und dann das Positiv belichtet. Die beiden unscharfen Negative führten dazu, dass sich besonders an den Kanten die Bilder überdeckten, was auf dem Positiv zu einer Verstärkung führte.

GIMPs Filter funktionieren analog, haben aber zusätzliche Parameter: So stellt „Schwelle“ ein, ab welchem Wert der Filter überhaupt wirken soll,

was die Qualität deutlich verbessert. Auch die Parameter „Radius“ und „Menge“ steuern einzelne Aspekte des Algorithmus. Als Regel geben erfahrene Anwender oft „geringer Radius, große Menge“ an. Aber wie immer sollte man hier experimentieren.



Über „Unschärf maskieren“ kann man ein Bild schärfen. 🔍

Das folgende Beispiel zeigt im Original einen leider unscharf aufgenommenen Baum.

Im ersten Schritt wurde das Bild mit dem Filter „Unschärf maskieren“ bearbeitet und so



Das Originalbild ohne Schärfung. 🔍

schon eine erhebliche verbesserung des Schärfeeindrucks erzielt.

Im zweiten Schritt wird ein zweites „unscharfmaskiertes“ Bild mit anderen Schärfungsparametern erzeugt. Danach wird das ganze Bild ausgewählt (**Strg** + **A**) und in die Zwischenablage kopieren (**Strg** + **C**). In dem zuerst erstellten Bild wird der Inhalt der Zwischenablage nun über „*Bearbeiten* → *Einfügen als* → *Neue Ebene*“ eingefügt. In der



Nach dem „Unschärf maskieren“. 🔍

Ebenenübersicht ist jetzt eine weitere Ebene mit dem Namen „Zwischenablage“ entstanden. Zum Abschluss ändert man den „Modus“ im Ebenendialog von „Normal“ auf „Multiplikation“, was die beiden Ebenen im Multiplikationsmodus kombiniert.

Dies führt neben einer allgemeinen Intensivierung von Farben und Kontrasten auch zu einer signifikant verbesserten Schärfe (auch des Blattwerks im Hintergrund). Die dadurch



Die Multiplikation der beiden unscharf maskierten Bilder. 🔍

herabgesetzte Helligkeit könnte nun auch noch angepasst werden. Möchte man abschließend wieder nur eine Ebene haben, kann man die beiden über „*Bild* → *Bild zusammenfügen*“ zu einer Hintergrundebene vereinen.

Schlusswort

GIMP ist ein mächtiges Werkzeug, was vor allem durch Ausprobieren kennen gelernt werden kann.



Auch die Onlinehilfe, zu erreichen über „Hilfe → Hilfe“ oder **F1**, ist lesenswert und erklärt die vielen Funktionen anschaulich. Viele Tastenkombinationen beschleunigen dabei den Zugriff auf genutzte Funktionen. Eine kurze Zusammenfassung der Tastenkombinationen und Menüs für *GIMP* gibt es bei O'Reilly zum Download [6].

Verwendete Printmedien:

- Günther, Karsten: „GIMP – kurz & gut“, O'Reilly, 2009

LINKS

- [1] <http://www.gimp.org/>
- [2] <http://www.freiesmagazin.de/freiesMagazin-2009-11>
- [3] <http://www.flickr.com/photos/pinksherbet/1114534807/in/set-72157610551917961/>
- [4] <http://www.flickr.com/photos/pinksherbet/>
- [5] <http://creativecommons.org/licenses/by/2.0/deed.de>
- [6] <http://www.oreilly.de/catalog/gimp/ggger/chapter/gimp2.6-sheet-landscape-A4.pdf>

Autoreninformation

Karsten Günther arbeitet seit nun mehr als 15 Jahren mit Linux, \LaTeX und andere freier Software, wie auch GIMP. Seit mehreren Jahren ist das sein bevorzugtes Werkzeug zur Bildbearbeitung, für das er auch seit einiger Zeit Schulungen abhält.

Diesen Artikel kommentieren



„Understocked“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/617>



X2goserver-home unter Ubuntu installieren von Heinz-M. Graesing

Die Software **X2go** [1] ist eine Remote-Desktop-Anwendung, welche die NX-Bibliotheken [2] nutzt, um die Bildschirmhalte zu übertragen. Der Vorteil gegenüber bildschirmaufzeichnenden Protokollen ist, dass die lokalen Fähigkeiten eines X-Servers mit genutzt werden können und die Darstellung oft deutlich schneller erfolgt. Gegenüber freeNX [3] verwendet X2go PulseAudio zur Soundweiterleitung, SSHFS zur Dateifreigabe und das sudo-System, um Berechtigungen zu setzen. (Auch für FreeNX gibt es Erweiterungen, die SSHFS und PulseAudio nutzen.)

Redaktioneller Hinweis: Der Artikel „X2goserver-home unter Ubuntu installieren“ erschien erstmals bei Pro-Linux [4] und wird mit freundlicher Genehmigung des Autors unter der GNU Free Documentation License [5] veröffentlicht.

Mit der Version 3.01 existiert nun auch eine native Version von X2goserver für Ubuntu, die sich sehr einfach installieren lässt. Gegenüber der Vollinstallation muss man zwar auf ein paar Features verzichten, aber um das Projekt kennen zu lernen, ist die „Home“-Variante ideal. Da jeglicher Datenverkehr über SSH getunnelt wird, ist eine X2go-Sitzung sehr sicher.

Installation

Da sich X2go momentan nicht in den Ubuntu-Paketquellen befindet, muss man eine neue

Quelle hinzufügen. Das Repository wird auf Server- und auf Clientseite benötigt. Es lassen sich natürlich auch alle Pakete auf einem System installieren, wenn ein Rechner Client und Server sein soll. Zuvor sollte allerdings der Schlüssel des Projekts installiert werden. Dazu wird eine Konsole benötigt. Folgende Befehle importieren den Schlüssel:

```
# gpg --keyserver wwwkeys.eu.pgp.~
net --recv-keys C509840B96F89133
# gpg -a --export C509840B96F89133 ~
| apt-key add -
```

Das Hinzufügen einer Paketquelle geht am einfachsten über das Ubuntu-Menü „System → Systemverwaltung → Software-Paketquellen“. Im Reiter „Andere Software“ lassen sich fremde Quellen hinzufügen. Hier fügt man folgende Zeile hinzu:

```
deb http://x2go.obviously-nice.~
de/deb/ lenny main
```

Nach der Aktualisierung stehen die X2go-Pakete der Paketverwaltung zur Verfügung.

Der Quellcode von X2go kann direkt über den das Repository [6] heruntergeladen werden.

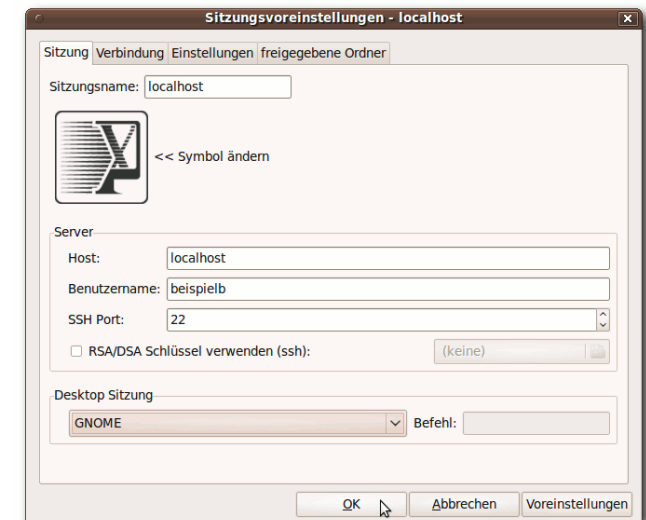
Installation auf dem Server

Auf dem Server muss nun das Paket **x2goserver-home** installiert werden. Man sollte unbedingt darauf achten, dass die Home-

Variante installiert wird. Die Vollversion würde nicht ohne Konfiguration laufen und eine PostgreSQL-Datenbank mit auf das System ziehen.

Installation der GNOME-Erweiterungen

Damit man Client-Dateisysteme mit dem Server verbinden und Sitzungen mit Hilfe eines Menüeintrags anhalten kann, werden die **x2go-gnomebindings** benötigt. Auch diese findet man in der Paketverwaltung. Nach der Installation sollte das Ubuntu-Menü „System“ den Punkt „X2go Sitzung unterbrechen“ enthalten. Mit diesem Menüpunkt kann man später eine laufende X2go-Sitzung anhalten und z. B. auf einem anderen Rechner weiterführen.



Sitzungsvoreinstellungen.



Installation auf dem Client

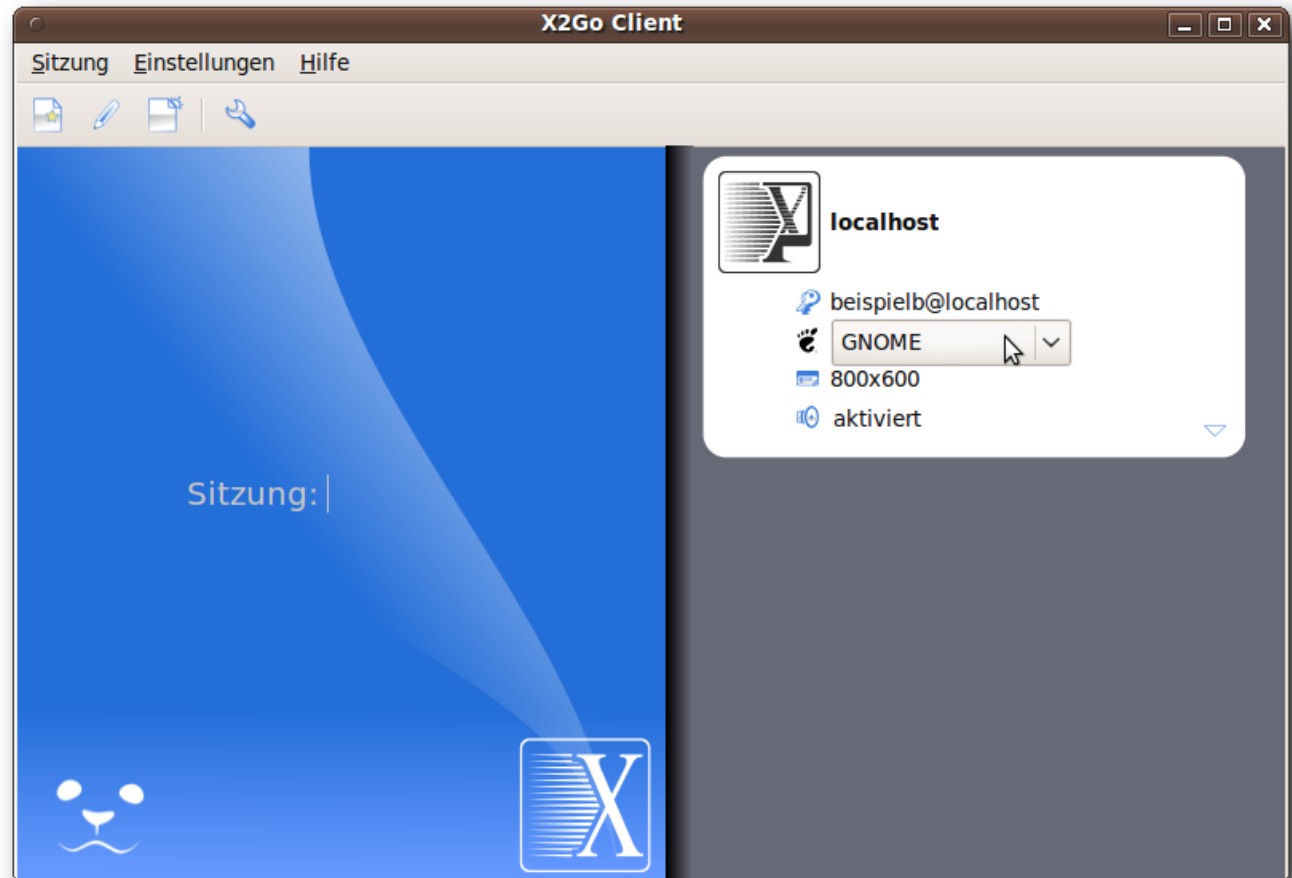
Für GNOME-Benutzer bietet das *X2go*-Projekt einen GTK-Client an. Neben diesem existieren noch Clients für Qt, die Konsole, Maemo, Windows und Mac OS X. So kann man auch bequem von einem Windows-PC auf seinen Linux-rechner (zum Beispiel über das Internet) zugreifen.

Der Client lässt sich – sofern man das Software-Repository installiert hat – nun ebenfalls mit dem Paketmanager finden. Das Paket nennt sich **x2goclient-gtk** und ist nach der Installation im GNOME-Menü unter „Anwendungen → Internet → *X2Go Client(gtk)*“ zu finden.

Konfiguration des Clients

Nach dem Start von *X2goclient* muss zunächst eine neue Sitzung angelegt werden. Dafür kann man das erste Symbol in der Werkzeugleiste nutzen.

Der neu gestartete Dialog fragt nun die wichtigsten benötigten Informationen ab. Der „*Sitzungsname*“ beschreibt die aktuelle Verbindung und ist frei wählbar. „*Host*“ beschreibt den Hostnamen oder die IP-Adresse des Servers. Dieser muss per SSH erreichbar sein. Kommt keine SSH-Verbindung zustande, wird auch *X2go* nicht funktionieren. Der „*Benutzername*“ entspricht dem Benutzer auf dem Serversystem. Ein Passwort kann nicht hinterlegt werden, aber es kann ein RSA/DSA-Schlüssel generiert und der öffentliche Teil zur Authentifizierung verwendet werden. Den „*SSH Port*“ sollte man nicht ändern – er ist

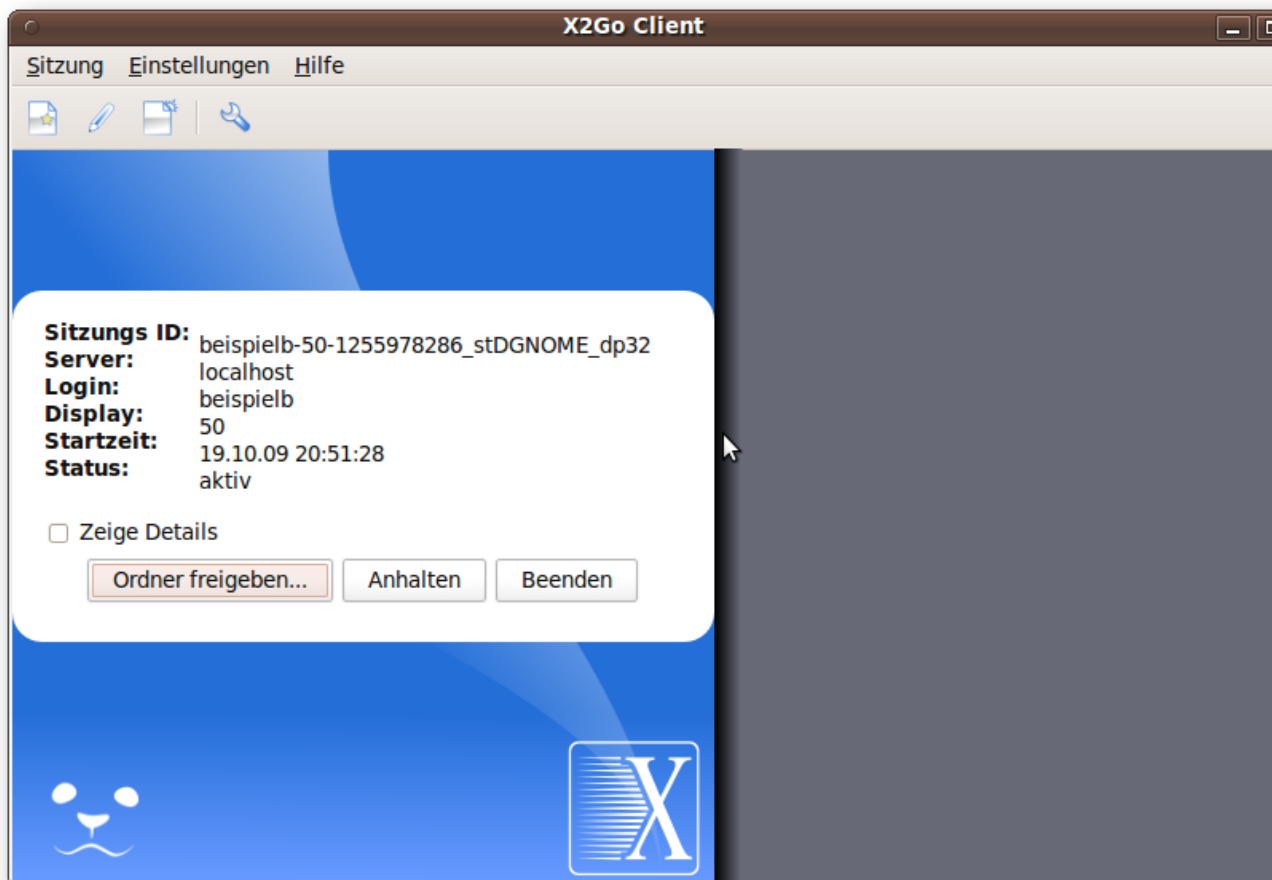


Start einer Sitzung. 🔍

voreingestellt identisch mit dem benötigten SSH-Server. Unter „*Desktop-Sitzung*“ lässt sich nun „*GNOME*“ auswählen. Man sollte natürlich darauf achten, dass die ausgewählte Desktopumgebung auch wirklich installiert ist.

Die Sitzung wird mit einem Klick auf „*OK*“ gespeichert und ist jetzt als Karte auf der

rechten Seite sichtbar. Hat man mehr als eine Sitzung angelegt, stapeln sich diese im rechten Bildschirmbereich. Ein Klick auf die Karte befördert diese in den linken Bereich und erwartet ein Passwort. Benötigt wird das Passwort des Server-Benutzers. Wurde dieses eingegeben, erfolgt eventuell ein Hinweis, dass der Host-Key nicht verifiziert werden konnte. Die Meldung



Client-Fenster bei laufender Sitzung. 🔍

erscheint aber nur, wenn zuvor noch kein Kontakt mit dem Server über SSH hergestellt wurde. In dem Fall akzeptiert man den Schlüssel und wartet, bis die Verbindung startet.

Die Sitzung wird in einem neuen Fenster gestartet, dessen Größe man vor dem Start in

den Einstellungen definieren oder während der Sitzung durch Ziehen des Rahmens anpassen kann. Im noch bestehenden Client-Fenster kann man auch während der Sitzung bequem Dateisysteme freigeben, indem man die Schaltfläche „Ordner freigeben...“ nutzt. Sitzungen können nun angehalten und auch auf einem an-

deren Rechner fortgeführt werden. Alle Benutzer der Gruppe **x2gousers** können sich nun über X2go anmelden. Falls das bei einem Nutzer nicht klappt, sollte die Gruppenmitgliedschaft überprüft werden.

Wer die Audio-Weiterleitung nutzen will, sollte sicherstellen, dass die Programme, die den Sound erzeugen, das angegebene Soundsystem benutzen. Voreingestellt ist PulseAudio, welches auch die wenigsten Probleme bereitet.

LINKS

- [1] <http://www.x2go.org/>
- [2] <http://www.nomachine.com/>
- [3] <http://freenx.berlios.de/>
- [4] <http://www.pro-linux.de/berichte/x2go-ubuntu.html>
- [5] <http://www.gnu.org/copyleft/fdl.html>
- [6] <http://x2go.obviously-nice.de/deb/>

Autoreninformation

Heinz-M. Graesing ist neben Oleksandr Shneyder der Hauptentwickler von X2go. Die Idee zur Software entstammt seiner Tätigkeit als Systemadministrator der Stadt Treuchtlingen.


[Diesen Artikel kommentieren](#)

ASCIIPortal – The GUI is a lie! von Dominik Wagenführ

Das Spiel Portal [1] hatte aufgrund seiner genialen Spielidee einen riesigen Erfolg im Jahr 2007. Die Relevanz des Spieles hat sogar die Wikipedia erkannt und eines der besten Computerspiel-Zitate („*The cake is a lie!*“) stammt daraus. Mit *ASCIIPortal* [2] von Joe Larson wird das Spielprinzip für Konsolenjunkies auch auf die Linux-Plattform portiert.

Zuerst ein wichtiger Hinweis: *ASCIIPortal* glänzt nicht mit überragender 3-D-Grafik. Dafür darf man umso erstaunter sein, welche Möglichkeiten ein reiner Zeichensatz einem in der Konsole bietet.



Der Titelschirm von *ASCIIPortal*. 

Spielprinzip

Das Spielprinzip von *ASCIIPortal* ist sehr schnell erklärt: Mit Hilfe einer Kanone kann man zwei verschiedenfarbige Portale (ein gelbes und ein blaues) an den Decken, Böden und Wänden eines zweidimensionalen Levels erzeugen, durch die man sich dann bewegen kann. Wichtig dabei ist der Impulserhaltungssatz, der da in Kurzform lautet: „*Was schnell rein geht, kommt schnell raus.*“ Lässt man sich also von einer Plattform in das eine Portal fallen, kommt man auf der Gegenseite mit dieser Geschwindigkeit auch wieder heraus und wird entsprechend in die jeweilige Richtung (also auch nach oben) geschleudert.

Installation

Die Installation ist denkbar einfach. Auf der Webseite [3] gibt es zwei Archive, je eines für 32-Bit- und 64-Bit Rechner. Der Quellcode des Spieles liegt den Archiven bei, also ist auch die Kompilierung auf anderen Systemen kein Problem, solange

folgende Entwicklerpakete installiert sind (die Namen stammen von einem Ubuntu-System):

- **build-essentials**
- **libSDL1.2-dev**
- **libSDL-mixer1.2-dev**

Nach dem Entpacken des Archivs kann man *ASCIIPortal* mittels

```
$ cd src
$ make
$ cp asciportal ..
$ cd ..
```

selber kompilieren. Wer das nicht will, findet im Hauptverzeichnis die Datei **asciportal**, die man einfach nur ausführen muss. Dafür reicht ein Doppelklick im Browser oder die Eingabe auf der Konsole:

```
$ ./asciportal
```

Zur reinen Ausführung des vor- oder eigenkompilierten Codes benötigt man folgende Pakete:

- **libSDL1.2** (bzw. **libSDL1.2-debian** in Debian/Ubuntu)
- **libSDL-mixer1.2**





Spielstart





Direkt nach dem Start drückt man irgendeine Taste („*Where is the any key?*“) und kann mit den Pfeiltasten den gewünschten Menüpunkt




Das grüne A muss zum weißen E. 🔍

auswählen. „1) Begin“ sollte es für den Anfang tun. Eine nette Computerstimme begrüßt einen als ungebetenen Besucher – Kuchen kann man also sicher keinen erwarten.

Mit den Pfeiltasten  und  bewegt man sich nach rechts und links. Die Portalkanone justiert man durch die Pfeiltasten  und . Etwas einfacher ist es, den Ziffernblock für die jeweilige Richtung der Kanone zu nutzen. Auf Leitern gibt es beispielsweise gar keine andere Möglichkeit.

Mit  oder  auf dem Nummernblock schießt man abwechselnd Portale. Da man da ganz schön durcheinanderkommen kann, welches Portal denn das nächste sein wird, ist es besser, mit den Tasten  (gelbes Portal) und  (blaues Portal) zu arbeiten.

Wie *ASCIIpOrtal* ansonsten funktioniert, muss jeder selbst herausfinden, denn auch das ist Teil des Spiels. :)

Ein Hinweis: Mit  schaltet man die Sichtweise der Portale um. Entweder ist der neue Sichtbereich mit (gelben und blauen) Linien markiert, gar nicht markiert oder komplett farbig ausgefüllt. Mit der ersten Auswahl (Standard) kommt man wohl am besten zurecht.

Weitere Levels

Sollte man die 50 Standardlevels durchgespielt haben, kann man auch noch die Zusatzlevel spielen. Dafür wählt man im Hauptmenü den Punkt „3) Change Map Set“ und gibt den Namen des Ordners mit den Dateien an. In der Version 1.1 vom 31. Oktober 2009 sind dies die Ordner **evil**, **extralevels** und **raisonbran648**. Die Standardlevels befinden sich im Ordner **maps**.

Leveleditor

Der Leveleditor besteht aus einem Texteditor der eigenen Wahl. Im Hauptordner des Spiels findet

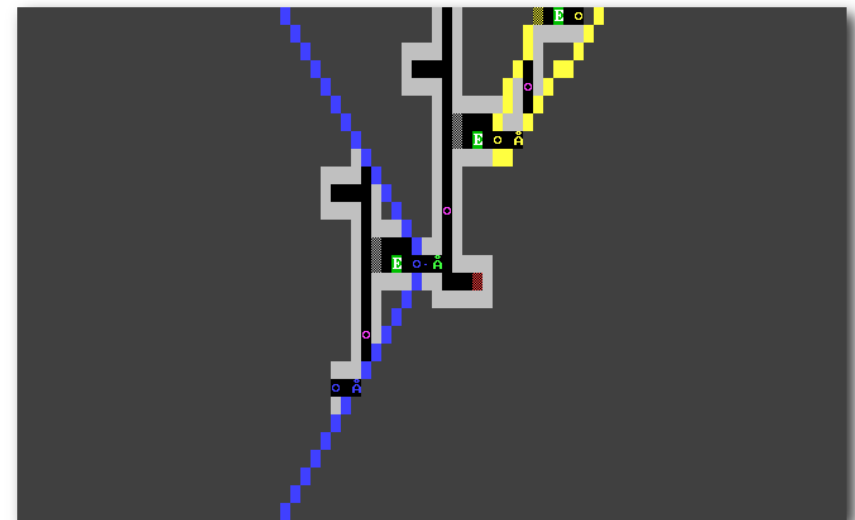
man die Datei **map_making_tips.txt**, die einige hilfreiche Hinweise gibt, wie man Levels erstellen kann.

Wie es zu erwarten war, besteht jedes der Spielelemente aus einem ASCII-Zeichen [4]. Auf diese Art kann man Spieler, Wände, Ausgänge, Türen etc. definieren und so das Level gestalten.

Zusätzlich kann man bis zu 10 Textnachrichten an bestimmten Stellen im Level einblenden lassen, die den Spieler unterhalten oder womöglich auch einmal helfen sollen.

Die Levels können dann in einem extra Verzeichnis gespeichert werden und sollten aufsteigend nummeriert sein: **001.txt**, **002.txt** usw.

Es gibt auch von einem Community-Mitglied einen Level-Editor, der aber nur für Windows pro-



Der Blick durch die Portale kann verwirrend sein. 🔍

grammiert wurde und auch nicht im Quellcode vorliegt.

Joe Larson plant aber vielleicht, einen Online-Editor zu programmieren, um die Level-Gestaltung zu vereinfachen.

Kommandozeilen-Parameter

Die Hilfe mit allen Kommandozeilen-Parametern erhält man mittels

```
music:ascii_threat2.mp3
name:Portile22
message:Do you have a plan if you miss the first time?
NNNNNNNNNNNN
NNNN ···· <N
NNN#=#NNNNN
NNN# #
NNN# #
NNN# #
#### #
# @ #
#### #
NNN# #####
NNN# $ · #
NNN# $ Q #
NNN# #####
NNN# #
NNN# #
NNN# #
NNN# #
N### #
N# · #####
N### · " #
NNN#####
```

Obiges Level im Texteditor. 

```
$ ./asciportal -h
```

Wichtig ist eigentlich nur die Option **-r 1024x768**, um das Spiel in einer höheren Auflösung spielen zu können. In Kombination mit dem Vollbildmodus (mittels **-f**) ist das ganz hilfreich (aber siehe unten).

Mit **-m Verzeichnis** lädt man ein bestimmtes Levelset beim Start des Spiels.

Fazit

ASCIIPortal fängt den Stil des großen Bruders sehr gut ein und portiert das Spielprinzip in faszinierender Weise auf die Konsole. Es macht wirklich Spaß, auch wenn – für manche vielleicht gerade weil – die Grafik minimal ist.

Die Bemerkungen des Computers sind recht witzig und die Musik von Steve Fenton erinnert an die guten alten Spielhallenzeiten.

Es gibt aber auch noch kleinere Bugs. So funktioniert die Option **-d** nicht und führt zu einem „Segmentation Fault“. Und bei

der Benutzung des Vollbildmodus schaltet das Spiel nach dem Beenden nicht zurück in die alte Desktopauflösung. Aber die Fehler werden sicherlich in einer späteren Version ausgemerzt (oder liegen vielleicht auch am Testrechner) und stören sowieso nicht weiter.

Bei Fragen kann man sich an die Community im Forum [5] wenden.

ASCIIPortal ist ein rundherum gelungenes und kurzweiliges Spiel für alle, die nicht immer Wert legen auf Displacement Mapping, High Definition Rendering oder Ambient Occlusion.

LINKS

- [1] [http://de.wikipedia.org/wiki/Portal_\(Valve\)](http://de.wikipedia.org/wiki/Portal_(Valve))
- [2] <http://cymongames.com/asciportal/> 
- [3] <http://cymongames.com/asciportal/#download> 
- [4] <http://de.wikipedia.org/wiki/ASCII>
- [5] <http://cymongames.com/forum/index.php?board=14.0> 

Autoreninformation

Dominik Wagenführ spielt sehr gerne unter Linux. Vor allem Geschicklichkeits- und Denkspiele machen ihm dabei viel Spaß.

Diesen Artikel kommentieren 

Abschluss des zweiten Programmierwettbewerbs

Der am 27. September 2009 gestartete zweite Programmierwettbewerb [1] wurde am 13. Dezember wie geplant beendet [2]. Zahlreiche Vollzeit- und Hobby-Programmierer haben sich hingesetzt und eine KI (Künstliche Intelligenz) für den Roboter geschrieben, der sich durch eine Fabrikhalle kämpfen muss. Dieser Artikel soll die Lösungen näher beschreiben und die Gewinner der einzelnen Kategorien bekannt geben.

Zuerst möchten wir uns bei allen Teilnehmern bedanken, die den Wettbewerb mit ihren Lösungen überhaupt möglich gemacht haben, denn nicht immer ist eine Teilnahme der Community bei solchen Aktionen selbstverständlich. Zusätzlich bedanken wir uns auch für das Lob und die Kritik der Teilnehmer, welche wir erhalten haben.

Die Aufgabe

Noch einmal kurz die Spielidee: Die KI jedes Teilnehmers sollte auf einer bestimmten Schwierigkeitsstufe einen Roboter durch eine Fabrikhalle steuern, die mit sich bewegenden Elementen wie Förderbändern oder Drehrädern und auch Gefahren wie Löchern und Schrottpressen ausgestattet war [3]. Pro Runde erhält die KI acht Bewegungskarten und muss sich fünf aussuchen, um den Roboter damit über das Spielbrett zu bewegen.

Die Schwierigkeitsstufe „leicht“ ließ dabei aber die Gefahren weg, sodass ein Roboter nicht ster-

ben konnte, in „mittel“ waren die Gefahren da, aber man konnte nicht vom Spielfeldrand fallen und bei „schwer“ musste der Roboter noch anpassen, dass er nicht über den Rand rutscht.

Als Extra-Kategorie sollte eine 3-D-Oberfläche bzw. isometrische 2-D-Ansicht der Fabrikhalle und des Roboters programmiert werden.

Als Preis winkten den Gewinnern Buchgutscheine im Wert von zehn, zwanzig, dreißig und fünfzig Euro.

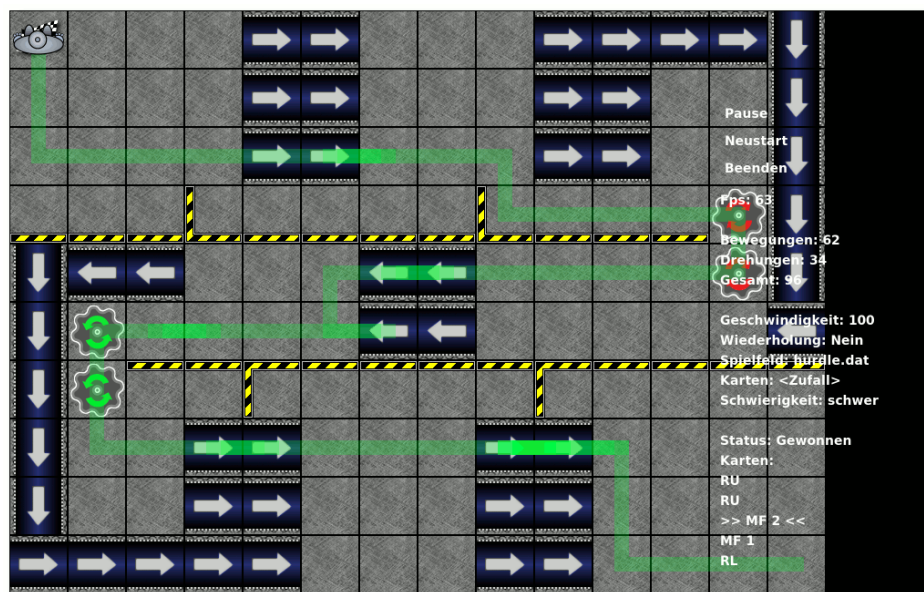
Der Lösungsansatz

Essentiell für das Spiel ist eine gute Wegfindung, sodass der Roboter es auch ins Ziel schafft. Die

einfachste Art der Entscheidung ist es, zu schauen, ob man sich dem Ziel durch die Bewegung nähert. Man rechnet also einfach den Abstand des Roboters zum Ziel aus (z. B. als Summe des Abstandes der horizontalen und vertikalen Felder) und vergleicht mit dem vorherigen Abstand. Dieses Vorgehen ist aber immer dann problematisch, wenn das Ziel von einem Hindernis wie einer Wand getrennt wird, denn die obige Methode unterscheidet nicht, ob der Roboter auf dem direkten Weg das Ziel überhaupt erreichen kann. Aus diesem Grund empfiehlt sich ein echter Wegfindungsalgorithmus.

Nach der Wegfindung haben sich die meisten Teilnehmer einfach alle 6720 Kombinationen (fünf aus acht Karten in geordneter Reihenfolge) angeschaut und überprüft, wo der Roboter landet und wie weit es von der neuen Position zum Ziel ist (siehe Wegfindung oben). Die Kombination mit der kürzesten Entfernung (bzw. geringstem Aufwand) wird dann gewählt.

Für „Faule“ wurde die Spielelogik (Einlesen der Dateien, Abarbeiten der



Bei „Hurdle“ muss man zwei Förderbänder auf einmal nehmen. 🔍

Sequenzen) als Bibliothek zur Verfügung gestellt, sodass man mehr Zeit in die Künstliche Intelligenz stecken konnte.

Die Spielbretter

Einige der Spielbretter sollen hier kurz vorgestellt werden, um auf einige Besonderheiten aufmerksam zu machen. Neben den regulären Spielbrettern gab es auch ein paar Testbretter, an denen die Intelligenz der KIs getestet werden konnte.

Hurdle

Der Hindernislauf war manchmal ein Problem, weil man zwei aufeinanderfolgende



Auf dem optimalen Weg ist man in „Oily“ schnell am Ziel. 🔍

Förderbänder überwinden musste. Daneben war es das einzige Spielbrett im Wettbewerb, welches eine labyrinthartige Struktur aufweisen konnte. Ansonsten war dies aber eines der einfachen Bretter.

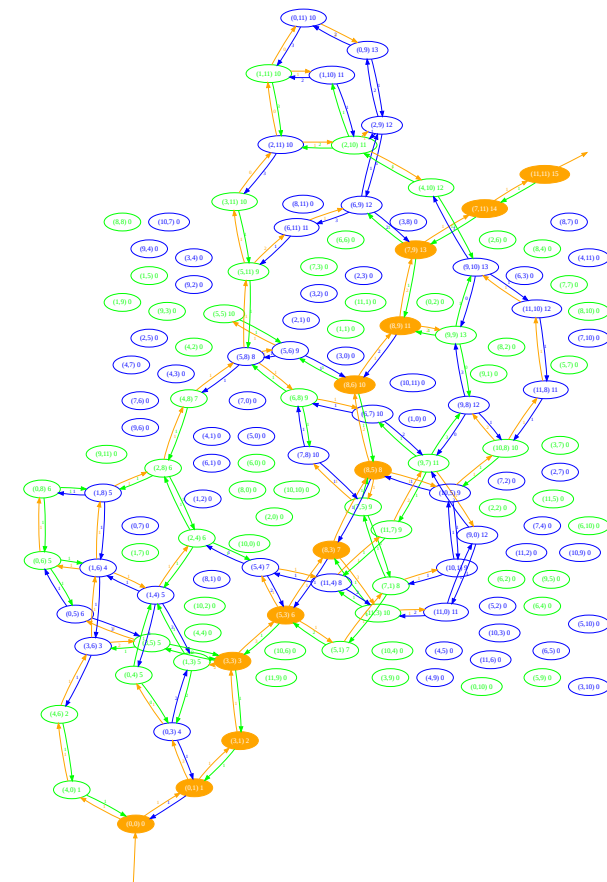
Zusätzlich war die rechteckige Anordnung bei manchen KIs in der jeweils ersten Version problematisch, was auch daran lag, dass auf der Wettbewerbsseite lange stand, dass die Spielbretter „voraussichtlich alle eine Größe von 12x12-Feldern [haben], wobei die KI sich natürlich darauf verlassen sollte, dass dem so ist.“ – Hier fehlte einfach ein kleines „nicht“. an der richtigen Stelle. Ein großes „Entschuldigung“ daher von unserer Seite, für diesen Fehler in der Ausschreibung.

Oily

Dies ist eine sehr ölige Angelegenheit. Auch wenn es fies erscheint, ist das Ziel mit sehr wenig Schritten zu erreichen, es gibt für die Lösung drei Wege.

Das Problem hier ist, dass man auf Ölfeldern nicht stehenbleiben bzw. abbiegen kann. Eine Wegfindung, die das nicht beachtet, schlägt dann einen Weg vor, dem man gar nicht folgen kann.

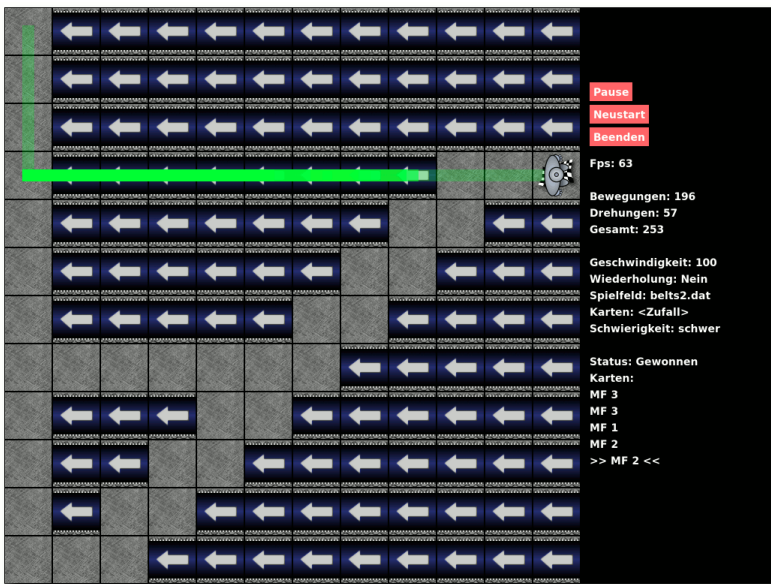
Ein guter Wegfindungsgraph beachtet dies, indem er solche Spielfelder löscht und die beiden Felder darunter und darüber bzw. rechts und links davon direkt verbindet.



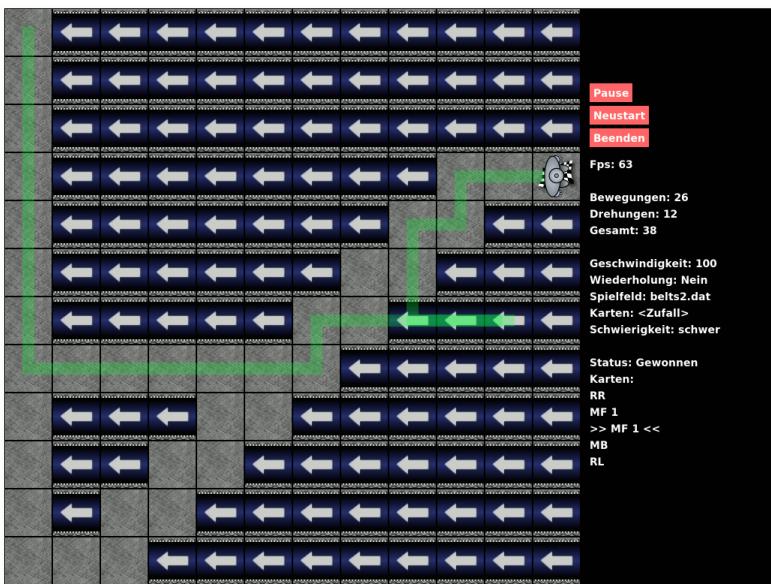
Der Graph mit kürzestem Weg in „Oily“. 🔍

Belts

Dieses Spielbrett zeigt, wie sinnvoll die Wegfindung der KI ist.



„Belts1“ – Mit dem Kopf durch die Wand. 🔍



Der einfache Weg. – 38 Bewegungen anstatt 253. 🔍

Zahlreiche KIs suchen sich zwar einen Weg zum Ziel, beachten dabei aber nicht die Art eines Spielfeldes. Das heißt, sie versuchen gegen die Laufrichtung eines Förderbandes anzurennen, obwohl es einen einfachen Weg um das Problem herum gibt.

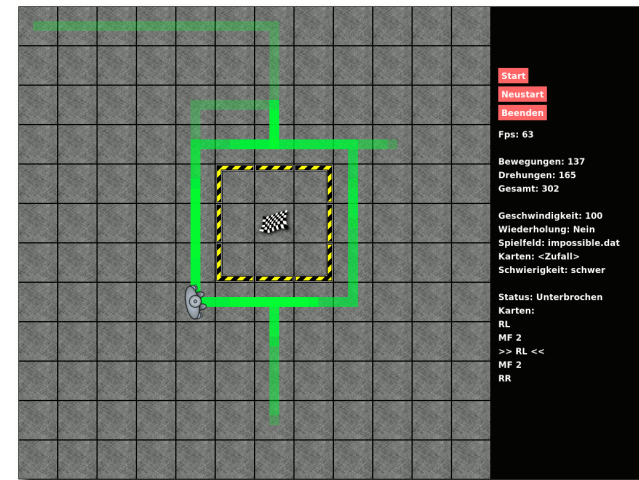
Wichtig war dies aber vor allem beim Testbrett „Gorge“, bei dem es rein optisch drei Wege zum Ziel gab, aber nur einer (der mittlere) wirklich sinnvoll beschritten werden konnte.

Das Spielbrett „Belts2“ testet das Verhalten der KI auf dem gleichen Spielfeld, wenn man sie in die richtige Richtung vor das Problem setzt. Aber auch hier sind viele stur und laufen lieber Marathon gegen die Laufrichtung des Förderbandes.

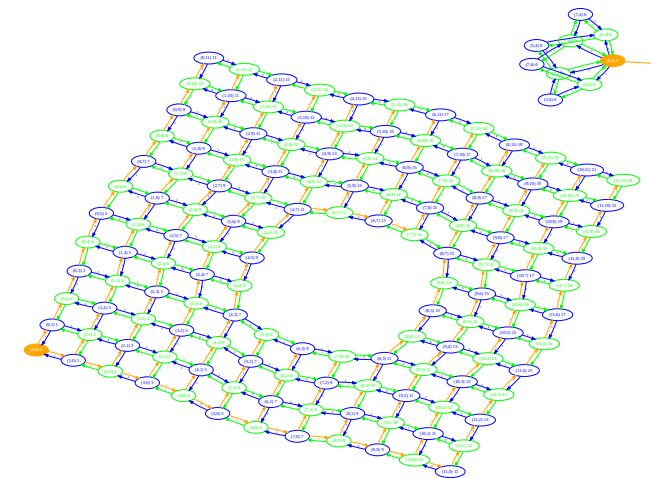
Impossible

Dieses Spielbrett war ein Test, was eine KI macht, wenn es gar keine Möglichkeit gibt, zum Ziel zu gelangen. Die meisten KIs haben das erkannt; wenn sie aber nur eine reine Abstandsrechnung vorgenommen haben, kreisten sie ewig um den Block herum, ohne zum Ziel zu gelangen.

Schaut man sich den Wegfindungsgraph an, sieht man sofort, dass es keine Lösung gibt, da Start und Ziel in zwei getrennten Graphteilen liegen.



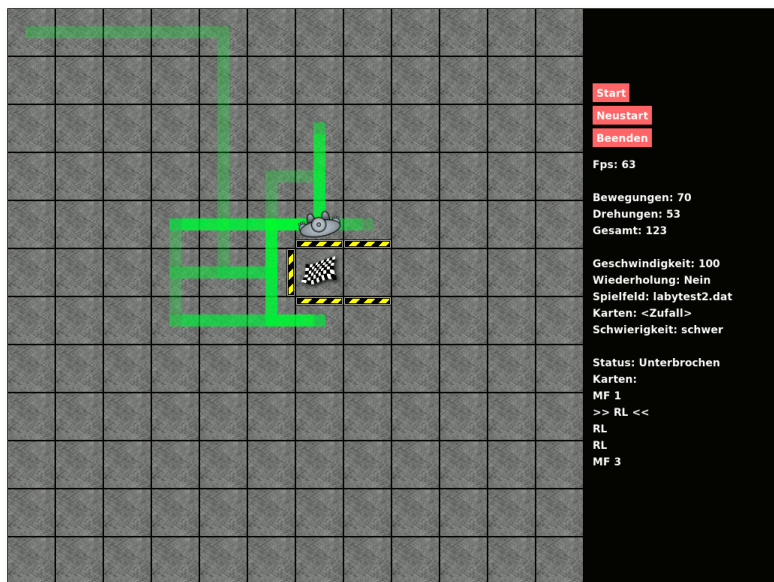
Wie der Geier um das Aas kreist der Roboter um das Ziel. 🔍



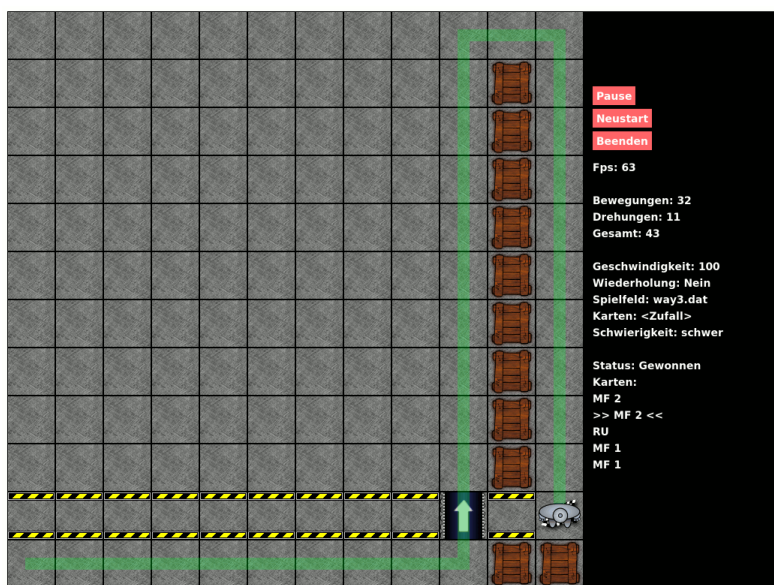
Der Graph zu „Impossible“. 🔍

Way1

Dieses einfache Beispiel zeigt, ob die KI einen echten Wegfindungsalgorithmus nutzt oder nur eine Abstandsberechnung.



Dem Ziel so nah und doch so fern. 🔍



Einmal um die Boxen kommt man zum Ziel. 🔍

Bei einer reinen Abstandsberechnung findet der Roboter nicht ins Ziel, da er seine vermeintlich gute Position (ein Feld neben der Flagge) für eine schlechtere aufgeben müsste, ehe er den Eingang findet.

Way2

Das Testspielbrett Way2 zeigt den interessanten Fall, dass das Ziel nicht durch eine Wand, sondern durch Löcher abgegrenzt ist. Ein paar KIs hatten dies nicht beachtet und fanden nicht ins Ziel.

Way3

Das letzte Testspielbrett zeigt eine Besonderheit, die nur die wenigsten KIs im Test beherrschten. Zum Ziel kommt man nur, indem man einmal um alle Boxen herumläuft und nicht versucht auf dem Förderband abzubiegen.

Viele KIs haben aber die Wegfindung so berechnet, als könnte man auf dem Förderband einfach nach rechts abbiegen und direkt auf die Zielflagge zulaufen. In dem Fall versucht der Roboter ständig diesem Weg zu folgen, kommt aber niemals ins Ziel.

Die einzelnen Lösungen

Es sollen nun alle Lösungen mitsamt den Programmierern kurz vorgestellt werden.

Spielstufe leicht

In dieser Kategorie gab es leider keine Einsendung, dazu später mehr.

Spielstufe mittel

Die Spielstufe mittel ist mit zwei Teilnehmern etwas unterrepräsentiert, aber zumindest kam ein Wettbewerb zustande.

Stefan Haller

Stefan Haller programmiert seit ca. fünf bis sieben Jahren. Den Wettbewerb hat er genutzt, um sich in Python einzuarbeiten und damit seine KI zu erstellen. Daneben konnte er sich auch erstmals mit der KI-Programmierung beschäftigen. Die Umsetzung hat sich insgesamt über vier bis fünf Wochen hingezogen.

Die KI testet alle Kartenkombinationen durch und wählt die aus, die am nächsten am Ziel liegt. Dabei wird auf eine Pfadkarte zurückgegriffen, die den Weg prüft und auch Wände berücksichtigt. Der gesamte Vorgang lässt sich gut parallelisieren, was auch umgesetzt wurde.

[Download der KI von Stefan Haller](#)

Horst Hanewinkel

Horst Hanewinkel programmiert seit ca. 30 Jahren, aber nur im privaten Bereich. Pascal ist die Sprache seiner Wahl, da sie einfach nachvollziehbar ist, aber dennoch mächtig genug, um Verwirrung zu schaffen. Die Teilnahme am Wettbewerb war eine gute Abwechslung zu seinem ganz andersartigem Alltag.

Die KI trägt für alle Felder mittels „Floodfill“ die Abstände zum Ziel ein, wobei Schieber und Förderbänder unberücksichtigt bleiben. Danach werden alle Kartenkombinationen erstellt und sortiert, sodass man gleichartige Kombinationen leichter verwerfen kann. Das beste Ergebnis wird dann gespeichert.

[Download der KI von Horst Hanewinkel](#)

Spielstufe schwer

In der Spielstufe schwer gab es die meisten Einsendungen. Insgesamt zehn Teilnehmer haben Ihre KI eingereicht, um den Roboter zum Ziel zu bringen.

Stefan Agner

Stefan Agner programmiert seit knapp 10 Jahren privat und beruflich. Seine KI ist in Java umgesetzt, da er sich mit der Sprache gut auskennt und so schnell eine Lösung präsentieren konnte. Für die Umsetzung hat er insgesamt ca. drei Tage gebraucht. Teilgenommen hat er, weil es für ihn eine Herausforderung war, eine KI zu programmieren und weil ihm die Dokumentation und Rahmenbedingungen des Wettbewerbs zugesagt haben.

Die KI liest zuerst das Spielbrett ein und bewertet jedes Feld in Abhängigkeit zur Entfernung zum Ziel. Dabei wird beim Ziel mit der höchsten Bewertung gestartet und jedes erreichbare Feld erhält eine um 1 dekrementierte Zahl. Ist ein Feld noch nicht bewertet und soll eine höhere Bewertung erhalten, wird diese gesetzt. Danach wird von allen Kartenkombinationen die ausgewählt,

bei der der Roboter am Ende auf dem höchst bewerteten Feld steht und in Richtung eines noch höher bewerteten schaut.

[Download der KI von Stefan Agner](#)

Mario Fuest

Mario Fuest programmiert erst seit ca. einem Jahr und hat sich für eine KI in Python entschieden, da es eine gute Einsteigerprogrammiersprache ist. Die Umsetzung hat sich dabei über die gesamten zwei Monate hinzogen, die der Wettbewerb andauerte. Teilgenommen hat er, weil er sich vom letzten Wettbewerb aus verbessern wollte.

Die KI merkt sich zu jedem Feld, wo man wie hinkommen kann und wie „teuer“ der Zug ist. Weiterhin ist sie in der Lage, das Ganze rückwärts darzustellen. Daraus lässt sich dann errechnen, welche Kosten man aufbringen muss, um von dem Feld in der Richtung das Ziel zu erreichen. Die KI nimmt nun einfach die acht Karten und wählt die fünf so aus, dass die oben beschriebene Minimalkosten plus die Zugkosten möglichst gering ausfallen.

[Download der KI von Mario Fuest](#)

Jan Gosmann

Jan Gosmann programmiert ungefähr seit 1999 und hat sich für eine KI in C++ entschieden, um die bereitgestellten Bibliotheken nutzen zu können. Die Umsetzung dauerte dabei nur zwischen fünf und acht Stunden. Daneben konnte er so auch mal wieder etwas in C++ programmieren.

Die KI testet alle Kartenkombinationen durch und wählt dann die aus, bei deren Position der Roboter (mit Beachtung von Hindernissen) die wenigsten Schritte zum Ziel hat. Diese Zahl der Schritte wird über eine Breitensuche ausgehend von der Zielposition ermittelt und für jedes Feld des Spielbrettes im Voraus gespeichert. Zusätzlich fließt auch die Blickrichtung des Roboters ein. Diese sollte möglichst zu einem Feld mit geringerer Entfernung zum Ziel sein.

[Download der KI von Jan Gosmann](#)

Andreas Hubmer

Andreas Hubmer programmiert seit ca. sieben Jahren, die letzten vier davon in seinem Informatikstudium. Für den Einsatz von Python sprach für ihn die schnelle Umsetzbarkeit, wobei ihn vor allem die interaktive Konsole unterstützte. Die Umsetzung der KI hat ca. 15 Stunden in Anspruch genommen. Teilgenommen hat er, weil er die Problemstellung interessant und die Ruby-GUI sehr hilfreich beim Testen fand.

Die KI berechnet zuerst für jedes Feld und jede Blickrichtung Distanzwerte zum Ziel. Je weiter weg ein Feld vom Ziel ist, desto höher ist dieser Wert. Das Ziel selbst hat Distanz 0. Im zweiten Schritt werden alle Auswahlen von fünf der acht möglichen Karten durchprobiert und am Spielfeld, ausgehend von der aktuellen Position, simuliert. Jene Karten, mithilfe derer das Feld mit der geringsten Distanz erreicht werden kann, werden schließlich ausgewählt.

[Download der KI von Andreas Hubmer](#)

Jens Jährig

Jens Jährig programmiert erst seit ca. fünf Monaten richtig und hat den Wettbewerb genutzt, um die Sprache Python zu erlernen. Für Python hat er sich entschieden, weil die Skriptsprache vielseitig nutzbar (Konsolen-, GUI- und Web-Anwendungen) und plattformübergreifend ist und daneben eine einfache und klare Syntax hat.

Die KI ist innerhalb von drei Wochen mit ein paar Stunden je Tag entstanden. Dabei erstellt eine einfache Logik (es werden nur Wände berücksichtigt) eine Distanz-Map des Spielbrettes. Danach werden alle nicht identischen Kartenkombinationen getestet und die gewählt, die den Roboter auf der Distanz-Map am weitesten nach vorne (also Richtung Ziel) bringt.

[Download der KI von Jens Jährig](#)

Momme Maraun

Momme Maraun programmiert bereits seit 29 Jahren. Seine KI hat er in Java umgesetzt, da er schon zu lange aus der C++-Programmierung heraus ist. Die Umsetzung hat dabei ca. 16 Stunden gedauert. Teilgenommen hat er, weil seine KI beim ersten **freiesMagazin**-Wettbewerb [4] nicht das konnte, was er wollte.

Die KI erstellt vom Ziel aus rückwärts eine Entfernungstabelle und beachtet dabei auch Sonderfälle wie Laufbänder, sodass eine Bewegung gegen die Laufrichtung des Bandes schlechter gewertet wird. Danach werden alle Kartenkombinationen durchgetestet und beste genommen.

[Download der KI von Momme Maraun](#)

Alexander Mergel

Alexander Mergel programmiert seit etwa 10 Jahren rein privat. Seine KI hat er in C++ programmiert, da es die einzige Sprache ist, die er einigermaßen sicher beherrscht und die Referenzimplementierung auch in C++ vorlag. Insgesamt hat er ca. 40 Stunden für die KI-Umsetzung investiert, wobei ein großer Teil für die Suche nach einem Algorithmus zur Berechnung des exakten Erwartungswerts investiert wurde. Dabei spielte es auch eine Rolle, dass er Wahrscheinlichkeitsrechnung sehr interessant findet und dies zur Teilnahme am Wettbewerb geführt hat.

Für die KI wird zuerst ein Graph mit den 12x12x4 möglichen Positionen des Roboters erstellt. Von jedem Knoten gibt es je Spielkartentyp eine ausgehende Kante, die die Kosten angibt. Für jeden Knoten wird dann eine relativ einfache Heuristik verwendet, um einen Erwartungswert der Kosten bis zum Ziel für eine optimal ziehende KI zu berechnen. Danach werden wie gewohnt alle Kombinationen durchgetestet und die mit dem besten Erwartungswert genutzt.

[Download der KI von Alexander Mergel](#)

Tom Richter

Tom Richter programmiert seit 1995 in eher unregelmäßigen Abständen, was ihm aber dennoch nicht die Freude nimmt. Seine KI ist in C++ erstellt, da er so auf die vorhandenen Bibliotheken der Engine zurückgreifen konnte. Daneben wollte er C++ schon immer mal kennen lernen. Für die Umsetzung hat er sechs bis sieben Nächte gebraucht. Teilgenommen hat er vor allem aus

Spaß am Programmieren, Ausdenken einer KI und weil er sich im Wettbewerb mit anderen messen wollte.

Beim ersten Aufruf analysiert die KI alle möglichen Positionen auf dem Spielbrett und ordnet jeder Position eine Punktzahl zu. Das Zielfeld bekommt eine sehr hohe Punktzahl zugeordnet, alle anderen die Punktzahl 0. Nun werden neue Punktzahlen berechnet, die sich aus dem Erwartungswert der Punktzahlen der Endfelder eines zufälligen Zugs (eine Karte+Bewegungen des Spielfelds) ergeben. Nach mehrmaliger Wiederholung spiegeln die Punktzahlen auf dem Spielbrett die Erreichbarkeit des Ziels von dieser Position aus wieder. Danach wird nach einer Vorauswahl die Kartenkombination gewählt, die die größte Punktzahl bringt.

[Download der KI von Tom Richter](#)

Frank Roth

Frank Roth programmiert, seit er 16 Jahre alt ist. Er nahm bereits am ersten Programmierwettbewerb teil und da dieser ihm Spaß machte, hat er erneut teilgenommen. Seine KI ist in C++ realisiert, da die Engine auch in C++ umgesetzt wurde. Für die Umsetzung hat er dabei insgesamt 12 Stunden gebraucht.

Im ersten Schritt ermittelt die KI für jede Kartenposition die zu gehende Entfernung zum Ziel. Förderbänder gegen die Laufrichtung werden dabei als schlechter angesehen. Im zweiten Schritt

werden alle möglichen Kartenkombinationen gebildet und daraus die ermittelt, welche die KI am weitesten voranbringt.

[Download der KI von Frank Roth](#)

Martin Sommer

Martin Sommer programmiert seit 1990 und hatte damals mit Basic auf einem C64 angefangen. Beruflich arbeitet er fast ausschließlich mit .Net unter Microsoft Windows und hat sich daher auch für diese Sprache und das Mono-Framework entschieden – auch, um dieses näher kennen zu lernen. Die erste Umsetzung der KI hat ca. 10-15 Stunden gedauert, die Fehlersuche danach aber noch einmal so lange. Die Teilnahme am Wettbewerb entstand aufgrund der Tatsache, dass er das Brettspiel „RoboRally“, auf dem der Wettbewerb basiert, vor einigen Monaten geschenkt bekommen hatte.

Die KI nutzt eine Monte-Carlo-Simulation, um die optimale Zugkombination zu berechnen. Zuerst wird dafür ein gerichteter Graph mit vier Knoten für jedes Feld auf dem Spielbrett erstellt. Dazu gibt es sieben ausgehende Kanten für die verschiedenen Kartentypen. Als Zielknoten wird der Knoten angegeben, auf dessen Feld man landet, wenn man eine der sieben Karten ausspielt. Für die Knotengewichtung wird eine leichte Abwandlung des Ford-Bellman-Algorithmus (Abstand aller Knoten zu einem bestimmten Zielknoten mit gewichteten Kanten) auf dem erzeugten Graphen benutzt. Die Gewichtung einer Kante hängt von der Wahrscheinlichkeit der Karte ab, welche dieser Kante zugrunde liegt. Am Ende werden

alle Kartenkombinationen durchgetestet und die beste ausgewählt.

[Download der KI von Frank Roth](#)

Dominik Wagenführ

Dominik Wagenführ ist Initiator des Wettbewerbs und hat neben der dummen Referenz-KI noch eine KI mit Wegfindung in C++ gebaut, um zu schauen, ob das Problem überhaupt lösbar ist.

Für die Grapherstellung verbindet er einfach benachbarte Felder, die nicht durch eine Wand getrennt sind. Ölfelder werden übersprungen und nur die benachbarten Nicht-Ölfelder verbunden (siehe Spielbrett „Oily“). Zusätzlich erhalten Förderbänder entgegen der Laufrichtung des Roboters einen höheren Distanzwert, sodass der Roboter nicht versucht gegen sie anzurennen (siehe Spielbrett „Belts2“). Ganz wichtig, um ein frühzeitiges Ausscheiden zu verhindern, ist auch, dass man mit dem letzten Zug einer Runde nie auf einem Förderbandende stehen bleibt, welches danach in ein Loch oder über den Spielfeldrand führt. Da man nicht weiß, welche Karten folgen werden, würde dies fast immer die Zerstörung des Roboters zur Folge haben.

Nach der Grapherstellung wurde der Algorithmus von Dijkstra benutzt [5], um von der Startposition des Roboters einen Weg zum Ziel zu finden. Der Algorithmus betrachtet dabei jeden Knoten und die Distanz, die zwischen zweien zurückgelegt werden muss. Diese Distanzen werden aufsummiert und der Weg mit der kleinsten Distanz im Graph markiert.

Danach wird versucht, dass der Roboter immer diesem Weg folgt; Alternativrouten werden nicht beachtet (oft gibt es mehrere Wege mit gleicher Kostenanzahl in einem Graphen).

Sollte es keinen Weg geben, wird nur versucht, den Abstand zwischen Roboter und Ziel zu minimieren, was dazu führt, dass der Roboter in dem Fall an Ecken oder Wänden hängen bleibt (siehe Spielbrett „Labytest2“).

Für die Analyse und graphische Darstellung kann der Graph als Graphviz-Datei [6] herausgeschrieben werden. Die Datei übersetzt man dann am besten mit

```
$ neato -Tpdf -o graph.dot
```

in eine PDF-Datei (oder mit -Tpng in PNG). Der Algorithmus von neato hat sich dabei als sinnvoller erwiesen, als das zumeist genutzte dot.

[Download der KI von Dominik Wagenführ](#)

3-D-Spielbrett

In dieser Kategorie gab es leider nur eine Einsendung.

Andreas Schiefer

Andreas Schiefer programmiert seit etwa zehn Jahren und hat eine 3-D-GUI in Python erstellt, die den Roboter durch die Werkhalle fahren lässt. Für Python hat er sich entschieden, weil man mit wenig Code schnell ans Ziel kommt und weil er es privat hauptsächlich einsetzt. Die Umsetzung zog sich über zwei Monate hin. Da Computergrafik ein großes Hobby von ihm ist und er gerade

kein Freizeitprogrammierprojekt hatte, kam der Wettbewerb gerade richtig.

Die 3-D-Modelle wurden alle mit Blender modelliert und mit einem eigenen Export-Skript exportiert. Die Texturen dazu wurden hauptsächlich

mit Gimp bzw. teilweise mit Inkscape erstellt. Das OpenGL-Fenster wird in Python mit Hilfe der Bibliothek **pyglet** erzeugt, welche auch zum Laden der Texturen verwendet wird und die Schnittstelle zu OpenGL in Python zur Verfügung stellt. Gerendert wird das gesamte Spielfeld sowie der

Roboter direkt mit OpenGL, ohne ein Grafik-Framework oder eine Spiele-Engine zu verwenden. Für das User-Interface wird noch „simplui“, ein OpenGL-GUI-Toolkit für pyglet verwendet. Aus dem Ordner lässt sich die GUI per

```
$ ./robotviewer PFAD_ZUM_BOARD.DAT
PFAD_ZUR_GLOBALSEQ.TXT
```

starten. Mit der Maus kann man die Ansicht drehen, zoomen und verschieben.

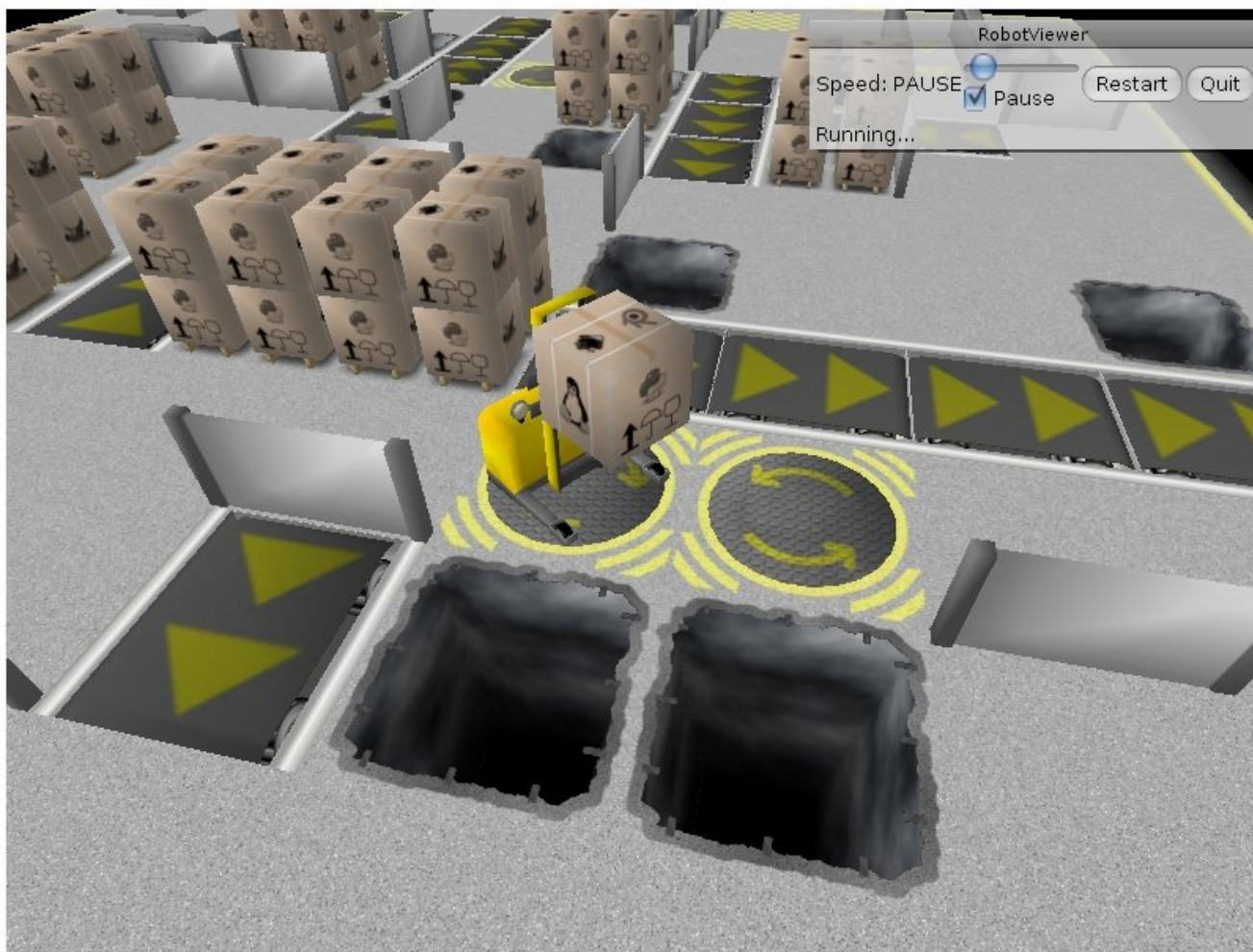
[Download der GUI von Andreas Schiefer](#)

Das Bewertungssystem

Zuerst der Hinweis, dass es keinerlei Plus- oder Minuspunkte auf die Umsetzung, also den Code selbst, gab. Das konnte jeder Programmierer halten, wie er wollte. Kommentare im Programm waren erwünscht, aber niemand wurde disqualifiziert, weil irgendwo einer fehlte. Einzig eine freie Lizenz gehörte zu den Bedingungen, damit **freiesMagazin** den Code auch verteilen darf.

Daneben musste eine KI im Schnitt 75% aller Spiele gewinnen, damit sie in einer Kategorie teilnehmen konnte. Das sollte ein gewissen Niveau der KI-Fähigkeiten sicherstellen und wurde auch von allen Teilnehmern erreicht.

Die KIs mussten auf 12 Spielbrettern mit je 100 vorberechneten Kartenstapeln den schnellsten Weg zum Ziel finden. Dabei gab es recht einfache Spielbretter, aber auch etwas trickreiche. Es wurde dann zusammengezählt, wie viele



Die Roboter-Ameise beim Ausliefern ihrer Fracht. 🔍

Runden, Karten und Bewegungen jede KI insgesamt brauchte und in eine handliche Formel gepresst [7]:

$$P = \frac{4MS - (Z + 0.5Y + K)}{100}$$

- P – Punktezahl insgesamt
- M – Mittelwert der Anzahl aller Züge aller Spieler von gewonnenen Spielen, d.h. Mittelwert(Züge)/Mittelwert(gewonnene Spiele)
- S – Anzahl der gewonnenen Spiele (Gewinn = Zielflagge erreicht)
- Z – Anzahl der aus den gewählten Karten direkt entstandenen Züge bei gewonnenen Spielen
- Y – Anzahl der vom Spielfeld entstandenen Züge bei gewonnenen Spielen
- K – Anzahl der gewählten Karten bei gewonnenen Spielen

Die Wertung unterscheidet sich dabei – wie schon am 14. Oktober vermutet – vom Originalansatz, bei dem der Faktor vor MS noch 2 war. Der Wert war aber so klein, dass es besser war, ein Spiel zu verlieren, damit man weniger Züge braucht, anstatt das Spiel zu gewinnen. Aus dem Grund ist der Faktor nun 4. Insgesamt ändert sich bei einer Änderung des Faktors bei der Platzierung nur sehr wenig.

Die Spielergebnisse und Gewinner

Man kann sich die kompletten Ergebnisse auch als OpenDocument herunterladen [8]. Auch die

Spielbretter mitsamt der vorberechneten Kartensapel stehen zum Download bereit [9]. Die Spielprotokolle sind aber zu groß, um sie selbst gepackt zum Download anbieten zu können.

Spielstufe leicht

Keine Einsendung – kein Gewinner. Damit der 10-Euro-Buchgutschein aber nicht verfällt, haben wir uns entschieden, diesen an den Zweitplatzierten der Spielstufe „schwer“ zu vergeben, da es dort so viele Teilnehmer gab.

Ergebnis der Spielstufe „mittel“						
Platz	Name	Gewonnen	Karten	eigene Züge	fremde Züge	Punkte
1.	Haller	1115	56640	62294	26852	3943,34
2.	Hanewinkel	1074	121589	109365	59994	2463,75

Ergebnis der Spielstufe „schwer“						
Platz	Name	Gewonnen	Karten	eigene Züge	fremde Züge	Punkte
1.	Sommer	1198	27397	35451	14321	1541,74
2.	Richter	1183	27136	35161	13451	1523,53
3.	Mergel	1184	28554	36242	11419	1510,57
4.	Maraun	1180	29983	37735	15798	1451,97
5.	Fuest	1157	29547	37113	12917	1433,91
6.	Hubmer	1178	31042	40427	13897	1420,22
7.	Agner	1178	37594	45785	19343	1273,89
8.	Jährig	1068	32450	39278	16300	1199,77
9.	Roth	1057	31802	38137	17578	1190,69
10.	Gosmann	1030	30208	37249	16328	1171,23

Spielstufe mittel

Da es nur zwei Teilnehmer gab, ist die Tabelle übersichtlich. Der Wert M wurde als 118,09 berechnet.

Herzlichen Glückwunsch an **Stefan Haller** zum ersten Platz in der Stufe mittel und natürlich auch zum 20-Euro-Buchgutschein.

Das Problem mit der KI von Horst Hanewinkel war, dass diese auf dem Spielbrett „Hurdle“ immer den Fehler „ERangeError: Range check error“ brachte und damit kein Spiel zu Ende geführt wurde. Das zweite Problem war eine extrem hohe Zuganzahl bei dem Spielbrett „Oily“, was auf einen Implementierungsfehler schließen lässt.

Hinweis: Inzwischen ist der kleine Fehler der Hanewinkel-KI behoben, die neue Version stand aber noch nicht bei Wettbewerbsende zur Verfügung.

Spielstufe schwer

Und nun wird es etwas spannender, denn immerhin zehn KIs kämpften um den ersten Platz in der Stufe „schwer“ (die KI von Dominik Wagenführ taucht in der Tabelle nicht auf). Der

Wert von M liegt bei 46,78.

Herzlichen Glückwunsch an **Martin Sommer**, der die meisten Spiele gewonnen (1198 von

1200) und damit zurecht den ersten Platz belegt und einen 30-Euro-Buchgutschein verdient hat.

Knapp dahinter liegt die KI von Tom Richter, der etwas weniger Spiele gewinnen konnte, dessen KI aber fast auf gleichem Niveau spielte. (Wie man oben lesen kann, unterscheiden sich die KIs aber auch nur sehr gering.) Wie bei der Spielstufe „leicht“ erwähnt, erhält **Tom Richter** damit den 10-Euro-Buchgutschein. Herzlichen Glückwunsch!

3-D-Spielbrett

Wenig überraschend hat Andreas Schiefer gewonnen, da er als Einziger eine GUI eingesandt hat, die darüber hinaus natürlich auch



Dieser Roboter wird leider platt gemacht. 🔍

funktioniert und die „Sterbeszenen“ schön animiert hat. Im Gegensatz zu den KI-Kategorien war es hier kein Zwang, dass es mindestens zwei Teilnehmer gibt.

Auch schön fanden wir die Animation der Förderbänder, Schieber und Drehräder. Diese konnten dem statischen Spielbrett Leben einhauchen.

Damit herzlichen Glückwunsch an **Andreas Schiefer** zu dem 50-Euro-Buchgutschein.

Fazit

Natürlich möchten wir uns abschließend noch einmal bei allen Teilnehmern bedanken, auch wenn diese nicht gewonnen haben. Der Großteil hat – wie oben zu lesen war – aus Spaß am Programmieren und mit dem Ziel, etwas Neues zu lernen, teilgenommen. Und auch uns hat es Spaß gemacht, den Wettbewerb zu veranstalten.

Etwas Selbstkritik muss aber auch sein: Die Unterscheidung in verschiedene Schwierigkeitsstufen war unsinnig. Der Großteil der Programmierer wollte sowieso das komplette Problem lösen. Und wer in der Stufe „mittel“ einfach einen Rand mit Löchern um das Spielbrett zieht, hat die Stufe „schwer“. Beim nächsten Mal wird es wohl wieder nur eine sinnvolle Kategorie geben.

Etwas schade war die geringe Teilnehmerzahl bei der 3-D-GUI. Wir hatten angenommen, dass es da draußen zahlreiche Grafiker und Programmierer gibt, die sich gerne kreativ austoben. Schade, dass dem nicht so ist. Ob es beim nächsten Mal wieder eine grafische Aufgabe geben wird, ist noch unklar, wobei dies natürlich auch von der Aufgabenstellung des Wettbewerbs abhängt.

Dennoch: Bei nächster Gelegenheit und einer guten Idee wird es sicherlich wieder einen Wettbewerb in dieser oder ähnlicher Form geben.

LINKS

- [1] <http://www.freiesmagazin.de/20090927-zweiter-programmierwettbewerb-gestartet>
- [2] <http://www.freiesmagazin.de/20091214-zweiter-programmierwettbewerb-beendet>
- [3] http://www.freiesmagazin.de/zweiter_programmierwettbewerb
- [4] <http://www.freiesmagazin.de/20090605-gewinner-des-programmierwettbewerbs-steht-fest>
- [5] <http://de.wikipedia.org/wiki/Dijkstra-Algorithmus>
- [6] <http://www.graphviz.org/> 
- [7] <http://www.freiesmagazin.de/20091014-bewertung-des-zweiten-programmierwettbewerbs>
- [8] <ftp://ftp.freiesmagazin.de/2009/2009-10-wettbewerb/ergebnisse.ods>
- [9] <ftp://ftp.freiesmagazin.de/2009/freiesMagazin-2009-10-wettbewerb.tar.gz>

Diesen Artikel kommentieren 



Leserbriefe

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

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

Leserbriefe und Anmerkungen

wmii

❖ [Es] ist anzumerken, dass der beschriebene wmii 3.5 hoffnungslos veraltet ist. Die „aktuelle“ Version ist 3.6 (und selbst diese ist bereits über zwei Jahre alt). Wenn sich jemand wirklich für wmii interessiert und all die schönen neuen Features haben will, wird er nicht umher kommen, sich wmii selbst zu bauen. Unter Arch wahlweise aus dem AUR (**wmii-hg**) oder von einem der vielen User-Repos, die die aktuelle Version anbieten.

Die zwei Jahre alte 3.6- oder die noch ältere 3.5-Konfiguration ist unter wmii 3.9a2 allerdings nicht mehr direkt lauffähig.

Apropos Konfiguration: Anstatt der „Standardkonfiguration“ mit einem Shellskript kann man auch die Konfiguration mittels Ruby- oder Python-Programm vornehmen. Beispiele hierzu sind wmii-hg beigelegt.

Dass der Abschnitt „Programmeinteilung“ die Begriffe Tag, View, Desktop und Client synonym verwendet (und das, wo es unter wmii gar keine „Desktops“ gibt! *g*), sei mal dahingestellt. :)

Und ob die Konfiguration einen Fehler enthält, verrät die Shell, aus der heraus man wmii gestartet hat, im Zweifelsfall also TTY1. Hier könnte man auch

```
wmii &> /var/log/wmii
```

in die **.xinitrc** schreiben, vorausgesetzt, **/var/log/wmii** ist durch den User beschreibbar.

Dirk Sohler (Kommentar)

❖ *Dass die Version 3.5/3.6 veraltet ist, ist klar, aber ich wollte keinen Artikel über einen Fenstermanager schreiben, den man sich selber bauen muss. Auch zu meinem Post auf der Mailingliste von suckless.org meinte der Maintainer von dwm, dass die Version veraltet ist [1]. Daher gibt es zu einem späteren Zeitpunkt vielleicht nochmal einen Artikel zu der neuen Version.*

Bei den Begriffen ist mir wohl ein Fehler unterlaufen. Mit Tag ist der „Desktop“, mit View der Mode und mit Client die jeweilige Anwendung gemeint.

*Das mit der **.xinitrc** kannte ich noch nicht. Wenn, dann sollte es aber*

```
wmii 2>&1 /var/log/wmii
```

heißen, falls die Nutzer keine Bash einsetzen.

Yannic Haupenthal

Erstellen geometrischer Skizzen mit kig

❖ Für alle, die es noch nicht wussten, es gibt auch recht hilfreiche Tools, um SVG-Dateien mehr oder weniger direkt und schön in \LaTeX einzubinden: Ein Python-Skript `svg2tex` [2] [und ein] (ehemaliges) Inkscape-Plugin `svg2latex` [3].

Artur Wachtel (Kommentar)

Yalm-Ende

❖ Schade. Das Wort sagt eigentlich schon alles. Die freien „benutzergenerierten“ Magazine haben m. E. eine wichtige Funktion in unserer Medienlandschaft. Sie kümmern sich um Inhalte, die die kommerziellen Medien (teilweise) wegen der zu kleinen zu erwartenden Leserquote nicht behandeln können. Sie schildern ohne professionelles Makeup Sachverhalte, geben ihre direkte Anwendersicht auf Geschehnisse und Produkte wieder. Unverzichtbar!

Um so schlimmer, dass eines der Besten nun als eigenständiges Produkt verschwunden ist. Allerdings freue ich mich, dass durch den Übergang zu **freiesMagazin** das Potential, das Yalm getragen hat, nicht verloren geht.

Als Herausgeber und Redakteur eines sehr kleinen, auf freiwilliger Arbeit von Usern basierenden, Community-Magazins weiß ich um die Probleme bei der „Beschaffung“ von Inhalten und der Sorge um die nächste Ausgabe. So sehe



ich im Negativen des Abschieds von Yalm auch das Positive der Verstärkung des Teams von freiesMagazin. **Wobo** (Kommentar)

❖ Die Erfahrung zeigt, dass je größer eine Community wird, das Verhältnis zwischen aktiven und passiven Nutzern immer schlechter wird. Ähnlich ist es wohl auch bei den Magazinen. Es gibt zwar immer mehr Leser, aber es kommen leider nicht immer mehr Autoren nach. Hier trifft wohl eher das Gegenteil zu, dass die Gewohnheit, dass es das Magazin schon so lange gibt, zur Annahme verleitet, man braucht keine Hilfe mehr.

Die Integration der Yalm-Mitglieder ist rückwirkend betrachtet sehr klein ausgefallen. Nur ein Korrektor hat den Weg zu uns ins Team gefunden, zumal wir auch schon eine recht große Mannschaft haben. Dafür sind aber ein paar neue Autoren bei uns aufgeschlagen, die hoffentlich auch in Zukunft den einen oder anderen Artikel beisteuern.

Dominik Wagenführ

Korrekturen und Ergänzungen

Lösungen zum Artikel „Shebang – All der Kram“, freiesMagazin 12/2009 [4]

Obwohl im Artikel stand, dass sich die beiden Skripte der ersten Aufgabe nicht unterscheiden, wiesen uns die beiden Leser Joachim Breitner und Holger Jakobs darauf hin, dass dies nicht stimmt:

```
$ cat x
#!/bin/sh
echo $0 $@
$ cat y
#!/bin/echo
$ ./x a "b c"
./x a b c
$ ./y a "b c"
./y a b c
```

Wie man sieht, wird das zweite Argument `b c` einmal korrekt ausgegeben und einmal verschwinden die „überflüssigen“ Leerzeichen. Der

Grund liegt darin, dass die Angabe `$@` das Argument `b c` als zwei Argumente an `echo` übergibt.

Man sollte also bei solchen Skripten immer darauf achten, dass Variablen und Parameter auch Leerzeichen enthalten können, die ggf. als Trennzeichen fungieren.

LINKS

- [1] <http://lists.suckless.org/dev/0912/2676.html>
- [2] <http://github.com/Oni/svg2tex>
- [3] <http://sites.google.com/site/richardhenwood/svg2latex2>
- [4] <http://www.freiesmagazin.de/freiesMagazin-2009-12>

Die Redaktion behält sich vor, Leserbriefe gegebenenfalls zu kürzen.

Die Leserbriefe kommentieren



„Still Raw“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/473>

Veranstaltungskalender

Messen

Veranstaltung	Ort	Datum	Eintritt	Link
4. LIT-OL	Oldenburg	13.02.2010	frei	http://www.lit-ol.de
FOSSGIS	Osanbrück	02.03.-05.03.2010	-	http://www.fossgis.de/konferenz/wiki/Main_Page
CeBIT	Hannover	02.03.-06.03.2010	38 EUR	http://www.cebit.de
12. Chemnitzer Linux-Tage	Chemnitz	13.03.-14.03.2010	5 EUR / 3 EUR erm.	http://chemnitzer.linux-tage.de
OpenExpo	Bern	24.03.-25.03.2010	-	http://www.openexpo.ch
Augsburger Linux-Infotag	Augsburg	27.03.2010	frei	http://www.luga.de
Grazer Linuxtage	Graz	25.04.2010	frei	http://linuxwochen.at
Linuxwochen Wien	Wien	06.05.-08.05.2010	frei	http://linuxwochen.at
LinuxTag	Berlin	09.06.-12.06.2010	12 EUR	http://www.linuxtag.de

(Alle Angaben ohne Gewähr!)

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

Vorschau

freiesMagazin erscheint immer am ersten Sonntag eines Monats. Die Februar-Ausgabe wird voraussichtlich am 7. Februar unter anderem mit folgenden Themen veröffentlicht:

- Rezension: Das Betriebssystem GNU/Linux
- Machinarium – In der Welt der Maschinen

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

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.

freiesMagazin-Index 2009

3

3D-Desktop

KWin 4.2 – Neues aus dem Land der wabernden Fenster 02/2009

A

Audio

Amarok 2: Der Wolf setzt zum Sprung an 02/2009

Audio und Video in Firefox 3.5 „Shiretoko“ 08/2009

B

Barrierefreiheit

Debian 5.0-Installation mit einer Braillezeile 06/2009

BeOS

ZevenOS – Linux goes BeOS 07/2009

ZevenOS meets Debian Lenny 08/2009

Bildbearbeitung

Scannen mit Skanlite 02/2009

Browser

Projektvorstellung Opera 01/2009

Songbird 1.0 erschienen 01/2009

Kurztipp: Songbird installieren 03/2009

Audio und Video in Firefox 3.5 „Shiretoko“ 08/2009

Buch

Rezension: Coding for Fun – IT-Geschichte zum Nachprogrammieren 04/2009

Rezension: Just for Fun – The Story of an Accidental Revolution 06/2009

Internetrecht – als Donationware! 07/2009

Debian GNU/Linux – Das umfassende Handbuch 09/2009

C

Community

freiesMagazin-Umfrage gestartet 01/2009

Auswertung der freiesMagazin-Umfrage 03/2009

Kurztest: Etherpad 04/2009

Linux-Herstellerunterstützung: DVD-Brenner-Update 04/2009

Rückblende zur CeBit 2009 04/2009

freiesMagazin-Programmierwettbewerb 04/2009

ECDL/ICDL wagt sich an OpenOffice.org 06/2009

Linuxtag – Erlebte Community 08/2009

41. DANTE-Mitgliedertagung in Esslingen 10/2009

Die GUI zum Programmierwettbewerb 10/2009

Erklärungen zum zweiten Programmierwettbewerb 10/2009

FAQ zum Programmierwettbewerb 10/2009

Rückblick: Ubuntu-Treffen auf der Ubucon 2009 11/2009

Weitere Informationen zum Programmierwettbewerb 11/2009

Google Summer of Code 12/2009

CRM

C.U.O.N. und mobile Dienste 07/2009

Was bieten freie CRM-Lösungen? – Im Vergleich: Sugar- 07/2009

CRM und vtiger CRM

D

Dateisystem

Das Dateisystem ext4 04/2009

Dateiverwaltung

Der KDE-Dateimanager Dolphin 02/2009

Der Logical Volume Manager 03/2009

Datenverwaltung

Datensammlung mit Tellico 04/2009

C.U.O.N. und mobile Dienste 07/2009

Debian	
Nachrichtenschnippel	01/2009
gNewSense – Das freie Ubuntu	04/2009
Debian GNU/Linux 5.0 „Lenny“ – Der Zustand der Distribution	05/2009
Geh spielen! – GoPlay zur Suche nach Software einsetzen	05/2009
Debian 5.0-Installation mit einer Braillezeile	06/2009
ZevenOS meets Debian Lenny	08/2009
Debian GNU/Linux – Das umfassende Handbuch	09/2009
Linux auf der Nintendo Wii	09/2009
Desktop	
Installation und Einrichtung von Avant Window Navigator	01/2009
LXDE im Test	03/2009
OS-Tan-Linux-Desktothintergründe	03/2009
wmii: Weitere Konfiguration	12/2009
E	
Editor	
Kurztest: Etherpad	04/2009
Kurztest: Bepin	05/2009
Kurztipp: Splitscreens mit Vim	05/2009
Kurztipp: Markierungen mit Vim	07/2009
Kurztipp: Suchen und Finden mit Vim	10/2009
Ordnung ins Chaos mit Org-Mode	10/2009
F	
Fedora	
Die zehnte Fedora-Ausgabe vorgestellt	01/2009
Kurze Vorstellung der elften Fedora-Ausgabe	08/2009
Fenstermanager	
wmii: Weitere Konfiguration	12/2009
G	
gNewSense	
gNewSense – Das freie Ubuntu	04/2009
GNOME	
Nautilus-Scripts	11/2009
Google	
Google Summer of Code	12/2009
Grafik	
Panoramabilder erstellen mit hugin	08/2009
Der Raytracer POV-Raytracer	09/2009
Von der Schaltung zur fertigen Platine mit EAGLE	10/2009
ASCII-Art mal anders mit ditaa	11/2009
Erstellen geometrischer Skizzen mit kig	12/2009
Open Clip Art – Die vergessenen Grafiken	12/2009
H	
Hardware	
Neues aus der Welt der Mobilgeräte	01/2009
Review: Motorola ROKR E8	03/2009
Linux-Herstellerunterstützung: DVD-Brenner-Update	04/2009
Openmoko – Free your phone	04/2009
Das Netbook-Touchpad: Ein ergonomisches Disaster	05/2009
AVR-Mikrocontroller unter Linux	06/2009
Linux auf der Nintendo Wii	09/2009
Grundbegriffe der Virtualisierung	10/2009
Von der Schaltung zur fertigen Platine mit EAGLE	10/2009
Hardwarekompatibilität unter Linux	11/2009
I	
Internet	
Nachrichtenschnippel	01/2009
Oxite: Microsoft auf neuen Wegen	01/2009

Internet (Fortsetzung)

Songbird 1.0 erschienen	01/2009
Kurztipp: Songbird installieren	03/2009
Kurztest: Bepin	05/2009
RSS-Feeds: Technik und Nutzen kurz vorgestellt	05/2009
Audio und Video in Firefox 3.5 „Shiretoko“	08/2009

K**KDE**

Amarok 2: Der Wolf setzt zum Sprung an	02/2009
Der KDE-Dateimanager Dolphin	02/2009
Die digitale Brieftasche: KWallet	02/2009
Eine kleine Einführung in KDevelop	02/2009
KDE4 wird erwachsen	02/2009
KWin 4.2 – Neues aus dem Land der wabernden Fenster	02/2009
Scannen mit Skanlite	02/2009

Kommerzielle Software

Duden Korrektor im frischen Wind	09/2009
----------------------------------	---------

Konsole

Kurztipp: Bastelstunde mit Pdftk	03/2009
Kurztipp: Splitscreens mit Vim	05/2009
Kurztipp: Markierungen mit Vim	07/2009
Kurztipp: Suchen und Finden mit Vim	10/2009
grep – Eine kleine Einführung	10/2009
Shebang – All der Kram	11/2009
Lösungen zum Artikel „Shebang – All der Kram“	12/2009

L**LaTeX**

Magazinerstellung mit \LaTeX – Das Layout	08/2009
41. DANTE-Mitgliedertagung in Esslingen	10/2009
ASCII-Art mal anders mit ditaa	11/2009

Linux allgemein

Distributionen aktuell	01/2009
Kernel-Rückblick	01/2009
Nachrichtenschnippel	01/2009
Januar und Februar im Kernelrückblick	03/2009
OS-Tan-Linux-Desktothintergründe	03/2009
Das Dateisystem ext4	04/2009
Ein Tuz für den Kernel	04/2009
Linux-Herstellerunterstützung: DVD-Brenner-Update	04/2009
Der April im Kernel-Rückblick	05/2009
AVR-Mikrocontroller unter Linux	06/2009
Der Mai im Kernel-Rückblick	06/2009
Die Vorgeschichte von Linux	06/2009
Mobiles Linux mit Moblin	06/2009
Rezension: Just for Fun – The Story of an Accidental Revolutionary	06/2009
Der Juni im Kernel-Rückblick	07/2009
Der Juli im Kernel-Rückblick	08/2009
Über magische Tasten aus der Klemme – SysRQs	08/2009
Der August im Kernel-Rückblick	09/2009
Einfach Easy Peasy	09/2009
Linux auf der Nintendo Wii	09/2009
Der September im Kernel-Rückblick	10/2009
Gefahren von Fremdquellen am Beispiel PPA	10/2009
Der Oktober im Kernel-Rückblick	11/2009
Hardwarekompatibilität unter Linux	11/2009
Der November im Kernel-Rückblick	12/2009

Lizenz

Free Software Foundation verklagt Cisco wegen GPL-Verletzung	01/2009
GPL-Lizenzschwierigkeiten	05/2009
Internetrecht – als Donationware!	07/2009

M**Magazin**

freiesMagazin-Umfrage gestartet	01/2009
Auswertung der freiesMagazin-Umfrage	03/2009
freiesMagazin-Programmierwettbewerb	04/2009
Informationen zum Programmierwettbewerb	05/2009
Das Ende des Programmierwettbewerbs	06/2009
Magazinerstellung mit L ^A T _E X – Das Layout	08/2009
freiesMagazin sucht Unterstützung!	08/2009
Die GUI zum Programmierwettbewerb	10/2009
Erklärungen zum zweiten Programmierwettbewerb	10/2009
FAQ zum Programmierwettbewerb	10/2009
Weitere Informationen zum Programmierwettbewerb	11/2009

Microsoft

Oxite: Microsoft auf neuen Wegen	01/2009
----------------------------------	---------

Mobil

Neues aus der Welt der Mobilgeräte	01/2009
Review: Motorola ROKR E8	03/2009
Openmoko – Free your phone	04/2009
Das Netbook-Touchpad: Ein ergonomisches Disaster	05/2009
Mobiles Linux mit Moblin	06/2009
Einfach Easy Peasy	09/2009

Multimedia

Songbird 1.0 erschienen	01/2009
Amarok 2: Der Wolf setzt zum Sprung an	02/2009
Kurztipp: Songbird installieren	03/2009
Ausgeflasht: Ein Versuch mit Gnash und Swfdec	05/2009
easyVDR – The easy VDR Distribution	07/2009
Audio und Video in Firefox 3.5 „Shiretoko“	08/2009

N**Netzwerk**

VNUML – Ein Netzwerksimulator mit User-Mode-Linux	08/2009
---	---------

O**Office-Suite**

ECDL/ICDL wagt sich an OpenOffice.org	06/2009
Erfahrungsbericht: OpenOffice.org 3.0	06/2009
Duden Korrektor im frischen Wind	09/2009
Open Clip Art – Die vergessenen Grafiken	12/2009

OpenSolaris

Das neue „Sonnensystem“ – Ein Blick auf OpenSolaris 2008.11	01/2009
---	---------

openSUSE

openSUSE 11.1	03/2009
---------------	---------

P**Paketverwaltung**

Geh spielen! – GoPlay zur Suche nach Software einsetzen	05/2009
Gefahren von Fremdquellen am Beispiel PPA	10/2009

PDF

Kurztipp: Bastelstunde mit Pdftk	03/2009
PDF-Zerteilung nach Maß	07/2009
Auf Klick folgt Schnitt: PDF Chain	08/2009
PDF-Betrachter im Test	08/2009
Xpdf	09/2009

Programmierung

Eine kleine Einführung in KDevelop	02/2009
Installation von KBasic	04/2009
Programmieren in KBasic – Eine kurze Einführung	04/2009

Programmierung (Fortsetzung)

freiesMagazin-Programmierwettbewerb	04/2009
GPL-Lizenzenschwierigkeiten	05/2009
Informationen zum Programmierwettbewerb	05/2009
NaturalDocs – Quelltexte natürlich dokumentieren	08/2009
Der Raytracer POV-Raytracer	09/2009
Die GUI zum Programmierwettbewerb	10/2009
Erklärungen zum zweiten Programmierwettbewerb	10/2009
FAQ zum Programmierwettbewerb	10/2009
Java, Teil 1 – Einführung in eine moderne Sprache	10/2009
Von der Schaltung zur fertigen Platine mit EAGLE	10/2009
Nautilus-Scripts	11/2009
Shebang – All der Kram	11/2009
Testen mit Perl	11/2009
Weitere Informationen zum Programmierwettbewerb	11/2009
Google Summer of Code	12/2009
Java, Teil 2 – Grafische Programmierung mit Swing	12/2009
Lösungen zum Artikel „Shebang – All der Kram“	12/2009

Präsentation

ASCII-Art mal anders mit ditaa	11/2009
--------------------------------	---------

R**Rezension**

Rezension: Coding for Fun – IT-Geschichte zum Nachprogrammieren	04/2009
Rezension: Just for Fun – The Story of an Accidental Revolution	06/2009
Internetrecht – als Donationware!	07/2009
Debian GNU/Linux – Das umfassende Handbuch	09/2009

RSS

RSS-Feeds: Technik und Nutzen kurz vorgestellt	05/2009
--	---------

S**Sicherheit**

Die digitale Brieftasche: KWallet	02/2009
Kennwortgedächtnis auslagern mit KeePassX	05/2009
Gentoo Linux mit Festplattenverschlüsselung	07/2009
Partitionen verschlüsseln mit dm-crypt und LUKS	07/2009
SSH-Port absichern	11/2009

Spiele

Nachrichtenschnippel	01/2009
World of Goo – Die Bälle sind los	03/2009
Toribash – Kampf mit (der) Physik	06/2009
Slingshot – Steinschleuder im Weltraum	09/2009
Mad Skills Motocross	12/2009

Systemverwaltung

Der Logical Volume Manager	03/2009
----------------------------	---------

T**Terminal**

Keine Angst vor awk – ein Schnelleinstieg	06/2009
---	---------

Terminverwaltung

C.U.O.N. und mobile Dienste	07/2009
Ordnung ins Chaos mit Org-Mode	10/2009

Tipps & Tricks

Kurztipp: Bastelstunde mit Pdftk	03/2009
Kurztipp: Songbird installieren	03/2009

U**Ubuntu**

Ubuntu 8.10 – Die neue Version „Intrepid Ibex“ vorgestellt	01/2009
Kurztipp: Songbird installieren	03/2009

Ubuntu (Fortsetzung)

gNewSense – Das freie Ubuntu	04/2009
ZevenOS – Linux goes BeOS	07/2009
Rückblick: Ubuntu-Treffen auf der Ubucon 2009	11/2009
Ubuntu 9.10 – Die neue Version „Karmic Koala“ vorgestellt	12/2009

V

Veranstaltung

Rückblende zur CeBit 2009	04/2009
41. DANTE-Mitgliedertagung in Esslingen	10/2009
Rückblick: Ubuntu-Treffen auf der Ubucon 2009	11/2009

Video

easyVDR – The easy VDR Distribution	07/2009
Audio und Video in Firefox 3.5 „Shiretoko“	08/2009

Virtualisierung

Virtualisierung im Allgemeinen und KVM unter Proxmox VE	05/2009
im Speziellen	
Grundbegriffe der Virtualisierung	10/2009

W

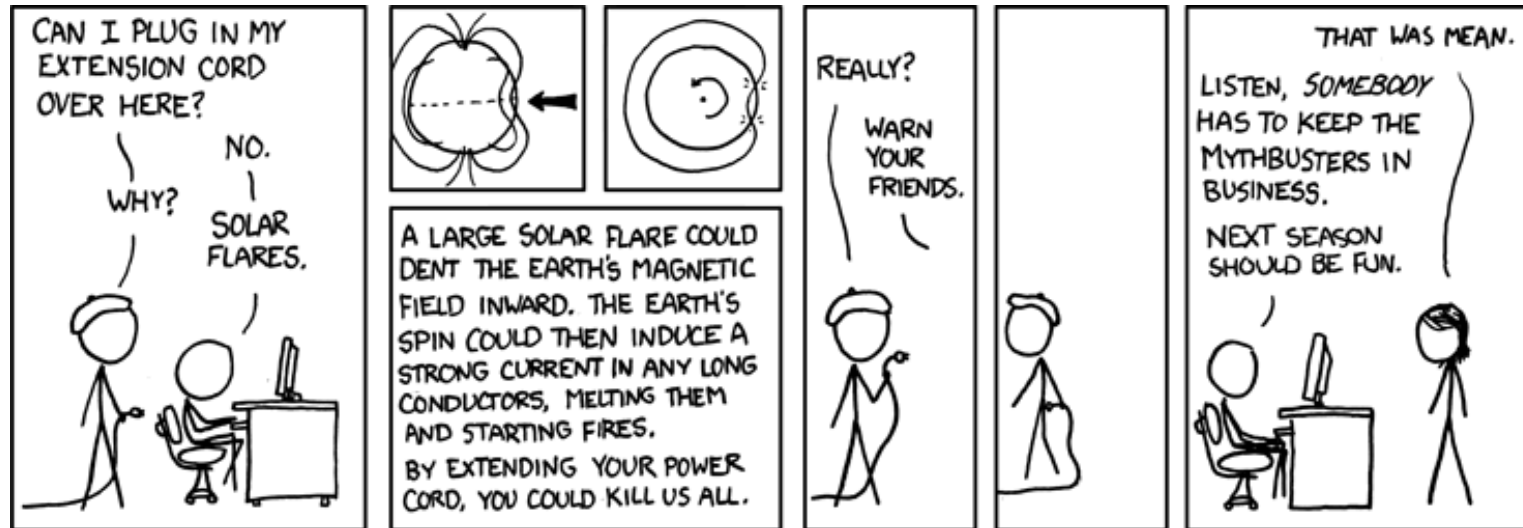
Wissen und Bildung

Erstellen geometrischer Skizzen mit kig	12/2009
---	---------

Z

ZevenOS

ZevenOS – Linux goes BeOS	07/2009
ZevenOS meets Debian Lenny	08/2009



„Induced Current“ © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/509>

Impressum

freiesMagazin erscheint als PDF und HTML einmal monatlich.
Redaktionsschluss für die Februar-Ausgabe: 24. Januar 2009

Kontakt

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

Autoren dieser Ausgabe

Hans-Joachim Baader **S.3**
Heinz-M. Graesing **S.33**
Karsten Günther **S.23**
Ralf Hersel **S.17**
Mathias Menzer **S.15**
Markus Schulze **S.20**
Dominik Wagenführ **S.10, S.36**

ISSN 1867-7991

Erscheinungsdatum: 10. Januar 2010

Redaktion

Dominik Honnef Thorsten Schmidt
Dominik Wagenführ (Verantwortlicher Redakteur)

Satz und Layout

Ralf Damaschke Yannic Hauptenthal
Marcus Nelle Sebastian Schlatow

Korrektur

Daniel Braun Frank Brungräber
Raoul Falk Karsten Schuldt

Veranstaltungen

Ronny Fischer

Logo-Design

Arne Weinberg ([GNU FDL](#))

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

Soweit nicht anders angegeben, stehen alle Artikel 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](#).