

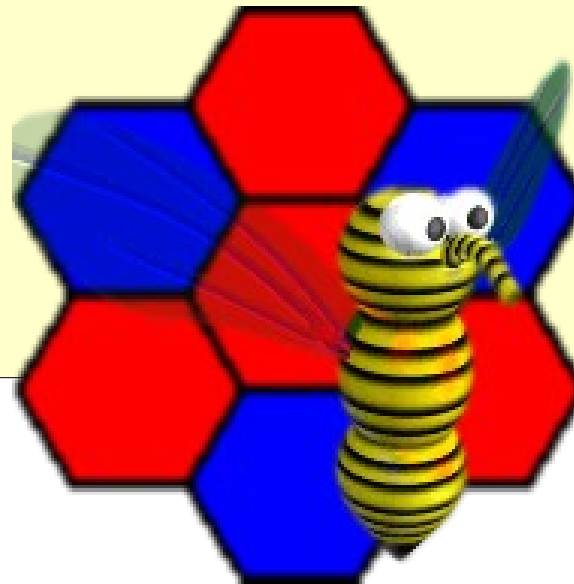


Wine

*Partitionierung mit
fdisk*



Exklusiv für Yalm:



The Hive Game

Vorwort

Es gestaltet sich wider Erwarten ungewöhnlich problematisch, wenn man im Rahmen des Vorworts stets interne Neuerungen oder andere, nennenswerte Änderungen groß ankündigen möchte, eben diese aber in einem Monat ausbleiben. Man stelle sich nur den Redakteur vor, fieberhaft nach einer passenden Idee suchend, hektisch an der vierten, noch immer dampfenden Tasse Kaffee nippend und immer wieder die Finger auf die Tastatur legend, nur um einige Zeilen zu schreiben welche im Anschluss daran lediglich mit einem missmutigen Kopfschütteln wieder gelöscht werden.

Aber gut, wir könnten nun behaupten, dass derzeit keine weiteren Entwicklungen vonnöten seien, schließlich funktioniert der interne Ablauf ohne nennenswerte Schwierigkeiten. Aber in Anbetracht der Tatsache, dass der geneigte Leser dies schon mehr als einmal gehört haben dürfte, sollte womöglich das Thema gewechselt werden... Zu einem anderen Zeitpunkt würde man nun einer mehr oder weniger geschickten Überleitung begegnen, hier aber muss erst einmal ein anderer Ansatzpunkt gefunden werden.

Wir könnten uns darüber auslassen, dass Vorworte neuerdings zu inhaltslosen Textblöcken degenerieren, aber selbst eine umfangreiche Abhandlung über dieses Thema wäre für unsere Zwecke wohl noch zu kurz. Aber was hieltet ihr von einer Erörterung zum in redaktionsinternen Kreisen kursierenden Traktat »Kaffee – Das Grundnahrungsmittel

des Redakteurs«? Auch nicht? Nun gut, aber was gäbe es noch für Möglichkeiten? Wir könnten eine »Nachricht des Monats«, wie etwa die Veröffentlichung von »Deine Lieblingsdistribution 2.0«, aufgreifen und diese im Zuge des Vorworts weiter ausbreiten, aber wollen wir das?

Wobei, eine Meldung gibt es schon, die insbesondere die Redaktion erfreut: Die Februar-Ausgabe von Yalm ist erschienen! Nachdem ihr euch von dieser unerwarteten Nachricht erholt habt, möchten wir nun erwähnen, dass es sich hierbei um ein mit überaus interessanten und lehrreichen Artikeln gefülltes Werk handelt. Und das Beste: In diesem Monat entstand nicht nur eine weitere Ausgabe, nein, sondern auch ein eigens für Yalm entwickeltes Spiel! Mehr hierzu findet ihr im Artikel »[The Hive Game](#)«. Aber auch die Artikel zu [Wine](#) und über das [Partitionieren mit fdisk](#) sind, wie im Grunde genommen das ganze Magazin, überaus lesenswert.

Damit möchten wir, nicht ohne ein befriedigtes Lächeln auf den Lippen, dieses Vorwort zu einem Ende bringen und euch viel Spaß beim Lesen wünschen. In der Zwischenzeit holen wir uns noch einen Kaffee.

Stefan Zaun

sciron@yalmagazine.org

Inhalt

Yalm - Vorwort	2
Vorwort.....	2
Yalm - Rückblick	3
Rückblick.....	3
Yalm - Magazin	4
Zusammenfassung der dritten Ubuntu Developer Week (UDW).....	4
Partitionierung mit fdisk	7
The Hive Game.....	10
Heimnetzwerk aufbauen – The Easy Way (1)	13
Verteilen von E-Mails im Netzwerk.....	21
Wine.....	24
Yalm - Bunte Seite	29
Bunte Seite.....	29
Rekursives Umbenennen – Lösung.....	30
Yalm - Schlussbemerkungen	32
Schlussbemerkungen.....	32

Rückblick

Ext4-Benchmarks

Das Dezember für den Produktiv-Einsatz freigegebene Dateisystem ext4 scheint ein würdiger Nachfolger des ext3-Systems sein. So kann ext4 sowohl schneller lesen als auch schreiben, lässt sich effizienter verwalten und bietet eine schnellere Dateisystemprüfung. Weitere Informationen liefert der Artikel [1] auf Pro-Linux.

Obama setzt auf freie Software

Amerikas neuer Präsident Obama versprach während des Wahlkampfes Kosten in den Behörden zu senken und Vorgänge transparenter zu machen. Dieses Versprechen will er nun u. a. durch Einsatz von OpenSource-Software einhalten. Immerhin werden ca. USD 400 Mrd. (rund EUR 300 Mrd.) für proprietäre Software ausgegeben, ein Großteil könne eingespart werden.

Kritiker bemängeln allerdings, dass es auch eines besseren Finanzmanagements bedarf, um Kosten zu senken. Wieviel wirklich eingespart wird, bleibt also abzuwarten.

Quelle: Ikhaya [2]

KDE 4.2 »The Answer« veröffentlicht

Während KDE 4.0 noch für Entwickler und KDE 4.1 für »Early Adopters« gedacht war, erschien Ende Januar KDE 4.2 für Endanwender. So sind



Die Plasmaoberfläche von KDE 4.2

Screenshot von kde4.de: http://www.kde4.de/?page_id=535

die neuen Technologien, die vor einem guten Jahr eingeführt wurden, ausgereifter und das System wirkt stabiler. Die Nutzer vermissen keine grundlegenden Funktionen mehr und können sich an Eye-Catchern erfreuen, da KDE 4.2 nun etliche Compositing-Effekte mit bringt.

Quelle: Ikhaya [3]

RPM 4.6 erschienen

Das Paketverwaltungsprogramm RPM, welches u. a. von Fedora, OpenSUSE sowie Mandriva genutzt wird, liegt in der Version 4.6 vor. Gemeinsam entwickelte man etliche Neuerungen, eine Liste kann unter [4] abgerufen werden.

So gibt es keine Sperren mehr für zu große Dateien, RPM arbeitet schneller und robuster und Fehlermeldungen sollen klarer sein.

Quelle: Pro-Linux [5]

Lenny veröffentlicht?

Vermutlich wird Debian 5.0 »Lenny« am 14. Februar, also mit dem Erscheinen dieser Ausgabe, veröffentlicht werden. Damit wird Lenny nun endlich Etch ablösen können, welches mittlerweile stark veraltete Software beinhaltet.

Quelle: Pro-Linux [6]

Mario Fuest

keba@yalmagazine.org

Informationen

- [1] ext4-Benchmarks: <http://www.pro-linux.de/berichte/ext4/ext4.html>
- [2] Ikhaya: Open Source im Weißen Haus <http://ikhaya.ubuntuusers.de/2009/01/22/open-source-im-weissen-haus/>
- [3] Ikhaya: KDE 4.2, Codename "The Answer", ist erschienen <http://ikhaya.ubuntuusers.de/2009/01/27/kde-4.2-codename-the-answer-ist-erschienen/>
- [4] Changelog von RPM 4.6: <http://lists.rpm.org/pipermail/rpm-announce/2009-February/000013.html>
- [5] Pro-Linux: RPM 4.6 erschienen: <http://www.pro-linux.de/news/2009/13779.html>
- [6] Pro-Linux: Debian 5.0 für 14. Februar geplant: <http://www.pro-linux.de/news/2009/13756.html>

Zusammenfassung der dritten Ubuntu Developer Week (UDW)

Was ist überhaupt die UDW? Sie hat das Ziel, neuen Ubuntu Entwicklern – und natürlich allen Interessierten – zu zeigen, wie die Entwicklung vonstatten geht, was dabei zu beachten ist, aber vor allem den Einstieg zu erleichtern.

Mit Ausnahme der »Getting Started«-Session, welche in 6 Sprachen parallel gehalten wurde, werden die Sessions in Englisch gehalten und finden im Channel #ubuntu-classroom auf irc.freenode.net statt. In diesem Channel finden regelmäßig Sessions zu den verschiedensten Themen bezüglich Ubuntu statt. Infos darüber findet man auf [1].

Zu sagen ist, dass ich leider nicht die Zeit hatte, allen Sessions beizuwohnen und mich daher im Folgenden auf den englischsprachigen Eintrag von Daniel Holbach auf der Ubuntu-News-Team-Mailingliste stütze. Im Original ist er hier [2] zu finden.

Die Logs aller Sessions finden sich auf der Wiki-Seite [3] im Ubuntuwiki. Aus Gründen der besseren Lesbarkeit haben wir davon abgesehen, jede Session einzeln zu verlinken. Wir bitten euch, bei Bedarf die Logs der gesuchten Session aus der Liste selbst herauszusuchen.

Tag 1

Alles begann mit einer zweistündigen Einführung namens »Getting Started«. Es war das erste Mal, dass diese Einführung in Englisch, Finnisch, Französisch, Deutsch, Italienisch und Spanisch zeit-

gleich stattfand. Für jene, die der englischen Sprache nicht oder nicht sehr mächtig sind, war das eine wichtige Hilfe, um ins »Projekt« Ubuntu einzusteigen. Es wurden einige grundlegende Dinge rund um die Ubuntu-Entwicklung abgedeckt: Unter anderem wurde ein GPG Schlüssel eingerichtet, mit Pbuilder herumgespielt etc. Aber auch über die Ubuntu-Entwicklung im Allgemeinen wurde diskutiert und natürlich wurden zahlreiche Fragen beantwortet. Was Daniel Holbach, einer der Organisatoren, besonders freute, war, dass alle von den »Getting-Started«-Sessions eifrig genug waren, um ihre Fragen anschließend auch in den folgenden englischsprachigen Sessions zu stellen.

Die folgende Session war »Packaging 101« wo das »hello«-Paket unter die Lupe genommen wurde. Obwohl es eigentlich ein eher langweiliges Paket ist – es schreibt nur »Hello World« auf die Konsole -, wurden dadurch viele Packaging-Konzepte erarbeitet und jeder freute sich auf mehr.

Nicolas Valcarel und Barry deFrees hielten anschließend die geniale Session »Working Well With Debian« ab, wo Fragen rund um die Kollaboration mit Debian, einem der wichtigsten »Upstreams« für Ubuntu, beantwortet wurden.

Ted Gould schloss den ersten Tag ab, indem er uns half, die GNOME Technologien besser zu verstehen. In seiner Session »Understanding GNOME technologies« erklärte er perfekt die Basis, auf der GNOME aufbaut. Das Interesse zeigt, dass wir bald eine große Anzahl an motivierten GNOME-Mitarbeitern haben werden.

Tag 2

Tom 'intellectonica' Berger machte den Anfang und sprach über »Launchpad Bug Tracking«. Er begann damit, herauszufinden, wie gut sich jeder mit dem Bug Tracker namens Malone auskennt: Jeder sollte sich auf einer Skala von 0-10 selbst bewerten. Wir hatten Leute an beiden Enden des Spektrums, und Tom hatte für jeden etwas Neues mit dabei. Als Markus Korn sich mit geschätzten 7.99 zu Wort meldete war Toms Antwort: "thekorn: give me a break, you're more like 11 :)". Für jene die Markus nicht kennen: er hat Python-Launchpad-Bugs geschrieben und hängt auch in python-launchpadlib drin. Super Session!

Der nächste war Ara Pulido, welche eine Session über QA-Tools (QA – Quality Assurance, Qualitätssicherung – d.Red.) im Gepäck hatte. Der BugSquad und das QA-Team haben eine super Arbeit geleistet, Tools zusammenzusuchen, welche das Leben sehr erleichtern, wenn man sich mit Bugs und QA beschäftigt. Ara verstand es gut, alles zu erklären, und war sehr bescheiden, als sie den Applaus mit den Worten »OK, genug, ich bin nicht Madonna« stoppte.

Leonard Richardson sprach in seiner Session über die großartige Launchpad Web Service API. Ebenso standen die Pläne des Projekts und wie es funktioniert auf der Agenda zusammen mit einigen interessanten Fragen. Gut gemacht Launchpad Team!

Anschließend erklärte das Security Team wie sie High-Quality Updates ausliefern. Marc, Kees und Jamie sprachen über ihren Workflow, was am wichtigsten ist wenn man Sicherheitsupdates macht und über die zahlreichen Initiativen, die wir haben, um Ubuntu sicherer zu machen. Macht weiter so, ihr macht das super!

Die letzte Session des zweiten Tages musste leider abgesagt werden. Martin Pitt sagte selbst: »Ich hab gerade etwas sehr dummes gemacht – Ich habe meinen Vortrag gelöscht! Es tut mir sehr leid, aber wir müssen leider abbrechen und es auf einen späteren Zeitpunkt verschieben«. Die Ankündigung, wann die Session nachgeholt wird, soll demnächst in seinem Blog [4] erscheinen.

Tag 3

Sebastian Bacher war der erste an diesem Tag und sprach über die Auslieferung eines GNOME-Releases und Millionen von Benutzern. Daniel Holbach war besonders aufgeregt dieser Session bei-zuwohnen, weil er für viele Monate Teil des ganzen war und sehr viel von Sebastian, Mr. Sebuild, gelernt hatte. Das Desktop Team ist sehr einladend und verrichtet tolle Arbeit. Nach dem Motto: »Es gibt immer etwas zu tun und das mit Spaß. Join in today!«

Daniel Holbach hatte noch einen weiteren Auftritt und erklärte, wie man Bugs in Ubuntu bereinigt. Er genoss die Session sehr, und sein Gefühl sagte ihm, dass es dabei nicht nur ihm so ging. Es wurde Harvest benutzt, um einfache Möglichkeiten zu finden, alle davon aus der Kategorie »resolved-Ups-tream«. Für Daniel ist es wichtig zu zeigen, dass Bugs zu bereinigen nicht sehr schwer ist. Wenn es dir Spaß macht, deine detektivischen Fähigkeiten täglich zu steigern, Sachen zum Laufen zu bringen, wenn du keine Scheu hast Fragen zu stellen und vorsichtig genug bist um Dinge zu testen, dann bist du definitiv die richtige Person fürs Team.

Danach war James Westby an der Reihe. Er brachte den Interessierten bei, wie man Bazaar für Packaging einsetzt. Er schnappte sich einen Bug, über den Daniel Holbach kurz zuvor gesprochen hatte, und demonstrierte die schiere Macht, die Bazaar verleiht, und demonstrierte Features, welche die Arbeit extrem erleichtern. Einfach großartig!

Freunde von Kubuntu bekamen alles, was sie jemals wollten und sogar noch ein bisschen mehr in Jonathan Thomas' vorzüglicher »Kubuntu Bug Squishing Session«. Er erklärte sehr gut, wie man Bugs bereinigt, wo man nach Fixes suchen muss und wie alles zum Schluss im Packaging zusammenkommt. Das Kubuntu Team braucht dringend Leute; wenn ihr also Interesse habt, seit euch nicht zu schade zu fragen.

Nick Barcet machte den Abschluss von Tag 3 und sprach darüber, wie man VMBuilder einsetzt, um

eine Testumgebung zu erzeugen. Bedauerlicherweise wurde Søren Hansen krank, sodass Nick alles alleine machen musste, was allerdings der Qualität keinen Abbruch tat. Nick hat seine Aufgabe perfekt gelöst und erklärte, warum vmbuilder so einzigartig ist und wie man sich bestens mit dem Development-Release spielen kann, ohne Angst haben zu müssen, dass sein »Echtssystem« in die Luft fliegt. Fantastisch!

Tag 4

Ara Pulido eröffnete mit einer Session über automatisiertes Desktop Testing. Sie verriet uns ein Geheimnis, eine schlichte Wahrheit: »Software zu testen ist Spaß – nicht langweilig, nicht nervtötend, einfach lustig.« Ihr Vortrag war sehr interessant. Schaut es euch an wie es funktioniert und wie man es schafft, dass Desktop Software sogar noch mehr rockt!

Neil Patel, Bill Filler und Pete Goodal hielten danach eine Session über Netbook Remix Q&A. Sie versprach, aufregend zu sein, und das war sie auch. Man erfuhr, was an den Netbooks so besonders ist, was die Pläne für die Zukunft sind und wie die Dinge möglicherweise einmal aussehen werden. Zudem ging man darauf ein, wie die Software der Netbooks aufgebaut ist. Um es in einem Satz zu beschreiben: »Just keep it simple«.

Die Session von Scott James Remnant über Boot Performance war eine der Sessions, die für mich sehr interessant war. Scott erklärte unter anderem die verschiedenen Vorgehensweisen, um die Boot-performance von Release zu Release steigern zu können. Aber auch Fehler der Vergangenheit und

Herausforderungen für die Zukunft wurden keinesfalls verschwiegen. Auch das Programm bootchart und dessen Rolle in der Entwicklung wurde unter die Lupe genommen. Einfach eine großartige Session!

Harald Sitter und Steve Stalcup sprachen über »Kubuntu Ninjas in Unicorn mode«. Die Session begann mit einer Erklärung, die Daniel versprochen hatte: »Ninjas sind magische, blauköpfige Affen mit Fledermausflügel und einem Horn, welches wie ein Zahnrad auf ihrer Stirn prangt und sprechen hauptsächlich »jibberish«, sodass die anderen Ubuntu-Entwickler sie nicht verstehen.« Glücklicherweise war die Session danach verständlicher und hat gezeigt, wie man ins Packaging von Kubuntu einsteigen kann.

Mirco Bauer und Jo Shield, gute Freund aus dem Mono Team berichteten anschließend, wie man Software für Mono paketierte. Auch diese Session war sehr informativ und zeigte viel gute Arbeiten, die Mono Software in Ubuntu und Debian integrieren.

Tag 5:

Lars Wirzenius zeigte, wie man sein .deb mit piuparts testet. Es war eine lustige Session, in der wichtige Features von piuparts besprochen wurden. Man kann es zum Beispiel als »Health-Check« für die Instabilität und Upgrade-Möglichkeit einsetzen. Danke Lars!

Michael Vogt brachte uns näher wie man mit python-apt seinen Spaß haben kann. Er erklärte die beweglichen Teile von apt und wie man sie gezielt einsetzen kann. Die gewählten Beispiele waren

leicht zu verstehen und vielleicht kommt ja der eine oder andere demnächst mit einem Script, welches unser aller Leben vereinfacht.

Anschließend waren Martin Albisetti und Paul Hummer an der Reihe und sprachen über Bazaar und Launchpad und wie man es verwendet. Es war eine großartige Session und zeigte hervorragend, warum die Kombination aus Bazaar und Launchpad so einmalig ist: Leicht zu benutzen, steht einem nicht im Weg und macht die ganze Arbeit. Fantastisch!

Ben Collins, eine Kernel Hacker der Sonderklasse, brachte uns bei, wie man Kernel-Module mit DKMS paketierte. Wie ein echter Kernel Hacker verstand er es, die Sachen kurz und bündig zu erklären. Wenn ihr Interesse habt im Kernel Team zu helfen, tut euch keinen Zwang an: Die Jungs beißen nicht!

Zum Abschluss brachten Cody Somerville und Charlie den Interessierten Xfce, welches in Xubuntu zum Einsatz kommt, näher. Hauptthema waren die Änderungen in der Version Xfce 4.6 und wie die Xubuntu-QA arbeitet. Schaut so aus als wird Xubuntu 9.04 exzellent!

Für all jene, die noch immer nicht genug haben, habe ich hier noch ein paar Informationen und Quellen die sicher den einen oder anderen Blick wert sind:

Solltet ihr diesmal die Ubuntu Developer Week verpasst haben, gibt es einerseits die Logs, andererseits habt ihr vielleicht Zeit, wenn die Session von Martin Pitt nachgeholt wird (Termin steht noch

nicht fest), und laut Daniel Holbach war das sicher nicht die letzte UDW. Für die nächste Entwicklungsphase nach dem Release von Jaunty wird es höchstwahrscheinlich wieder eine UDW geben.

Sicher einer der besten Orte um in die Entwicklung von Ubuntu einzusteigen ist die »Getting Started«-Wikiseite [5] beziehungsweise auch die Einstiegsseite des BugSquad [6].

Sollten Fragen auftreten, bieten sich die IRC-Channel auf Freenode an. Die wichtigsten sind #ubuntu-motu, #ubuntu-bugs, #ubuntu-translation.

Des weiteren sollte man den Blog von Daniel Holbach [7] und Ikhaya [8] im Auge behalten, da hier oft Ankündigungen von diversen Events wie Hugh Day, Bug Jams, etc. nachzulesen sind.

Christoph Schmied

schmiedc@yalmagazine.org

Informationen

[7] <https://wiki.ubuntu.com/Classroom>

[8] <https://lists.ubuntu.com/archives/ubuntu-news-team/2009-January/000382.html>

[9] <https://wiki.ubuntu.com/UbuntuDeveloperWeek>

[10] <http://martinpitt.wordpress.com/>

[11] <https://wiki.ubuntu.com/MOTU/GettingStarted>

[12] <https://wiki.ubuntu.com/BugSquad/GettingInvolved>

[13] <http://daniel.holba.ch/blog/>

[14] <http://ikhaya.ubuntuusers.de/>

Partitionierung mit fdisk

Klein, praktisch, schnell. Für alle, die mit fdisk mehr als *fdisk -l* anfangen wollen, haben wir uns an die Partitionierung und Formatierung in der Konsole gewagt.

Streikt der X-Server, oder man hat nur ein Terminal zur Verfügung, ist man auf Konsolenprogramme angewiesen. Umso wichtiger ist es, in diesen Situationen mit der Kommandozeile umgehen zu können. Diesmal probieren wir das Partitionieren mit *fdisk*.

Primäre, erweiterte und logische Partition

Informationen über Partitionen werden normalerweise in der Partitionstabelle im MBR (Master-Boot-Record) der jeweiligen Festplatte gespeichert [1]. Da in dieser aber nur Platz für 4 Partitionen ist und der Rahmen heutzutage leicht gesprengt wird, musste man sich einen Trick einfallen lassen.

Die Lösung war die Einführung einer primären »erweiterten« Partition. Diese Partition kann nun theoretisch beliebig viele logische Partitionen enthalten, wird jedoch vom Linux Kernel bei S-ATA Festplatten auf 15 beschränkt. Die Partitionsnummern von logischen Partitionen starten, anders als bei primären, bei 5.

Für das Partitionieren hat das nun folgende Auswirkungen: Will man mehr als 4 Partitionen haben, muss man nach der dritten Primären eine erweiterte Partition erstellen.

Partition	Größe
/	15-20 GB; bei Platzmangel minimal 10 GB
/home	der Rest
Swap	bei aktuellen Desktop-Systemen reichen 512 MB; Falls man Suspend-To-RAM nutzen will, sollte sie gleich groß sein wie der RAM <i>Hinweis:</i> Die Faustregel »doppelt soviel Swap-Speicher wie RAM« gilt nur bei alten Kernen!
/boot	150 MB; bei einem eigenen Kernel ohne initramfs reichen aber auch schon 32 MB <i>(separate Partition wird nur benötigt, falls das Root-Dateisystem verschlüsselt wird)</i>

Tabelle 1: empfohlene Partitionsgrößen

Hinweis für Windows Dualboot: Windows lässt sich nur auf einer primären Partition installieren!

Partitionen

Für den normalen Desktopbenutzer ist eine Aufteilung in drei Partitionen (*/*, */home*, *Swap*) interessant; soll das Root-Dateisystem verschlüsselt werden, muss noch eine extra */boot*-Partition angelegt werden.

Wenn man eine */boot*-Partition verwendet, sollte man darauf achten, dass in der *menu.lst* auf die */boot* Partition gerootet wird und der Kernel und das *initramfs* sich daher auf der obersten Ebene befinden.

Beispiel für */boot* auf *sda1*:

```
title Linux 2.6.27-r8
root (hd0,0)
kernel /bzImage root=/dev/sda2 udev
```

Da GRUB auch erweiterte Partitionen booten kann, ist es egal, ob die Partitionen auf primären oder logischen Partitionen angelegt werden.

Partitionieren

Alle folgenden Befehle müssen mit Root-Rechten durchgeführt werden. Ubuntu-Nutzer stellen ein »*sudo*« vor jeden Befehl.

Vor einer Partitionierung ist es generell sinnvoll, sich die bisherige Partitionsstruktur einmal anzusehen. Hilfreiche Befehle sind da z. B.:

```
fdisk -l
```

oder

```
df -h
```

Um die Partitionierung schließlich zu starten geben wir Folgendes ein:

```
fdisk /dev/sdx
```

X steht dabei für die Festplattennummer. Die erste Festplatte wäre somit sda. IDE Festplatten werden alternativ mit »hdx« angesprochen.

Nun befindet man sich in der fdisk-Konsole, welche mit eigenen Befehlen gesteuert wird.

Eine Übersicht über die Befehle erlangt man durch die Eingabe von »m«. Die Wichtigsten sind in Tabelle 2 kurz aufgelistet:

Will man eine Partition löschen, gibt man »d« ein. Sind mehrere Partitionen vorhanden, müssen wir zuerst die Partitionsnummer angeben. Gibt es je-

doch nur eine Möglichkeit, wird die Partition automatisch selektiert und gelöscht.

Für das Erstellen einer neuen Partition benutzt man »n«. Darauf folgt die Entscheidung zwischen einer erweiterten/primären/logischen Partition und deren Partitionsnummer. Nun fehlt nur noch das Setzen der Größe. Beim ersten Zylinder nimmt man durch Druck auf »Enter« den Standardwert. Beim letzten Zylinder geben wir dann die gewünschte Größe ein. Für eine 150 MB Partition gibt man beispielsweise

```
+150M
```

ein, für eine 150GB Partition

```
+150G
```

Ab und an ist es fein zu schauen, wie weit man schon ist. Eine Übersicht über den Status der Par-

titionstabelle kann man sich mit »p« verschaffen (Abb. 1).

Sollte man einen anderen Dateisystemtyp als ext2/3/4 auf der Partition installieren wollen, muss man ihn mit »t« setzen. Mit »L« kann man sich eine Übersicht über die einzugebenden Dateisystemtypen ausgeben lassen (Abb. 2). Für uns sind vor allem die Hexcodes b (FAT32) und 82 (Swap) interessant.

Das Setzen eines Bootflags ist mittlerweile unnötig geworden. Sollte es aber beim Booten Probleme geben, die nicht mit Grub-Fehlern [2] zusammen-

```
Befehl (m für Hilfe): p
```

```
Platte /dev/sda: 200.0 GByte, 200049647616 Byte
255 Köpfe, 63 Sektoren/Spuren, 24321 Zylinder
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes
Disk identifiziert: 0x03e9144c
```

Gerät	boot.	Anfang	Ende	Blöcke	Id	System
/dev/sda1	*	1	5	40131	83	Linux
/dev/sda2		6	1830	14659312+	83	Linux
/dev/sda3		1831	2323	3960022+	82	Linux Swap / Solaris
/dev/sda4		2324	24321	176698935	83	Linux

Statusübersicht mit dem Befehl »p«

Befehl	Auswirkung
p	gibt eine Übersicht über die Partitionen aus
d	löscht eine Partition
n	erstellt eine neue Partition
t	setzen des Dateisystemtyps
a	setzen der Bootflag
m	zeigt die Hilfe an
w	schließt die Partitionierung ab und speichert die Partitionstabelle
q	Beenden der Partitionierung ohne zu speichern

Tabelle 2: Befehlsübersicht von fdisk

Befehl	Auswirkung
mkfs.ext2 /dev/sda1	erstellt ein ext2-Dateisystem auf sda1
mkfs.ext3 /dev/sda1	erstellt ein ext3-Dateisystem auf sda1
mkfs.ext4 /dev/sda1	erstellt ein ext4-Dateisystem auf sda1
mkfs.vfat -F 32 /dev/sda1	erstellt ein FAT32-Dateisystem auf sda1; benötigt das Paket »dosfstools«
mkswap /dev/sda1	erstellt ein Swap-Dateisystem auf sda1

Tabelle 3: Befehle für das Erstellen von Dateisystemen

hängen, kann es sicher nicht schaden, die Bootflag zu setzen.

Falls alle Einstellungen getätigt wurden, kann die Konfiguration nun mit »w« in die Partitionstabelle geschrieben werden.

Erstellen des Dateisystems

Nachdem partitioniert wurde, ist der Speicher noch nicht nutzbar. Er muss zuerst mit einem Dateisystem versehen werden. Die Partition muss dazu ausgehängt werden. Die bekanntesten Befehle dafür sind in Tabelle 3 aufgeführt.

Aus Gründen der Datenintegrität raten wir generell zu ext3-Dateisystemen. Nach dem Erstellen des Dateisystems kann die Partition wieder gemountet werden. Im Falle einer Swap-Partition muss man anstatt des normalen mount-Befehls

```
swapon /dev/sda1
```

verwenden.

0	Leer	1e	Verst. W95 FAT1	80	Old Minix	bf	Solaris
1	FAT12	24	NEC DOS	81	Minix / old Lin	c1	DRDOS/sec (FAT-
2	XENIX root	39	Plan 9	82	Linux Swap / So	c4	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	83	Linux	c6	DRDOS/sec (FAT-
4	FAT16 <32M	40	Venix 80286	84	OS/2 verst. C:-	c7	Syrinx
5	Erweiterte	41	PPC PReP Boot	85	Linux erweitert	da	Nicht-DS-Daten
6	FAT16	42	SFS	86	NTFS volume set	db	CP/M / CTOS /
7	HPFS/NTFS	4d	QNX4.x	87	NTFS volume set	de	Dell Utility
8	ATX	4e	QNX4.x 2nd part	88	Linux Plaintext	df	BootIt
9	AIX bootfähig	4f	QNX4.x 3rd part	8e	Linux LVM	e1	DOS access
a	OS/2 Bootmanage	50	OnTrack DM	93	Amoeba	e3	DOS R/O
b	W95 FAT32	51	OnTrack DM6 Aux	94	Amoeba BBT	e4	SpeedStor
c	W95 FAT32 (LBA)	52	CP/M	9f	BSD/OS	eb	BeOS fs
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a0	IBM Thinkpad hi	ee	GPT
f	W95 Erw. (LBA)	54	OnTrackDM6	a5	FreeBSD	ef	EFI (FAT-12/16/
10	OPUS	55	EZ-Drive	a6	OpenBSD	f0	Linux/PA-RISC b
11	Verst. FAT12	56	Golden Bow	a7	NeXTSTEP	f1	SpeedStor
12	Compaq Diagnost	5c	Priam Edisk	a8	Darwin UFS	f4	SpeedStor
14	Verst. FAT16 <3	61	SpeedStor	a9	NetBSD	f2	DOS sekundär
16	Verst. FAT16	63	GNU HURD / SysV	ab	Darwin boot	fb	VMware VMFS
17	Verst. HPFS/NTF	64	Novell Netware	b7	BSDI fs	fc	VMware VMKCORE
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fd	Linux raid auto
1b	Verst. W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard ver	fe	LANstep
1c	Verst. W95 FAT3	75	PC/IX	be	Solaris boot	ff	BBT

Hexabelle der Dateisystemtypen

Erstellen des Disklabels

Bei einer externen Festplatte oder einem USB-Stick ist es sinnvoll, ein Disklabel zu vergeben. Der Datenträger wird so bei eingeschaltetem Automount immer an einem fixen Mountpunkt gemountet. Nennen wir unseren Datenträger beispielsweise »Musik«, wird er beim Anschließen nach `/media/Musik` gemountet.

Auch hier darf die Partition beim Ausführen des Disklabel-Befehls nicht gemountet sein.

Bernhard Posselt
ray@yalmagazine.org

Informationen

- [1] <http://de.wikipedia.org/wiki/Partitionstabelle>
- [2] <http://www.gentoo.org/doc/de/grub-error-guide.xml>

The Hive Game

Das gab es noch nie! Exklusiv für das Yalm-Magazin ist ein Computerspiel entstanden. Das Spiel »Hive« ist in Python geschrieben und verwendet Tkinter als Grafikbibliothek. Der Artikel beschreibt das Spiel, erklärt wo es bezogen werden kann und lädt zur Mitarbeit bei der Weiterentwicklung ein.

Willkommen beim Hive-Spiel

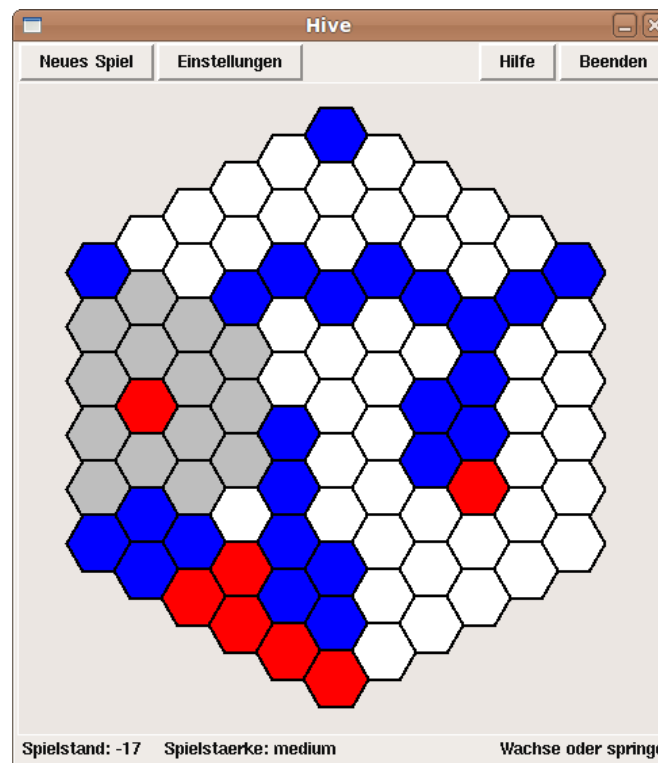
Wie auf dem Bild zu sehen ist, ähnelt das Spiel einer Bienenwabe, daher kommt sein Name. Man schlüpft in die Rolle eines Imkers, der sich um den Erhalt und die Gesundheit seines Bienenstocks kümmern darf. Unglücklicherweise geht zur Zeit ein gefährlicher Virus um, der die Waben im Bienenstock infiziert. Die Aufgabe des Imkers ist es, die Wabe zu impfen um die Infektion zu bekämpfen. Im nächsten Kapitel wird erklärt, wie man die Zellen der Wabe vor dem Virus schützt. Stellt man sich dabei schlaue an, kann die Wabe vor einer vollständigen Infektion bewahrt werden.

Spielregeln

Man spielt gegen den Computer. Der Spieler ist der Imker der die Zellen in der Wabe impft. Der Computer spielt die Rolle des Virus, der versucht, mehr und mehr der Zellen zu infizieren. Zu Beginn ist die Wabe fast leer außer einigen roten und blauen Zellen. Solange die Grundeinstellungen nicht verändert werden, hat der Spieler die roten Zellen und der Computer die blauen.

Der Spieler beginnt, indem er auf eine seiner (roten) Zellen klickt. Dann werden alle Nachbarn die-

ser Zelle grau markiert. Es gibt zwei Typen von Nachbarzellen, nahe und ferne Nachbarn. Die nahen Zellen befinden sich in unmittelbarer Nachbarschaft zu der ausgewählten (roten) Zelle. Die fer-



Hive in Aktion

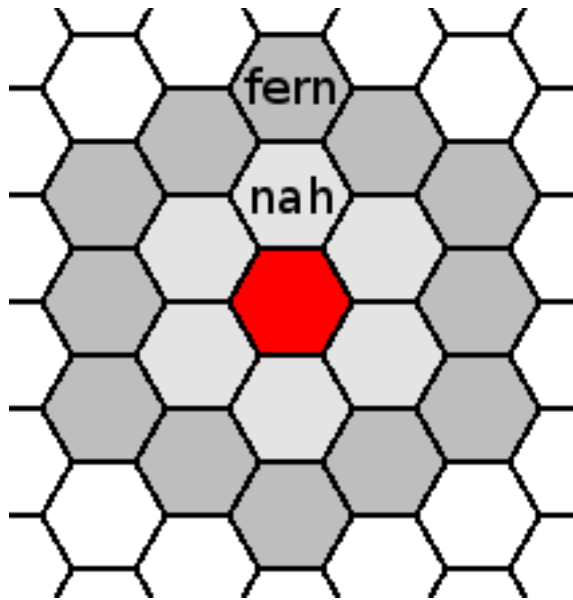
nen Nachbarn liegen weiter außen neben den nahen Nachbarn. Das kann man sich wie zwei Ringe um die Zellen herum vorstellen; den inneren Ring bilden die nahen Zellen und den äußeren Ring formen die fernen Zellen. Die Abbildung verdeutlicht das Konzept: die hellgrauen Zellen sind nahe Zellen, während die mittelgrauen ferne Nachbarn sind. Im Spiel werden beide Nachbartypen im gleichen Grau dargestellt.

Nachbarzellen können geimpft werden. Der Spieler kann entweder auf einen nahen oder einen fernen Nachbarn klicken. Es ist auch möglich, die zuvor gewählte Zelle (rot) durch einen Klick wieder abzuwählen um eine andere rote Zelle auszuwählen. Der Unterschied zwischen nahen und fernen Nachbarn liegt in der Art der Züge, die dorthin erlaubt sind. Wenn man einen nahen Nachbarn auswählt, dehnt sich das Impfserum zu dieser Zelle hin aus; die Anzahl der geimpften Zellen wächst. Abschließend sind zwei Zellen geimpft: die alte rote Zelle und der nahe Nachbar. Bei fernen Nachbarn springt die Impfung zu dieser Zelle über. Danach hat man die alte Zelle verloren und den fernen Nachbarn hinzu gewonnen. Am besten probiert man das aus; es ist viel einfacher als hier beschrieben. Wenn man zu einer Nachbarzelle wächst (nah) oder springt (fern), werden alle infizierten Zellen (blau) die die neue Zelle berühren geimpft. Sie ändern ihre Farbe von blau zu rot.

Hier noch einmal das Vorgehen in drei einfachen Schritten:

1. wähle eine eigene Zelle
2. wähle die Art des Zuges aus: wachsen oder springen
3. prüfe, welche infizierten Zellen nun geimpft sind (aus blau wird rot)

Jetzt macht der Computer seinen Zug und wendet dabei die gleichen Regeln an. Er wird versuchen weitere Zellen in der Wabe zu infizieren. Das ist auch mit bereits geimpften Zellen möglich. Er kann weiße und rote Zellen infizieren.



Nahe und ferne Nachbarn

Unten am Bildschirm (in der Statuszeile) gibt es zwei Hinweisbereiche. In der linken Ecke wird die Differenz zwischen geimpften (roten) und infizierten (blauen) Zellen angezeigt. Zum Beispiel: falls

es 20 blaue Zellen und 15 rote Zellen gibt, zeigt der Spielstand -5 an. Eine negative Zahl bedeutet, dass der Spieler im Rückstand ist. Im Beispiel ist der Spieler 5 Zellen hinterher. In der rechten Ecke erklären kurze Hinweise was als nächstes zu tun ist bzw. ob und wer das Spiel gewonnen hat.

Schritt für Schritt wird der Spieler gegen den Computer um Zellen kämpfen. Das Spiel endet, wenn einer der Spieler keinen gültigen Zug mehr machen kann. Dies ist auch dann möglich, wenn es noch leere Zellen in der Wabe gibt. Falls beide Spieler die gleiche Anzahl Zellen besitzen, endet das Spiel unentschieden.

Einstellungen

Am oberen Rand des Spiels gibt es einige Schalter. Die meisten davon sind selbsterklärend. Der Schalter »Einstellungen« öffnet einen Dialog in dem die Spielstärke und weitere Optionen ausgewählt werden können. Die Einstellungen werden gespeichert und sind damit auch bei einem Neustart des Programms noch gültig. Mit »Beenden« wird das Spiel verlassen.

Im Fenster mit den Einstellungen kann die Spielstärke verändert werden. Es gibt vier Stärken: adaptive, easy, medium und hard. Die Spielstärke »easy« ist für absolute Anfänger und für Kinder gedacht. Der Computer verfolgt dabei nicht das Ziel zu gewinnen, sondern macht Züge, die ihm einen größtmöglichen Freiheitsgrad (viele Zugmöglichkeiten) für den nächsten Zug geben. In der Spiel-



Dialog mit Einstellungen

stärke »medium« versucht der Computer das Spiel zu gewinnen, indem er möglichst viele Zellen infiziert und dabei auch den Freiheitsgrad berücksichtigt. In dieser Einstellung ist der Computer relativ leicht zu besiegen. Bei der Spielstärke »hard« benutzt der Computer drei Algorithmen, um das Spiel für sich zu entscheiden. Er beachtet den Freiheitsgrad, er maximiert die infizierten Zellen und er hält seine Zellen nahe beieinander. Jetzt ist es nicht mehr so einfach den Computer zu schlagen (der Autor hat es bisher noch nie geschafft). Als weitere Spielstärke gibt es »adaptive«. Hierbei ändert sich die Spielstärke in Abhängigkeit vom Punktestand. Wenn der Computer denkt, er würde verlieren, schaltet er eine oder zwei Spielstärken höher und umgekehrt.

Außerdem kann eingestellt werden, welcher Spieler den ersten Zug macht.

Die Hilfe zum Spiel (Schaltfläche »Hilfe«) öffnet eine HTML-Datei in einem Web Browser. Im Feld »Web Browser« kann der Name des Web Browsers angegeben werden. Dies ist nur auf Linux Systemen notwendig; auf Windows Computern wird der Browser gestartet, der mit der Dateiendung *.html verknüpft ist. Die Einstellung im Feld »Web Browser« wird bei Windows Systemen ignoriert. Bei Linux führt das Spiel, nachdem auf die Hilfe-Schaltfläche geklickt wurde, einen Kommandozeilenbefehl aus, z. B.:

```
firefox help_de.htm
```

Der Name »firefox« wird dabei aus dem Feld »Web Browser« genommen. Wer unter Linux einen anderen Web Browser verwendet, kann in dieses Feld den entsprechenden Eintrag machen. Vorheriges Ausprobieren auf der Kommandozeile hilft, den richtigen Befehl zu finden.

Das Hive-Spiel unterstützt Mehrsprachigkeit. Beim Starten des Programms wird das Betriebssystem nach seiner Länderkennung gefragt. Falls diese »de« für den deutschsprachigen Raum ist, verwendet das Spiel die deutschen Spracheinstellungen. Bei »en« (für Englisch) werden englische Einstellungen für die Texte und die Hilfe verwendet. Bei einer anderen Länderkennung schaltet das Spiel auf englische Einstellungen um. Wem das nicht gefällt, kann dies leicht ändern. Die Datei *hive_language.cfg* enthält alle Spracheinstellungen. Dort können beliebige weitere Sprachen hinzugefügt werden. Für Französisch muss die erste

Zeile in der Datei so aussehen: »name|de|en|fr«. In allen weiteren Zeilen fügt man nun nach dem trennenden Strich »|« die französische Übersetzung ein. Das Spiel merkt nun, dass es in der Sprachdatei auch französische Daten gibt.

Installation

Das Spiel Hive wird bei Launchpad [1] verwaltet. Von dort kann die aktuelle Version heruntergeladen werden. Die ZIP-Datei muss nun in ein beliebiges Verzeichnis entpackt werden. In diesem Verzeichnis befindet sich die Python Datei »hive.py«. Um diese Datei starten zu können, muss vorher Python 2.x installiert werden. Auf Linux-Systemen ist Python meistens bereits vorhanden. Unter Windows kann die Datei *hive.py* durch einen Doppelklick direkt gestartet werden, da die Endung *.py bei der Installation von Python mit eben diesem verknüpft wird. Unter Linux kann entweder in einer Konsole zum Hive Verzeichnis navigiert werden, um dort mittels *python hive.py* das Spiel zu starten. Als Alternative bietet sich ein Startknopf (Starter) an. Dazu gibt es im Hive Verzeichnis bereits ein Shell Script mit dem Namen *hive.sh*. Nachdem man darin den Pfad zum Hive-Verzeichnis korrigiert hat, kann ein Starter dieses Shell-Script aufrufen und damit das Spiel starten. *Achtung:* das Shell-Script muss ausführbar gemacht werden. Im Verzeichnis gibt es auch eine Icon-Datei *hive64_icon.png*, die für den Starter verwendet werden kann.

Zusammenarbeit

Ursprünglich war geplant, das Spiel bei SourceForge für die Zusammenarbeit mit der Community bereit zu stellen. Da der Autor dort aber »vor lauter Bäumen den Wald nicht mehr gesehen hat«, wurde das einfachere und übersichtlichere Kollaborationsportal Launchpad von Canonical für das Projekt-Hosting ausgewählt. Dort gibt es die Projektübersicht, der Source-Code ist verfügbar, Bugs können gemeldet werden, Anforderungen zur Weiterentwicklung (so genannte Blueprints) werden dort verwaltet und vieles mehr.

Genug der vielen Worte – now it's time to play.

Ralf Hersel
rhersel@yalmagazine.org

Informationen

[1] Hive Game im Launchpad:
<https://launchpad.net/hivegame>

Heimnetzwerk aufbauen – The Easy Way (1)

Da hat man einen PC und ist glücklich. Irgendwann kommt aber der Wunsch nach einem zweiten PC oder einen Laptop/Netbook und der Gedanke zur Vernetzung der beiden auf. Sei es um vom 2.PC aus zu drucken oder die Bilder des letzten Urlaubs auch auf der heimischen Grillfete vorzeigen zu können. Den einfachen Aufbau eines Heimnetzwerks soll dieser Artikel veranschaulichen.

Voraussetzung für ein funktionierendes Heimnetzwerk ist das Vorhandensein von mindestens zwei PCs mit Netzwerkanschluss, obwohl man strenggenommen auch nur mit einem PC ein Netzwerk de facto einrichten kann. Ob die beiden nun Laptops oder Desktop-PCs sind ist unerheblich.

Basiswissen

Klassischerweise besteht ein Netzwerk aus mindestens zwei Komponenten, die über eine Netzwerkschnittstelle miteinander kommunizieren.

Zur Übertragung der Daten wird häufig das TCP/IP Netzwerkprotokoll genutzt. In diesem Netzwerk identifizieren sich die Rechner über eine IP-Adresse, sie besteht aus einer Zahlenfolge, die dem PC eine weltweit einzigartige Adresse zuteilt – zumindest wenn man mit dem Internet verbunden ist. So hat jeder Server und jeder Client, der mit dem Internet verbunden ist, eine einmalige IP-Adresse [1].

Im Heimnetzwerk sieht das etwas anders aus. Zwar sind auch hier alle angeschlossenen PCs mit einer IP ausgestattet, nur ist diese nicht weltweit einmalig. Für die internen IP-Adressen im Heim-

netzwerk ist der Adressraum 192.168.0.0 bis 192.168.255.255 nach RFC 3330 reserviert [1].

Wer die IP-Adresse seines Rechners wissen möchte, der gibt in der Konsole einfach:

```
ifconfig <Schnittstelle>
```

ein. *<Schnittstelle>* steht hierbei für die Netzwerkschnittstelle, über die der Rechner kommuniziert. Dabei kann es sich zum Beispiel um Netzwerkkarten für Ethernet oder Wireless-LAN-Anschlüsse handeln; sie erhalten zumeist Bezeichnungen in der Form: »eth0« bzw. »wlan0«. Es besteht jedoch auch die Möglichkeit, dass Schnittstellen existieren, die keiner Hardware zugeordnet sind. Praktische Vertreter hierfür wären das Loopback-Device [2] »lo« oder Virtualisierungsschnittstellen wie »vmnet0«.

DHCP

Die Zuweisung der IP-Adresse kann manuell konfiguriert werden oder aber dynamisch über einen DHCP-Server erfolgen. DHCP steht für »Dynamic Host Configuration Protocol«, durch dieses Protokoll werden die IP-Adressen automatisch an die

angeschlossenen PCs zugewiesen. Doch nicht nur die IP, sondern alle dazugehörigen Netzwerkeinstellungen werden durch DHCP automatisch übermittelt. Im Normalfall reicht es aus, wenn man ein Netzkabel mit dem PC verbindet, da der Empfang von DHCP-Daten bei vielen PCs standardmäßig als erste Wahl bei Netzwerkverbindungen versucht wird.

Der DHCP-Client übermittelt hierbei dem Server seine MAC-Adresse [3], der Server weist dem Client anschließend je nach Konfiguration die IP-Adresse manuell, automatisch oder dynamisch zu. Die MAC-Adresse ist ähnlich der IP-Adresse weltweit einmalig. Nur ist hier der Unterschied, dass jeder Netzwerkcontroller mit einer einzigartigen MAC-Adresse ausgestattet ist.

Der DHCP-Server merkt sich einfach die MAC-Adresse des PCs bzw. des Netzwerkcontrollers mit der vergebenen IP. Wer die MAC-Adresse seiner Netzwerkschnittstelle wissen möchte, der gibt in der Konsole einfach

```
ifconfig
```

ein. Hinter dem Begriff »Hardware Adresse« steht dann die MAC-Adresse der Schnittstelle. Wer hier aufmerksam ist, der wird feststellen, dass jede Netzwerkschnittstelle auf dem PC eine andere MAC-Adresse hat.

Ausgestattet mit diesem Basiswissen kann man sich nun die Konfiguration seines DHCP-Servers (in der Regel ist das der Router) genauer ansehen.

Dazu gibt man im Browser die IP des Routers ein. Normalerweise lautet sie 192.168.0.1. Benutzer einer Fritz!Box geben in die Adresszeile des Browsers *fritz.box* ein. Um die IP des Routers herauszubekommen, gibt es 2 Wege. Der erste ist der Blick ins Handbuch des jeweiligen Routers. Der zweite Weg ist der Aufruf des Befehls

```
route
```

als Root in der Konsole. In der Zeile, die mit *default* anfängt, sieht man in der 2. Spalte die IP des Routers und ganz am Schluss in der Spalte *iface* sieht man die Netzwerkschnittstelle, für die die Einstellung gilt. In den Optionen des DHCP-

Server Dienstes kann man dann die Einstellung für »statisches DHCP« auswählen. So stellt man sicher, dass immer die gleiche IP für den Rechner reserviert bleibt. Das funktioniert aber nur, wenn der PC bereits erfolgreich mit dem Router kommuniziert.

Gemeinsam nutzen

Um die weiteren Schritte nachzuvollziehen, gehe man von folgender Konfiguration aus:

```
PC 1 IP-Adresse: 192.168.0.100
```

```
PC 2 IP-Adresse: 192.168.0.101
```

```
Router IP-Adresse: 192.168.0.1
```

Um zu schauen, ob die beiden PCs sich im Netzwerk »sehen« können, gibt man auf der Konsole folgenden Befehl ein (als Root):

```
nmap -sP 192.168.0.1-255
```

Bei wem das Paket *nmap* nicht standardmäßig installiert ist, der sollte sich das über seinen Paketmanager herunterladen.

Der Befehl scannt die Netzwerkrange von 192.168.0.1 bis 192.168.0.255. Für einfache Netzwerke reicht das aus.

Sollte alles normal abgelaufen sein, kann man den nächsten Schritt wagen. Wenn nicht, sollte man die Verkabelung prüfen.

```

Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mc - ~  jan@ubuntu-netbook: ~  root@ubuntu-netbook: /h...
Interrupt:18
lo      Link encap:Lokale Schleife
        inet Adresse:127.0.0.1  Maske:255.0.0.0
        inet6-Adresse: ::1/128  Gültigkeitsbereich:Maschine
        UP LOOPBACK RUNNING  MTU:16436  Metrik:1
        RX packets:7066 errors:0 dropped:0 overruns:0 frame:0
        TX packets:7066 errors:0 dropped:0 overruns:0 carrier:0
        Kollisionen:0 Sendewarteschlangenlänge:0
        RX bytes:4643469 (4.6 MB)  TX bytes:4643469 (4.6 MB)

jan@ubuntu-netbook:~$ iwconfig -a
-a      No such device

jan@ubuntu-netbook:~$ ifconfig eth1
eth1    Link encap:Ethernet  Hardware Adresse 00:21:00:7a:e1:c0
        inet Adresse:192.168.0.107  Bcast:192.168.0.255  Maske:255.255.255.0
        inet6-Adresse: fe80::221:ff:fe7a:e1c0/64  Gültigkeitsbereich:Verbindung
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
        RX packets:647349 errors:0 dropped:0 overruns:0 frame:451863
        TX packets:586965 errors:6 dropped:0 overruns:0 carrier:0
        Kollisionen:0 Sendewarteschlangenlänge:1000
        RX bytes:401027771 (401.0 MB)  TX bytes:82146825 (82.1 MB)
Interrupt:18

jan@ubuntu-netbook:~$

```

Wertvolle Informationen liefert der Befehl *ifconfig*

Spätestens jetzt stellt sich die Frage, wie man die Daten des jeweils anderen Rechners nutzen kann.

Dazu gibt es viele Wege. In dieser Ausgabe soll es um die Verbindung via NFS und die gemeinsame Druckerverwendung gehen.

NFS – Das Netzwerk-Dateisystem

Der größte Vorteil dieses Protokolls [5] liegt darin, dass die Dateien nicht wie bei FTP übertragen werden müssen um sie zu nutzen, sondern die Dateien so behandelt werden, als wenn sie auf dem lokalen Rechner liegen würden. Bevor man dies

aber nutzen kann, gilt es einige Vorbereitungen zu treffen.

Auf jedem Rechner des Netzwerks, der per NFS auf Daten zugreifen oder diese bereitstellen soll, muss das Paket *nfs-common* installiert sein. Soll ein Rechner nur auf NFS-Daten zugreifen, so reicht die Installation dieses Paketes aus. Soll ein Rechner allerdings auch Daten bereitstellen, also als Server dienen, dann muss das Paket *nfs-kernel-server* zwingend installiert werden. Die beiden Paketnamen stammen aus den Repositories von Ubuntu und können namentlich bei anderen Distributionen abweichen. Im Paketmanager ist aber stets erklärt, welche Funktionen ein Paket bereitstellt.

Angenommen, PC 1 soll der Server und PC 2 der Client sein, so sind auf dem ersten PC beide Pakete zwingend zu installieren.

Mit dem Befehl:

```
exportfs
```

kann man auf PC 1 sehen, welche Verzeichnisse per NFS im Netzwerk verteilt werden. Standardmäßig steht in der Ausgabe des Befehls nichts. Um Verzeichnisse per NFS bereitzustellen, ist Handarbeit gefragt.

Permanente Freigaben

In einer Konsole auf PC 1 gibt man den Befehl:

```
nano /etc/exports
```

```

Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mc ~ jan@ubuntu-netbook: ~ root@ubuntu-netbook: /h...

jan@ubuntu-netbook:~$ ifconfig eth1
eth1      Link encap:Ethernet  Hardware Adresse 00:21:00:7a:e1:c0
          inet Adresse:192.168.0.107  Bcast:192.168.0.255  Maske:255.255.255.0
          inet6-Adresse: fe80::221:ff:fe7a:e1c0/64  Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
          RX packets:647349 errors:0 dropped:0 overruns:0 frame:451863
          TX packets:586965 errors:6 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:401027771 (401.0 MB)  TX bytes:82146825 (82.1 MB)
          Interrupt:18

jan@ubuntu-netbook:~$ nm
nm                nmblookup          nm-tool
nmap               nm-connection-editor
nm-applet          nm-system-settings
jan@ubuntu-netbook:~$ nmap -sP 192.168.0.1-255

Starting Nmap 4.62 ( http://nmap.org ) at 2009-01-27 10:42 CET
Host 192.168.0.1 appears to be up.
Host 192.168.0.100 appears to be up.
Host 192.168.0.101 appears to be up.
Host 192.168.0.107 appears to be up.
Host 192.168.0.151 appears to be up.
Nmap done: 255 IP addresses (5 hosts up) scanned in 32.978 seconds
jan@ubuntu-netbook:~$

```

Erreichbare Rechner im Netzwerk

als Root ein. In der Datei stehen bereits einige Beispiele für die Einträge. Generell gilt hier folgende Syntax:

```
/pfad_der_Freigabe IP_des_Clients →  
(verschiedene Optionen für  
Dateibehandlung)
```

Wer das Ganze lieber mit gedit oder einem anderen Editor machen möchte, sollte beachten, dass beim Programmaufruf entweder ein *gksu* bei GNOME oder ein *kdesu* bei KDE vorangestellt wird. Das bewirkt, dass der Editor oder das Programm mit Root Rechten ausgeführt wird. Denn nur Root darf innerhalb von */etc* schreiben.

Angenommen man möchte, dass PC 2 vollen Zugriff auf das */home*-Verzeichnis von PC 1 bekommen soll, so lautet der Eintrag in die Datei */etc/exports* wie folgt:

```
/home 192.168.0.101(rw, sync)
```

Das Leerzeichen muss mit der [TAB] Taste gesetzt werden. Der Eintrag bedeutet nichts anderes als dass das */home*-Verzeichnis von PC 1 mit vollem Schreibzugriff (*rw*) von PC 2 (192.168.0.101) genutzt werden kann. Die Option *sync* sorgt dafür, dass ein ständiger Abgleich des Verzeichnisses gemacht wird und so Änderungen an Dateien automatisch weitergeleitet werden und damit das Verzeichnis per NFS ständig aktuell ist. Mehr Optionen erhält man in der manpage zu *exports*:

```
man exports
```

Nachdem man die Datei */etc/exports* gespeichert hat, sollte man den NFS-kernel-Server neu starten. Das gelingt mit dem Befehl:

```
/etc/init.d/nfs-kernel-server restart
```

Die Datei *exports* wird nun neu eingelesen und ab jetzt kann man per NFS auf das */home*-Verzeichnis zugreifen.

Ob der Server funktioniert kann man mittels:

```
exportfs
```

auf der Konsole nachprüfen. Als Ausgabe erhält man nun die Aussage, dass das Verzeichnis */home* an die IP des PC 2 mit den Optionen *rw/sync* freigegeben ist.

Auf PC 2 sollte man nun nur noch ein Verzeichnis anlegen, in welches man die NFS Freigabe mounten kann. Als Beispiel kommt hier */media/pc1* zum Einsatz. Um schlussendlich die NFS-Freigabe auf PC 2 einzubinden, gibt man folgenden Befehl als Root in der Konsole ein:

```
mount -t nfs 192.168.0.100:/home →  
/media/pc1
```

Dies dauert einen kleinen Augenblick. Erfolgt keine Fehlermeldung, kann man unter */media/pc1* das */home*-Verzeichnis von PC 1 nutzen. Zu beachten ist hierbei allerdings, dass nur Root auf PC 2 Schreibrechte in */media/pc1* hat. Wer dies ändern möchte, sollte die Rechte für das Verzeichnis auf

den User umstellen. Empfehlenswert ist das allerdings nicht.

Wer nun aber die Freigabe auf mehr als nur einen PC erweitern möchte, der kann innerhalb der Datei */etc/exports* des Servers entweder jede IP einzeln angeben oder sogenannte Wildcards (Platzhalter) wie *»*** oder *»?** verwenden. Möchte man beispielsweise jedem PC im Netzwerk mit der IP-Range von *.100* bis *.109* eine Freigabe zur Verfügung stellen, trägt man als IP in *exports* folgendes ein:

```
192.168.0.10?
```

Möchte man die Freigabe auf das gesamte Netzwerk ausdehnen, so gibt man folgendes als IP ein:

```
192.168.0.0/24
```

Hiervon ist allerdings abzuraten, da dann jeder, der sich im Netzwerk befindet, diese NFS Freigabe mounten kann, egal ob dieser legal oder illegal im Netzwerk unterwegs ist.

Hat man alle Freigaben in die Datei *exports* eingetragen und den NFS-kernel-Server neu gestartet, möchte man eigentlich sicher gehen, dass auf den Client-PCs die Freigaben automatisch beim Booten eingehängt werden.

Dazu muss auf den Clients die Datei */etc/fstab* editiert werden. Ähnlich dem Eintrag in *exports* folgt die Syntax zwingend folgenden Regeln:


```
IP_des_Servers:/freigabe_des_servers →
lokaler_mountpoint dateisystem →
defaults 0 0
```

Anhand des Beispiels müsste der Eintrag bei PC 2 wie folgt lauten:

```
192.168.0.100:/home /media/pc1 →
nfs defaults 0 0
```

Die Leerstellen müssen zwingend per [TAB]-Taste gemacht werden und am Ende der Zeile muss man ENTER drücken, da sonst eine Fehlermeldung beim Booten die Folge wäre.

Um zu schauen, ob der Eintrag frei von Fehlern ist, wird zuerst die bisher auf `/media/pc1` eingehängte NFS-Freigabe per:

```
umount /media/pc1
```

ausgehängt. Ein anschließender Befehl:

```
mount -a
```

sollte nun automatisch die Freigabe unter `/media/pc1` einhängen. Ob dies funktioniert hat, sieht man entweder per

```
mount
```

auf der Konsole oder per

```
df -h
```

Bei dem `df`-Befehl ist der große Vorteil, dass man hiermit auch gleich den verfügbaren und benutzten Plattenplatz des jeweiligen mounts mitgeteilt be-

kommt. Mit diesem Wissen kann man nun nach Herzenslust NFS-Freigaben im Heimnetzwerk aufstellen.

Temporäre Freigaben

Wenn man nicht möchte, dass ein bestimmtes Verzeichnis oder eventuell eine ganze Festplatte per-

manent per NFS freigegeben wird, kann man das auch temporär machen.

Der große Vorteil von temporären Freigaben ist, dass sie maximal bis zum nächsten Reboot existieren und man keine Systemdateien editieren muss.

```
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
mc - ~
jan@ubuntu-netbook: ~
root@ubuntu-netbook: /h...
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
gvfs-fuse-daemon on /home/jan/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=jan)
192.168.0.100:/home on /media/test type nfs (rw,addr=192.168.0.100)
jan@ubuntu-netbook:~$ sudo umount /media/test/
jan@ubuntu-netbook:~$ sudo mkdir /media/pc1
jan@ubuntu-netbook:~$ sudo mount -t nfs 192.168.0.100:/home /media/pc1
jan@ubuntu-netbook:~$ mount
/dev/sda5 on / type ext3 (rw,relatime,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
/proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
varrun on /var/run type tmpfs (rw,nosuid,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
lrmm on /lib/modules/2.6.27-7-generic/volatile type tmpfs (rw,mode=755)
securityfs on /sys/kernel/security type securityfs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,noexec,nosuid,nodev)
gvfs-fuse-daemon on /home/jan/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=jan)
192.168.0.100:/home on /media/pc1 type nfs (rw,addr=192.168.0.100)
jan@ubuntu-netbook:~$
```

Erfolgreicher mount der NFS Freigabe

Dazu benötigt man lediglich den Befehl

```
exportfs host:/pfad
```

Als Host gibt man entweder die IP des Rechners an, der darauf zugreifen soll, oder man verwendet einen Platzhalter, um an mehrere IP zu verteilen. Der Pfad ist das Verzeichnis, welches man freigeben möchte. Wichtig hierbei ist, dass der Befehl als Root ausgeführt und der Doppelpunkt zwischen Host und Pfad mit angegeben ist. Sonst folgt eine Fehlermeldung. Mittels

```
exportfs
```

kann man nachprüfen, ob alles funktioniert hat, denn nun müsste das Verzeichnis und die IP des Hosts in der Ausgabe des Befehls stehen.

Sicherheit

Sofern man die Freigaben nur an einzelne IPs verteilt, ist man auf der sicheren Seite. Sollte man mit Platzhaltern arbeiten, so ist die Gefahr groß, dass Eindringlinge diese Freigaben mounten und so Datenbestände stark gefährden können.

Eindringlinge gelangen meist über nicht oder nur sehr rudimentär (WEP) verschlüsselte W-LAN-Verbindungen ins Netzwerk. Hier sollte man den allgemeinen Empfehlungen folgen und mindestens WPA einsetzen. Mehr zur Verschlüsselung von W-LAN Netzen gibt es z. B. bei Netzwerktotal.de [3]

Fazit NFS

Einfacher als mit NFS kann man seine Verzeichnisse im Netzwerk anderen nicht zugänglich machen. Natürlich leidet unter der Einfachheit die Sicherheit etwas. Ein weiterer Wermutstropfen ist, dass NFS-Freigaben nur mit Linux-Systemen gehandelt werden können. Sobald ein Windows PC mitmischen möchte, muss man auf SAMBA [6] ausweichen. Zwar gibt es unter Windows auch Software um NFS einzubinden, aber diese ist weder sicher noch bedienerfreundlich.

Drucken im Netzwerk mit CUPS

Das Common Unix Printing System (CUPS) [7] ist das Druckerverwaltungs- und -steuerprogramm unter Linux.

Hat man einen PC, hat man meist auch schon einen Drucker installiert und in Betrieb. Dies geschieht in aller Regel mittels diverser grafischer Programme. Unter GNOME dient dazu der Befehl

```
system-config-printer
```

und unter KDE der Befehl

```
cupsdconf
```

oder

```
kaddprinterwizard
```

Sollte man noch keinen Drucker eingerichtet haben, so ruft man das entsprechende grafische Tool

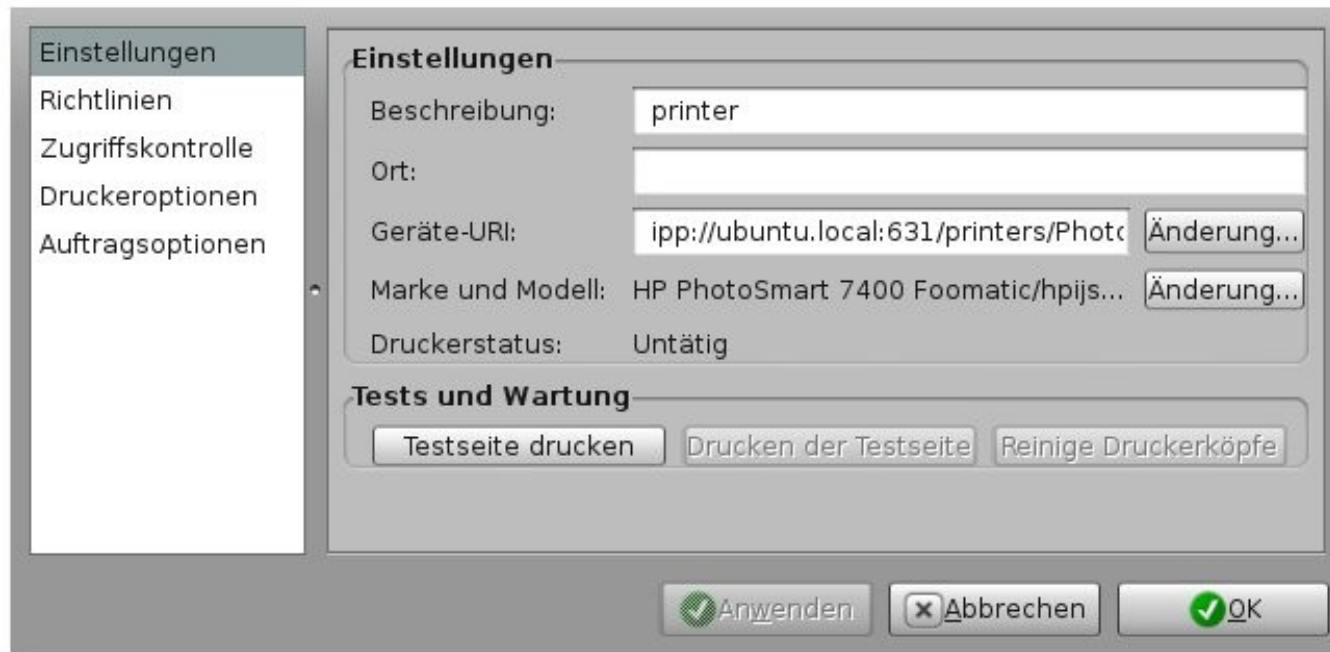
aus dem Menü von GNOME/KDE oder, sollte man einen anderen WindowManager einsetzen, per Konsole auf.

Meist erkennen die Tools automatisch einen angeschlossenen Drucker und stellen eine Auswahl an Druckertreibern zur Verfügung. Sollte dies nicht der Fall sein, so schaut man z. B. auf der Homepage des Druckerherstellers nach einem Druckertreiber für Linux. Doch auf den Herstellerseiten wird man meist enttäuscht. Besser ist es, z. B. auf Openprinting [8] nach einem passenden Treiber zu schauen. Die Treiber unter Linux tragen die Endung *ppd*.

Wird man auch dort nicht fündig, so gibt es mit Turboprint [9] eine gängige Alternative. Die dortigen Treiber gibt es in zwei Varianten: Einmal als kostenloser Treiber mit eingeschränktem Funktionsumfang oder als kostenpflichtigen Treiber mit voller Funktionalität. Gerade wer einen Drucker wie z. B. Lexmark sein eigen nennt, wird dort meist fündig.

Wieder zurück im grafischem Tool gibt man den Standort des Treibers an und installiert diesen. Ab sofort kann man unter Linux drucken was das Zeug hält.

Um das Ganze nun aber auch jedem PC im Netzwerk zu gestatten, ist etwas Handarbeit gefragt. Im Tool von KDE kann man das ganze auch grafisch



Druckereinrichtung unter GNOME

machen. Leider fehlt dem GNOME Tool diese Möglichkeit.

Zuerst öffnet man auf dem Rechner, an dem der Drucker angeschlossen ist, per Editor die Datei `cupsd.conf`. Dies findet man meist unterhalb von `/etc/cups`.

Man begibt sich in den Abschnitt `#Show shared Printers ...` und ändert dort den Eintrag mit der IP unter Verwendung eines Platzhalters von z. B. `192.168.0.100` in `192.168.0.*`

Dann verändert man den Wert von `Browsing` auf `on`. Das bewirkt, dass jeder PC, dessen IP mit `192.168.0.` anfängt, den Drucker quasi sehen kann. Doch noch gestattet CUPS keinen Zugriff darauf.

Der Abschnitt zwischen `<Location>` und `</Location>` regelt den Zugriff auf den CUPS Server. Auch hier ändert man den Wert von `Allow` auf die IP `192.168.0.*`

Wer den Druckerzugriff nur auf bestimmte PCs beschränken möchte, der kann statt des Platzhalters auch die genaue IP angeben.

Die restlichen Abschnitte innerhalb der Datei sollten unangetastet bleiben, da man sonst Sicherheitslöcher produziert. Die Passwortabfrage im Abschnitt `# Default authentication type` kann man beruhigt per `#` ausklammern.

Wer unbedingt die Admin-Pages des CUPS Servers von anderen Rechnern aus ansehen oder ändern möchte, sollte im Bereich `# Restrict access to the admin pages...` die entsprechenden Änderungen vornehmen. Davon wird aber abgeraten.

Ist dies getan, startet man den CUPS Deamon/Server per

```
/etc/init.d/cups restart
```

oder

```
cupsys restart
```

einfach neu.

Zurück an den Rechner, welcher per Netzwerk auf den Drucker zugreifen möchte. Da die Einstellungen, die man bisher vorgenommen hat, ein Broadcasting von CUPS auslösen, sollte man auf den PCs, die auf den Drucker via Netzwerk zugreifen möchten, den CUPS-Dienst ausschalten.

Sollte man das nicht machen, wissen die laufenden CUPS-Server nicht, wer wirklich Server und wer Client ist und welche Warteschlange bearbeitet werden soll.

Nun ruft man das grafische Tool zur Druckereinstellung auf und klickt beim GNOME-Tool auf *Aktualisieren*. Schon sollte man die verteilten Drucker des CUPS-Servers sehen. Um das Ganze zu testen, klickt man auf Eigenschaften des jeweiligen Druckers und schickt eine Testseite an den Drucker. Innerhalb weniger Sekunden sollte der Drucker reagieren und die Seite ausdrucken. Man könnte meinen, dass es das jetzt war. – Irrtum.

Um den Anwendungen auf dem Client mitzuteilen, dass das System über einen Drucker verfügt, erledigt man einfach folgende Schritte.

Man startet als Root auf dem Client wieder den *cupsd*-Dienst. Anschließend klickt man auf *Neuen Drucker hinzufügen* und gibt dort unter *Sonstiges* folgenden Standort für den Drucker an:

```
ipp://IP_des_Servers:631/printers
```

Die Port 631 ist der Standardport, auf welchem der CUPS Druckaufträge entgegennimmt. Ist das geschehen, muss man den Treiber für den Drucker angeben. Falls nicht schon geschehen kopiert man diesen einfach vom Server auf den Client via USB-Stick oder per NFS-Freigabe. Man kann ihn natürlich auch schnell von den entsprechenden Seiten herunterladen.

Hat man den Treiber angegeben, kann man dem Drucker noch einen lokalen Namen geben und angeben wo sich der Drucker befindet. Dies sind

aber nur optionale Angaben. Allerdings ist es sinnvoll, dem Drucker einen Namen zu geben.

Hat man das erledigt, sieht man nochmals alle Einstellungen in der Übersicht. Neben der Ortsangabe, die mit *ipp://* beginnt, befindet sich eine Schaltfläche *Änderung*, die man anklicken sollte. Im folgenden Dialog sieht man neben der sogenannten URI noch eine Schaltfläche *Suche Warteschlange*. Nachdem man diese angeklickt hat, erscheint ein Fenster mit allen verfügbaren Druckern auf dem Server. Man wählt den passenden aus und klickt auf *OK*.

Etwas weiter unten in dem Dialog ist noch eine Schaltfläche *Verifizieren*. Klickt man diese an, erscheint ein Dialog welcher einem sagt, dass man den Drucker benutzen darf. Meist wird bei der URI folgendes auffällig, nämlich dass die IP einem Namen gewichen ist, da jeder PC im Netzwerk einen Namen hat (siehe Konsole *Nutzer@Name-PC*). Dieser Name wird anstatt der direkten IP verwendet.

Jetzt sollte alles passen und man kann z. B. unter OpenOffice im Druckdialog den soeben installierten Drucker anwählen.

Fazit CUPS

Nach etlicher Handarbeit, die einem die grafischen Tools doch immer mehr abnehmen, kann man bequem in Netzwerk drucken. Da CUPS nicht besonders wählerisch ist, können auch Nutzer mit Win-

dows über einen CUPS-Server drucken. Der einzige Nachteil ist, dass zum Drucken der Server und der Drucker eingeschaltet sein müssen. Wer möchte, kann zur Konfiguration seines Druckers auch das Web-Interface von CUPS benutzen. Dies ist unter *localhost:631* erreichbar. Durch die grafischen Tools geht die Konfiguration aber immer einfacher von der Hand.

Jan Radecker

cyclame@yalmagazine.org

Informationen

- [1] <http://tools.ietf.org/html/rfc3330>
- [2] <http://de.wikipedia.org/wiki/Loopback>
- [3] <http://de.wikipedia.org/wiki/MAC-Adresse>
- [4] <http://de.wikipedia.org/wiki/IP-Adresse>
- [5] http://de.wikipedia.org/wiki/Network_File_System
- [6] <http://www.netzwerktotal.de/wlansicherheit.htm>
- [7] <http://cups.org/>
- [8] <http://www.linuxfoundation.org/en/OpenPrinting>
- [9] <http://turboprint.de/>

Verteilen von E-Mails im Netzwerk

Wer mehr als nur einen PC besitzt, wünscht sich oft, dass man an jedem Rechner im Netzwerk seine E-Mails abrufen kann. Hierfür bieten Anbieter wie GMX oder Google in der Regel alle derzeitigen Standard-Protokolle wie IMAP, POP3 oder auch SMTP an. Doch wie kann man die E-Mails aller Konten sammeln und sinnvoll im Netzwerk verteilen?

Zwei beliebte Protokolle zum Verwalten der Mailbox sind zum einen das IMAP- und zum anderen das POP3-Verfahren [1]. Im Wesentlichen unterscheiden sich die beiden Arten darin, dass bei IMAP-Verwendung die E-Mails auf dem Server bleiben und bei POP3 die E-Mail komplett heruntergeladen und lokal gespeichert wird – wobei auch das POP3 Protokoll einen Verbleib der E-Mails auf dem Server vorsieht. Dies ist aber in den meisten Fällen im jeweiligen E-Mail Programm deaktiviert.

Grundlagen

Wer sich schon einmal ohne laufendem X-Server in der Konsole angemeldet hat, dem ist sicherlich aufgefallen, dass da irgendwas von Mail steht.

Unter Linux, das ja ein Mehrbenutzersystem ist, kann man sich gegenseitig E-Mails schicken. Aber nicht nur das funktioniert. Einige Prozesse wie z. B. Cron senden ihre Fehlermeldungen an den User, unter dem der Prozess gestartet wurde. Auch im Heimnetzwerk kann man E-Mails zwischen den Rechnern austauschen. Deshalb ist es sinnvoll, jedem Rechner im Netzwerk einen eindeutigen Namen zuzuordnen. Zugrunde liegt

```

Datei Bearbeiten Ansicht Terminal Reiter
mc - ~
.fetchmailrc [-M--] 4 L:[ 3+21
#Server lund1.IMAP
server imap.lund1.de
proto imap
#Benutzer
user hans
pass geheim
#Optionen
to musterhans
#Server pop.lund1.de
server pop.lund1.de
proto pop3
#Benutzer
user yvonne
pass geheim
#Optionen
keep
1 Hilfe 2 Speich 3 Markier 4 Ersetze 5 Kopiere

```

.fetchmailrc mit mehreren Konten

dem ganzem ein sogenannter MTA (Mail-Transfer-Agent) [2] wie z. B. sendmail, exim oder Exchange.

So kann z. B. Benutzer A am Rechner »ubuntu« an den Benutzer B am Rechner »ubuntu2« eine E-Mail schicken. Dieses System kann man sich zu nutze machen um einen E-Mail Server für das Heimnetzwerk zu installieren.

Sammeln der E-Mails mit Fetchmail

Voraussetzung für den E-Mail-Server ist natürlich ein funktionierendes Heimnetzwerk.

Zuerst lädt man über den Paketmanager das Paket fetchmail herunter. Nach der Installation verlässt man den Paketmanager wieder und startet einen Editor seiner Wahl, denn fetchmail erwartet eine Konfigurationsdatei im *Home*-Verzeichnis des Benutzers, welcher schlussendlich das Programm startet.

In dieser Datei muss man angeben, von welchem Konto man die E-Mails holen möchte und an wen die E-Mails weitergeleitet werden sollen. Wie die Datei am Ende aussehen soll bzw. was darin stehen sollte, erklärt diese Beispieldatei:

```

# fetchmail Konfigurationsdatei
# Server, von dem die Mail geholt werden soll
server      pop3.web.de
user        hans

```

```
pass      streng_geheim
proto     pop3
to        musterhans
```

Diese Datei speichert man unter dem Namen `.fetchmailrc` im *Home*-Verzeichnis und setzt die Rechte für die Datei auf den Wert von 710. Dies geschieht mit dem Befehl:

```
chmod 710 .fetchmailrc
```

Natürlich kann man auch mehrere E-Mail Konten in die Datei eintragen. Die Begriffe zur Erklärung:

- *server*: E-Mail Server
- *user*: Benutzername
- *pass*: Passwort
- *proto*: Verwendetes Protokoll (z. B. POP3 oder IMAP)
- *to*: Welcher lokale Benutzer soll diese E-Mails bekommen

Sollten mehrere E-Mail Konten auf dem E-Mail-Server existieren, so kann man die Angabe des Servers in der Datei so lange weglassen, bis sich dieser ändert. Eine weitere Option ist `keep`. Sollte diese Option für ein Konto angegeben werden, so bleiben die E-Mails auf dem Server und es werden nur die Kopfzeilen heruntergeladen. Die Option ist sehr gut zum testen der Konfiguration geeignet.

Sobald man `fetchmail` startet, erwartet es, dass die in der Konfigurationsdatei angegebenen Benutzer auch auf dem Rechner existieren. Sollte ein Benut-



Einrichtungsdialog Thunderbird

zer noch nicht auf dem System vorhanden sein, so legt man ihn schnell per

```
adduser oder useradd
```

an. Wichtig hierbei ist, dass man sich das Passwort für den Benutzer merkt, denn später in der Konfiguration des E-Mail Programms wird man dies eintragen müssen.

Ist alles erledigt, kann man `fetchmail` zum ersten Mal starten. In der Ausgabe auf der Konsole oder im Terminal sollte nun angezeigt werden, auf welches Konto zugegriffen wurde und welches neue E-Mails enthält.

Nun wäre es aber sinnvoll, wenn `fetchmail` ständig im Hintergrund laufen würde und innerhalb eines bestimmten Intervalls automatisch auf neue E-Mails prüfen würde. Das erreicht man mit einem der diversen Optionsschalter des Befehls.

Wenn `fetchmail` automatisch alle 500 Sekunden nach neuen E-Mails schauen soll, reicht der Aufruf:

```
fetchmail -d 500
```

Der Schalter `-d` steht hierbei für Daemon, also Hintergrunddienst. Weiterhin sollte man sich die Startskripte unter `/etc/init.d/` anschauen. In der dortigen Datei zu `fetchmail` kann man den Daemon, der normalerweise alle 300 Sekunden nachprüft, ob neue E-Mails vorhanden sind, verlängern, ganz ausschalten oder andere Optionen hinzufügen.

Da man jetzt die Grundeinstellungen vorgenommen hat und der Prozess von alleine arbeitet, kann man sich daran machen, den entsprechenden Server aufzusetzen.

Verteilen der E-Mails

Am Anfang des Artikels wurde der grobe Unterschied zwischen IMAP und POP3 erklärt. Was liegt also näher, als einen IMAP Dienst auf dem PC zu installieren?

Am einfachsten geht das mit dem Paket `uw-imapd`. Dieses Paket steht jeder Distribution per Paketmanager zur Verfügung und muss über diesen nur installiert werden. Sollte das Paket wider Erwarten nicht über den Paketmanager verfügbar sein, so kann man es auch aus den Quellen [3] installieren.

Der Grundsatz von uw-imapd lautet grob übersetzt: »Installieren, fertig.«. Und genau so verhält es sich auch. Das Einzige, was uw-imapd voraussetzt, ist, dass neue E-Mails unter `/var/mail` gespeichert werden. Wer fetchmail richtig konfiguriert hat, hat mit dieser Regel keine Probleme. Unter diesem Verzeichnis werden standardmäßig alle E-Mails unter dem jeweiligem Benutzernamen in einer Art Textdatei gespeichert. Nun startet man als User das Programm mit dem Befehl:

```
imapd
```

Tja, und schon hat man aus dem PC einen IMAP E-Mail Server gemacht der nun auf Anfragen wartet. Alle Einstellungen wie z. B. Sicherheitsoptionen oder auch der Port 143 sind bereits voreingestellt und bedürfen im Normalfall keiner Änderung. Sollte das nicht reichen, so genügt ein Aufruf von:

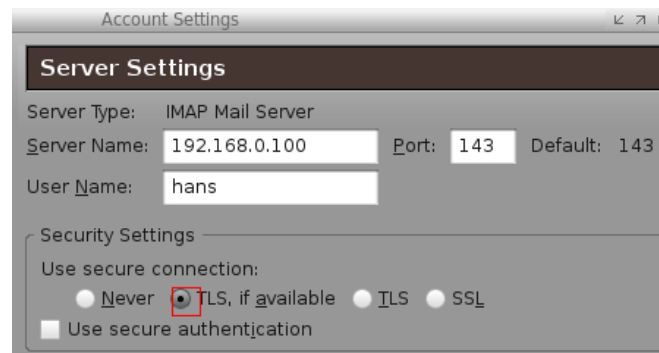
```
dpkg-reconfigure uw-imapd
```

Konfiguration des E-Mail Programms

Man startet sein bevorzugtes E-Mail Programm. Für die Einstellungen zum Abrufen und Senden wird hier stellvertretend das Programm Thunderbird verwendet:

Nach dem Start von Thunderbird gibt man seinen Namen und seine E-Mail Adresse an. Bei der Auswahl des Kontotyps gibt man IMAP an. Als Server trägt man die IP des Rechners ein, auf dem der fetchmail Dienst läuft. Als Postausgangsserver kann man die Vorgaben des E-Mail Servers (z. B. smtp.1und1.de) verwenden.

Im nächsten Schritt muss man den Benutzernamen angeben, unter welchem man die E-Mails vom Server holen möchte. Werden z. B. die E-Mails für den Benutzer »hans« unter dem Benutzernamen »musterhans« gespeichert, so muss in



Sichere Verbindung garantiert

das Feld von Thunderbird der Benutzer »musterhans« eingetragen werden. Im folgendem Schritt weist man dem Konto noch einen Namen zu. Jedes weitere Konto wird auf die gleiche Weise eingerichtet. Damit ist der Konfigurationsdialog beendet.

Als nächstes klickt man in Thunderbird auf den Reiter »Bearbeiten« und wählt dort die Kontoeinstellungen an. Im nun folgendem Dialog wählt man unter dem Konto den Unterpunkt »Server Settings« aus und klickt unter »Security Settings« den Punkt »TLS, falls verfügbar« an. Nun speichert man die Einstellungen.

Sollte alles funktionieren, werden bald die ersten E-Mails eingehen und man kann die ganze Funktionalität seines E-Mail Programms nutzen.

Weitere Möglichkeiten

Wer möchte, kann die eingegangenen E-Mails auch an weitere Programme wie z. B. procmail [4] übergeben. Es ist auch möglich, die eingehenden E-Mails gleich von z. B. spamassassin [5] prüfen und vorselektieren zu lassen. Um Clients, die keinerlei Verbindung in das Internet haben (oder keine haben dürfen), das Senden von E-Mails zu erlauben, müssen die ausgehenden Mails an einen MTA (Mail Transfer Agent) wie zum Beispiel sendmail übergeben werden. Ein Howto dazu ist z. B. auf Linuxhaven [6] erhältlich.

Fazit

Mit recht einfachen Mitteln lässt sich ein funktionaler und einfacher E-Mail-Dienst im Heimnetzwerk installieren und verwalten. Da die Mails, die man von den einzelnen Konten herunterlädt, lokal auf dem Heimrechner gespeichert werden, empfiehlt sich natürlich ein regelmäßiges Backup der Daten.

Jan Radecker

cyclame@yalmagazine.org

Informationen

- [1] <http://www.uni-koeln.de/rrzk/mail/server.html>
- [2] http://de.wikipedia.org/wiki/Mail_Transfer_Agent
- [3] <http://www.washington.edu/imap/>
- [4] <http://www.procmail.org/>
- [5] <http://de.wikipedia.org/wiki/SpamAssassin>
- [6] <http://www.linuxhaven.de/dlhp/HOWTO/DE-Offline-HOWTO-5.html>

Wine

Wer mit Linux arbeitet, aber auf Windows-Programme nicht verzichten möchte, kann mit Hilfe von Wine [1] sehr viele dieser Programme fast ohne Performance-Einbußen unter Linux benutzen.

Was ist Wine?

Als sich abzeichnete, dass Linux im Bereich der Betriebssysteme keine Eintagsfliege bleiben wird, kam der Wunsch auf, für Windows geschriebene Programme unter Linux ausführen zu können. Deswegen gründete sich 1993 das Wine-Projekt, das nach 15 Jahren mit seinem Release 1.0 endlich eine stabile Version vorgestellt hat. Bei dem Namen handelt es sich um ein rekursives Akronym für *Wine is not an emulator*, auf dessen Bedeutung im späteren Verlauf des Artikels noch weiter eingegangen wird. Die Mitarbeiter haben es sich zum Ziel gesetzt, möglichst viele Windows-Programme unter Linux und anderen Posix-kompatiblen Betriebssystemen wie Mac OS X oder BSD bereitzustellen.

Während der letzten Jahre hat sich eine Zusammenarbeit zwischen Google und Wine entwickelt, die beiden Partnern zum Vorteil gereicht: Google engagiert sich finanziell oder mit Personal, Wine stellt im Gegenzug notwendige Software-Implementierungen für einige nicht-Windows-Betriebssysteme bereit. Chrome, Googles neuer Webbrowser, ist jedoch merkwürdigerweise nur mit einigem Aufwand lauffähig, obwohl er (ebenso wie der Safari-Browser) auf dem Quellcode des Linuxbrowsers-Konqueror basiert.

Die Arbeitsweise von Wine

Wine stellt für Unix-Systeme Ersatzkomponenten für Windows bereit, ohne dabei Geschwindigkeit für die Nachbildung (Emulation) des Windows-Betriebssystems zu verlieren. Um mit Wine arbeiten zu können, ist eine Installation von Windows nicht nötig. Lediglich die Programme, deren Nutzung erwünscht ist, müssen vorhanden sein. Wie aus dem Namen hervorgeht, emuliert Wine nicht die jeweilige Windows-Umgebung, die ein Programm benötigt um zu funktionieren, sondern setzt den Hebel beim Windows Application Programming Interface, kurz API genannt, an. Bei einer API, auch Programmierschnittstelle genannt, handelt es sich um eine Verbindung, die es Programmierern u. a. ermöglicht, Software daran anzubinden und damit auf dem Betriebssystem lauffähig zu machen. Da es sich bei Windows um ein proprietäres Betriebssystem handelt, dessen Quellcodes nicht für jedermann einsehbar sind, bleibt es für das Wine-Team schwierig die jeweilige API durch eine selbst programmierte zu ersetzen. Das Problem hierbei ist, dass es sich, zumindest bei umfangreichen Programmen, nicht um einen standardisierten Vorgang handelt, sondern die Entwickler jedes Mal ordentlich tüfteln müssen. Auch der anschließende Testaufwand ist immens. Gering sind hingegen die

Bedürfnisse von Wine. Gemeinsam mit seinem Linux Basis-Betriebssystem benötigt es weniger Ressourcen als Windows für sich allein. Wenn keine Windows-Anwendung läuft, benötigt Wine weder Arbeitsspeicher noch Prozessorkraft. Lediglich 20 MB Speicherplatz müssen ihm eingeräumt werden.

Die FAQ im Wine-Wiki [2] geben detailliert Auskunft über die Arbeitsweise, Einsatz, Installation und Fehlerbehebung von Wine.

Installation und Konfiguration

Man findet die aktuelle und stabile Version 1.0.0-1 in den Ubuntu-Repositories. Die neueste Version mit der Bezeichnung Wine 1.1.14 ist noch nicht stabil und befindet sich noch in Entwicklung. Die Installation von Wine erfolgt entweder über die Paketverwaltung oder per Terminal mit den Befehlen:

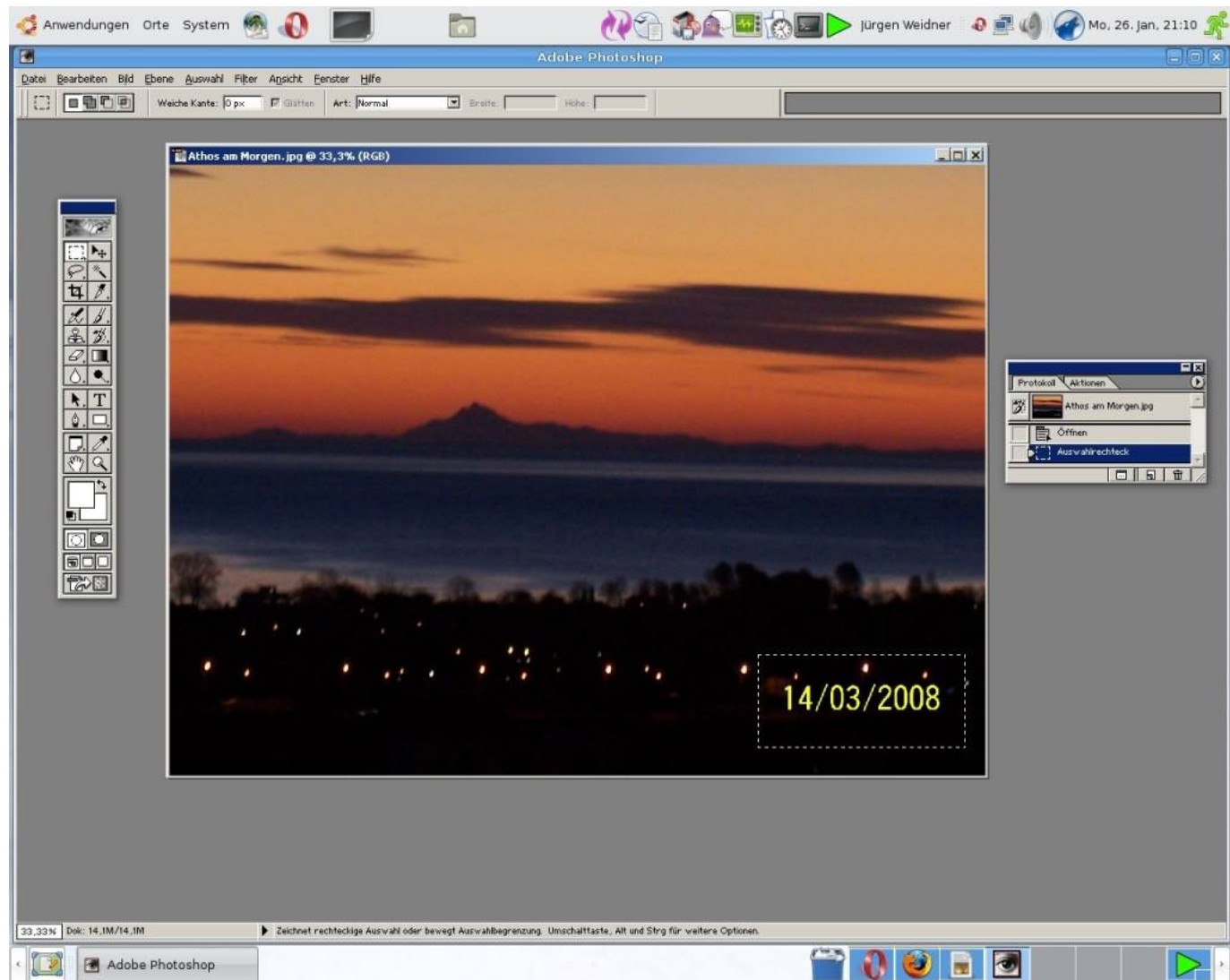
```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install wine
```

Anschließend findet man Wine unter *Anwendungen* in der Menüleiste. Als nächster Schritt bietet sich die Konfiguration an. Unter *Konfiguriere Wine* findet sich ein Fenster, das es ermöglicht, einige Einstellungen vorzunehmen. Standardmäßig ist in Anwendungen XP als Windows-Version eingestellt. Aus einer Liste kann von der Uralt-Version Windows 2.0 bis zu Vista das für die eigenen

Zwecke notwendige Windows einstellen. Sollten man AUDIO-Support benötigen, so wählt Wine einen »bevorzugten Treiber« für das System aus. Der Button *Test Sound* gibt darüber Aufschluss, ob Wine richtig gewählt hat. Falls nicht, kann man die für sein Linux passenden Einstellungen manuell einrichten. Hinter dem Reiter *Grafik* verbirgt sich u. a. die Bildschirmauflösung. Durch Ziehen des Schiebereglers bestimmt man die gewünschte Darstellungsgröße. *Desktop-Integration* gestattet es uns, Einfluss auf die Verknüpfung der Windows-Shell mit dem jeweiligen Linux-Pendant zu nehmen. Eigene Dateien wird hier beispielsweise mit `/home/Linuxuser` gleichgesetzt. Änderungen im Segment *Bibliotheken* sind nur notwendig, wenn dem Programm die zum Start und zur korrekten Ausführung notwendigen DLLs fehlen (siehe »Die Feinjustierung«). Hinter *Laufwerke* findet man die Zuordnung der von Windows vergebenen Laufwerks-Buchstaben zu den Linux-spezifischen Bezeichnungen, wobei C: von Wine als `../drive_c` verwaltet wird. Hier nicht angezeigte Laufwerke können manuell hinzugefügt werden. Hat man eine der Einstellungen verändert, so ist ein Neustart von Wine erforderlich, damit die Neuerung übernommen wird.

Installation von Windows-Programmen

Um nun Windows-Programme zu installieren, geht man wie gewohnt vor. Wine muss vor Programminstallationen nicht separat gestartet werden. Es erkennt selbstständig Windows-Software und aktiviert sich automatisch. Legt man eine CD in sein



Adobe Photoshop in Ubuntu

Laufwerk, so öffnet sich nach Doppelklick auf *setup.exe* das Windows-typische Setup-Fenster und man installiert das Programm wie üblich.

Möchte man ein Programm aus dem Internet herunterladen, so bietet z. B. Firefox direkt das Öffnen mit dem Wine Windows-Programmstarter an.

Nachdem mit einem Klick auf *OK* unsere Zustimmung gegeben haben, öffnet sich sowohl das Download-Fenster des Browsers, als auch das Setup-Fenster der zu installierenden Software. Auch hier durchläuft die Installation die Windows-übliche Prozedur. Schon während des Vorgangs fühlt man sich wie auf einem Windows-Rechner. Die Icons und die Gestaltung der Fenster lassen keinen Zweifel daran, für welches Betriebssystem Wine seine Dienste offeriert. Sollte sich übrigens, was bei einer Installation von mehreren CDs sehr lästig ist, eine CD nicht wieder auswerfen lassen, so hilft der Konsolenbefehl *wine eject* [3] weiter.

Um unerwünschte Software wieder zu entfernen, aktiviert man unter *Anwendungen – Wine* die Funktion *deinstalliere Wine Applikationen*. Im sich nun öffnenden Wine Anwendungsentferner wählt man das zu löschende Programm aus und bestätigen mit Klick auf *Entfernen*. Die Nachfrage des Programms, ob wir wirklich sicher sind, beantwortet man *Ja*. Wine weist darauf hin, dass sich auf diese Weise Software nebst zugehörigen Dateien und Bibliotheken nicht zuverlässig entfernen lässt. Um wirklich sicher zu sein, dass alle Komponenten entfernt wurden, muss Wine selbst deinstalliert werden, was entweder mit Hilfe der Paketverwaltung oder mit dem Befehl *rm* per Terminal erledigen (siehe »Risiken«) .

Windows Programme nutzen

Zunächst ist wichtig zu erwähnen, dass Wine niemals als root gestartet werden sollte. Da es nicht unmöglich ist, dass Viren, Trojaner, Würmer etc.,



Konfiguration der AUDIO-Einstellung

die für Windows geschrieben wurden, den eigenen Computer befallen, darf man ihnen keinesfalls den Weg ins Innere des Systems öffnen. Also startet man Wine nur als normaler User. Auch wenn man mit einem Linux-System in diesem Falle derzeit nicht viel zu befürchten hat: Sobald man sich in einem Netzwerk mit Windows-Rechnern befindet, ist die Gefahr der Weiterverbreitung groß. Auch wird

aus denselben Gründen vor einem Download von Treibern aus dem Internet gewarnt.

Seit Wine-Versionen ab 0.9.60 ist es nicht mehr notwendig, das Terminal zu benutzen. Für den normalen Umgang mit dem Programm gibt es eine grafische Oberfläche, die der von Windows stark ähnelt. Es wird empfohlen, alle 3D-Desktops wie Beryl oder Compiz zu deaktivieren, es könnte ansonsten zu Komplikationen mit der Darstellung kommen. Die installierten Programme findet man normalerweise unter *Anwendungen – Wine-Programme*. Sollte das nicht der Fall sein, so bietet die Funktion *Durchsuche C:* an, die durch Wine installierten Programme auf der fiktiven Festplatte *C:* zu durchstöbern.

Welche Programme funktionieren mit Hilfe von Wine ?

Die Faustregel heißt: Je älter das Programm ist, desto größer ist die Wahrscheinlichkeit, dass es von Wine unterstützt wird. Das Team bemüht sich, so viel Windows-Software wie möglich auf unixoiden Systemen nutzbar zu machen. Inzwischen ist auch die Anbindung sehr komplexer Software gelungen. Eine verbindliche Aussage hierüber gibt die Wine Application Database [4]. Je nach Funktionssicherheit wird der Status Platin, Gold oder Silber vergeben. Aber auch Programme, die diesen Status nicht besitzen, sind möglicherweise doch noch zum Anspringen zu bewegen. In einem solchen Fall lohnt es sich Kontakt mit dem Wine Users Forum [5] aufzunehmen. Die in der Platinum List geführte Software funktioniert auf einer Wine

Standard-Installation ohne weiteres Zutun problemlos. Kleinere Einstellungen sind notwendig, um mit der Software der Gold-List einwandfrei arbeiten zu können. In die Silver-List schafft es eine Anwendung, die vom Nutzer noch etwas Nacharbeit erfordert. Um den gesamten Inhalt der sehr umfangreichen Listen anzeigen zu lassen, genügt ein Klick auf Platinum, Gold oder Silver. Auf der Seite *Browse Apps* findet man auch eine Suchfunktion. Derzeit sollen über 1000 Windows-Programme und Spiele unter Linux mehr oder weniger gut funktionieren.

Die Feinjustierung

Da manche Windows-Anwendungen nicht ohne weiteres funktionieren, muss eventuell nachgeholfen werden. Gelegentlich sorgen fehlende Dynamic Link Librarys (auch als DLLs bekannt) dafür, dass der Programmstart nicht funktioniert. Abhilfe schafft das Shell-Script *Winetricks* [6], das beim Download fehlender DLLs behilflich ist. In Wine ist auch eine Registry enthalten. Um hier Einträge zu ändern, gibt man *regedit* [7] in ein Terminal ein, und der Registry-Editor wird geöffnet. Hierbei handelt es sich um eine grafische Oberfläche, die es erlaubt, Einträge hinzuzufügen oder bestehende zu ändern. Allen, die nicht mit den notwendigen Wissen ausgestattet sind, helfen die *Useful Registry Keys* [8] bestimmt weiter.

Kommerzielle Anbieter

Auf Wine basierend wird CrossOver von der Firma CodeWeavers [9] vertrieben und steht in keiner

Konkurrenz zu dem Projekt. Ja, man könnte CrossOver sogar als kommerziellen Arm von Wine ansehen. Beide entwickeln den Code weiter und profitieren gemeinsam von den Ergebnissen ihrer Arbeit. Dafür, dass CrossOver so stark von Open Source profitiert, bedankt es sich, indem das Gehalt einiger hauptamtlicher Wine-Entwickler von der Firma bezahlt wird. Gegenstand des Unternehmens ist, Windows-Programme und -Spiele auf Li-

nux- und Mac-Systemen verfügbar zu machen. Im Gegensatz zu Wine hat man hier eher Firmenkunden im Visier und zielt weniger auf Spiele ab.

Cedega widmen wir nur einen Seitenblick, denn die Daseinsberechtigung dieses Ablegers der Firma TransGaming Technologies [10] ist umstritten. Dieser Anbieter konzentriert sich, im Gegensatz zu CrossOver, hauptsächlich auf den Spielektor.

The screenshot shows the WineHQ website interface. At the top, there are navigation links for WineHQ, Wiki, AppDB, Bugzilla, and Forums. A search bar is located on the right. The main content area is titled 'WINEHQ World of Warcraft 3.0.x'. Below the title, there is a category breadcrumb: 'Category: Main > Games > Online (MMORPG) Games > World of Warcraft > 3.0.x'. The main content is divided into several sections: 'Name', 'Version', 'License', 'URL', 'Votes', 'Rating', 'Wine Version', 'Free Download', 'Description', 'Selected test results', 'What works', 'What does not', 'What was not tested', 'Additional Comments', and 'Test Results'. The 'Test Results' section contains a table with columns for Distribution, Test date, Wine version, Installs?, Runs?, Rating, and Submitter.

	Distribution	Test date	Wine version	Installs?	Runs?	Rating	Submitter
Current	Debian GNU/Linux "Lenny"	Jan 25 2009	1.1.13	Yes	Yes	Platinum	Argenta
Show	Ubuntu 8.10 "Intrepid" (+ variants like Kubuntu)	Jan 25 2009	1.1.13	Yes	Yes	Gold	Anthony Totarella
Show	Ubuntu 8.10 32-bit Desktop Edition	Jan 21 2009	1.1.12	N/A	Yes	Gold	Luis
Show	Arch Linux (rolling release)	Jan 10 2009	1.1.12	N/A	Yes	Gold	Binaldus

Ein Windows-Programm aus der Platinum-List

Auch hier bildet Wine das Fundament, auf dem Cedega aufgebaut ist. Es flossen allerdings keine Weiterentwicklungen an das Wine-Projekt zurück. Gegenwärtig gibt es kaum Gründe, warum man kostenpflichtigen Produkten jener Firma den Vorzug geben sollte, denn man arbeitet dort mit veraltetem Code des Open Source-Projekts. Wine besitzt zurzeit eine bessere Unterstützung für Direct3D, sodass viele Spiele mit Hilfe von Wine in spürbar besserer Qualität zu spielen sind. Einzig und allein proprietärer Kopierschutz wird besser unterstützt. Zudem überprüft Cedega die Originalität der Kopie zusätzlich selbst. Deshalb lassen sich unter diesem Anbieter Sicherheitskopien kaum einsetzen.

Risiken

Wie oben bereits gesagt, ist Wine nicht resistent gegen Windows-Schadware-Angriffe von infizierten Datenträgern und aus dem Internet. Wine fordert einige Maßnahmen [11], um sich dagegen zu schützen:

Vorsicht bei Links, deren URL man weder kennt noch versteht.

- Kein Programm mit grafischer Oberfläche als root starten (inkl. Wine).
- Es wird empfohlen, einen Virens scanner wie z. B. ClamAV zu nutzen.

- Gegebenenfalls sollte das Wine-Laufwerk Z: entfernt werden, da es Zugang zum root-Verzeichnis hat.
- Programme, die infiziert sein können (ein Virens scanner durch einen Windows-Virens scanner kann hier Klarheit bringen), sollten nur in virtuellen Maschinen genutzt werden

Eine böse Falle lauert in einigen älteren Versionen von GNOME [12] auf den Nutzer. Wird Wine gelöscht und in den Müll verschoben, so wird beim Leeren des Mülleimers unser `/home` Verzeichnis und alle editierbaren Einträge des Verzeichnisses `/root` gelöscht!

Seit GNOME 2.21.92 ist dieser Fehler behoben. Ubuntu 8.04 arbeitet mit GNOME 2.22.

Fazit

Wenn man die Programme, die in der Platinum- und Gold-List aufgeführt sind, nutzt, so ist Wine unbedingt als eine Bereicherung zu betrachten. Wendet man sich jedoch Applikationen zu, die nicht optimal unterstützt werden, ist Wissen und Geduld gefragt. Trotzdem kann der Versuch, manche Windows-Software unter Wine laufen zu lassen, zu einem frustrierenden Erlebnis werden.

Jürgen Weidner
joschi@yalmagazine.org

Informationen

- [1] <http://www.winehq.org/>
- [2] <http://wiki.winehq.org/FAQ>
- [3] <http://wiki.winehq.org/eject>
- [4] <http://appdb.winehq.org/>
- [5] <http://forum.winehq.org/>
- [6] <http://wiki.winehq.org/winetricks>
- [7] <http://wiki.winehq.org/regedit>
- [8] <http://wiki.winehq.org/UsefulRegistryKeys>
- [9] <http://www.codeweavers.com/products/>
- [10] <http://www.transgaming.com/>
- [11] <http://wiki.winehq.org/FAQ#head-1c91cac836dd52754c846d2ef62be4f346eebe87>
- [12] <http://www.gnome.org/>

Bunte Seite

Firefox wie Vi bedinen

VIMpertor [1] ist ein Firefox-Addon, welches per Voreinstellung die GUI-Elemente (also die Menü- und Toolbar) ausblendet und Firefox wie den Texteditor Vim steuern lässt. Neben der Navigation per Tastatur funktionieren so auch einige Kommandos wie `:quit`. Es gibt auch solche, die man nicht vom Editor kennt, z. B. wird mittels `:dialog preferences` das Einstellungsfenster von Firefox angezeigt.

Dateiübertragungen bei Pidgin

Per Default ist es vielen Nutzern nicht möglich, Dateien über Pidgin zu versenden. Zumindest für XMPP hilft es einen Port freizugeben: Nachdem im Router bspw. der Port 8010 geöffnet worden ist, kann in Pidgin unter *Werkzeuge – Einstellungen – Netzwerk – Port-Bereich*, auf dem gehört werden soll, manuell bestimmen der entsprechende Port angegeben werden.

Für ICQ soll es hilfreich sein, unter *Konten – Konten verwalten – ICQ-Nummer – Bearbeiten – Erweitert* den Haken bei *Benutze immer den AIM/ICQ-Proxyserver...* zu setzen. Bei unseren Tests funktionierte dies sowohl mit ICQ 6.5 als auch mit ICQ 6.1, nicht aber mit der Version 6.0.

Nautilus-Scripts-Pack

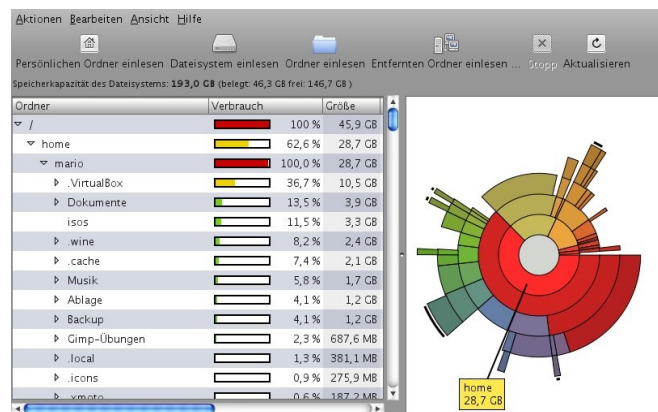
Nautilus lässt sich recht einfach durch Skripte erweitern. Eben solche mit dem Order bzw. der Datei als Argument können in den Skriptordner des Browsers, `~.gnome2/nautilus-scripts`, gespeichert werden. Jede Datei, die dort ausführbar ist, lässt

sich über das Kontextmenü *Skripte* erreichen. gnomelook.org hat eine eigene Kategorie [2] für derartige Skripte. Wer nicht lange suchen, Nautilus aber dennoch erweitern will, sollte sich das »Nautilus Scripts Pack« [3] ansehen.

Analyse des belegten Speicherplatzes

Besonders vergleichsweise kleine 80-GiB-Festplatten sind recht schnell voll und man fragt sich warum. Einige Tools visualisieren die Platte und können so die Frage beantworten. So gibt es für GNOME Baobab und für KDE Filelight. Diese Programme zeigen den Platz in einem Kuchen- und Balkendiagramm sehr übersichtlich an.

Einen anderen Weg gehen Graphical Disk Map (GdMap) und der Dateibrowser Konqueror: Sie visualisieren die Platte als Rechteck, welches in viele kleinere Rechtecke unterteilt ist, größere Rechtecke bedeuten mehr Platz.



Baobab visualisiert die Festplattenbelegung

Wer sich zwischen den Ansichten nicht entscheiden kann, sollte vielleicht einen Blick auf *KDirStat* werfen: Das Programm kombiniert beide Ansichten. Für die Konsole gibt es ebenfalls einige Programme, u. a. *discus*.

Alle diese Programme sollten sich in der Regel über den Paketmanager aktueller Distributionen installieren lassen.

Mario Fuest
keba@yalmagazine.org

Denk an die Milch

Remember The Milk [4] ist ein Web 2.0 Portal zur Aufgabenverwaltung. Die Nutzung ist sehr komfortabel; so kann man seine Aufgaben gruppieren, mit Tags versehen und mit anderen teilen. Neue Aufgaben können ebenfalls per E-Mail hinzugefügt werden und alle Aufgaben sind als Atom-Feed sowie im iCalendar-Format verfügbar.

Thomas Rudolph
holloway@yalmagazine.org

Informationen

- [1] Firefox-Addon VIMPerator: <http://vimperator.org/trac/wiki/Vimperator>
- [2] Kategorie »Nautilus Scripts« bei gnome-look.org: <http://gnome-look.org/index.php?xcontentmode=188>
- [3] Nautilus Scripts Pack: <http://gnome-look.org/content/show.php/Nautilus+Scripts+Pack?content=90330>
- [4] Remember The Milk Portal: <http://www.rememberthemilk.com/>

Rekursives Umbenennen – Lösung

In der letzten Yalm-Ausgabe haben wir nach einer kurzen und verständlichen Lösung für das Umbenennen von Dateien über Verzeichnisse hinweg gefragt. Bis zum 20. Januar haben wir die Einsendungen der Leser gesammelt. Diese möchten wir nun vorstellen und mit unserer Musterlösung vergleichen.

Rätsel

Zur Wiederholung hier noch einmal das Rätsel: »Wie kann Miriam ihre oga-Dateien, die sich über hunderte von Unterverzeichnissen verteilen in ogg-Dateien umbenennen? Gesucht ist ein eleganter Befehl, um dieses rekursive Umbenennen auf der Konsole durchzuführen. Elegant bedeutet kurz, aber verständlich.«

Lösungsvorschläge

Innerhalb von fünf Tagen nach Erscheinen der Yalm Ausgabe Nr. 01/2009 haben wir die Vorschläge der Leser gesammelt. Wir möchten uns für die rege Teilnahme ganz herzlich bedanken. Hier sind die Namen der Leser in der Reihenfolge ihrer Einsendungen:

Dennis Pattmann, Josef Bichlmeier, Clemens Kaposi, Sacha Wüstemann, Daniel Herzog, Alexander Schier, Toni Burgarello, Lukas Elsner, Simon Gerber, Karl-Heinz Waldmann, Dennis Schreiber und Rico Engelman.

Die eingereichten Lösungsvorschläge lassen sich in zwei Gruppen unterteilen:

4. Lösungen mit find

5. Lösungen mit for

Ein Vertreter der *find*-Variante ist z. B. die Musterlösung, die sich der Autor ausgedacht hatte, als das Rätsel geschrieben wurde. Sie sieht so aus:

```
find -name '*.oga' -exec rename -s/\.oga$/\.ogg/' {} \;
```

Eine genaue Erklärung zu diesem Befehl folgt weiter unten in diesem Artikel. Sehr ähnliche Lösungsvorschläge wurden von Sascha Wüstemann, Daniel Herzog, Lukas Elsner, Karl-Heinz Waldmann und Dennis Schreiber eingereicht.

Ein Beispiel für die *for*-Variante ist die Einsendung, die wir von Dennis Pattmann erhalten haben (dies war übrigens die allererste Einsendung):

```
for i in `find /home/miriam/Musik/ -name "*.oga"; do name=`echo $i | awk -F. '{print $1}'` && mv $name.oga -> $name.ogg; done
```

Die mit Abstand kürzeste Version kommt von Clemens Kaposi, unserem Einsender Nr. 3:

```
for i in **/*.oga(.); mv $i $i:r.ogg
```

Diese Befehlsfolge ist so kurz und einfach, dass der Autor nicht glauben wollte, dass sie funktioniert. Einen kleinen Haken gibt es tatsächlich, der Befehl funktioniert nicht in der Bash, sondern in der Z-Shell (zsh). Diese sehr mächtige Shell kann mit dem gleichlautenden Paket installiert werden.

Eine kurze Erläuterung zum Befehl: das »do« zu Beginn einer Schleife kann in der zsh weggelassen werden, wenn die Schleife so wie hier nur aus einem Befehl besteht. Das Suffix »:r« hinter »\$i« entfernt die Dateierweiterung. Schließlich sorgt die Wildcard »**« dafür, dass vom aktuellen Pfad ausgehend alle Unterverzeichnisse rekursiv durchlaufen werden.

Am meisten Mühe hat sich Toni Burgarello mit der Auflösung des Rätsels gemacht. Von ihm stammt diese Lösung:

```
find . -type f -name "*.[0o][Gg][Aa]" | while read FILENAME; do mv -> "${FILENAME}" "${FILENAME%.[0o][Gg][Aa]}.ogg" ; done
```

Na, ob sich Miriam das merken kann?

Erklärung

Kommen wir nun zur Erklärung der Musterlösung, die stellvertretend für alle eingesandten *find*-Varianten gilt:

```
find -name '*.oga' -exec rename →
's/\.oga$/\.ogg/' {} \;
```

Der Befehl `find -name '*.oga'` sucht alle Dateien, die den String `.oga` in ihrem Namen haben, ab dem Verzeichnis, aus dem der Befehl ausgeführt wurde, inklusive aller Unterverzeichnisse. Durch das Argument `-exec` wird das Ergebnis von `find` an den nächsten Befehl weitergeleitet.

Vom Befehl `rename` gibt es zwei unterschiedliche Ausführungen; die eine hat die Syntax:

```
rename from to file
```

also z. B.:

```
rename .htm .html *.htm
```

um alle Dateien eines Verzeichnisses mit der Endung `htm` in `html` umzubenennen.

Diese Variante funktioniert nicht im Zusammenspiel mit dem Befehl `find` (zumindest beim Autor nicht).

Die andere Variante von `rename` gibt es nur auf Debian-basierten Distributionen und ist in Perl implementiert. Die Syntax lautet:

```
rename [ -v ] [ -n ] [ -f ] perlexpr
[ files ]
```

`[-v]` zeigt die veränderten Dateien an (»verbose«)

`[-n]` macht einen Trockenlauf mit Anzeige der Dateien (»no-act«)

`[-f]` überschreibt bestehende Dateien (»force«)

`[perlexpr]` ist eine Perl Regular Expression [1]

`[files]` sind die zu bearbeitenden Dateien

Wir verwenden die zweite Variante (weil sie auf Ubuntu installiert ist). Die `perlexpr` lautet

```
's/\.oga$/\.ogg/'
```

und bedeutet: substituieren `oga` mit `ogg`. Der Grundaufbau sieht so aus:

```
's/alt/neu/'
```

Gesucht wird nach der Dateiendung `.oga`. Der Punkt hat in der Perl Expression eine andere Bedeutung (»match any character«); deshalb wird er mit `\` escaped, damit er auch wirklich als Punkt angesehen wird. Das `$` bedeutet »Ende der Zeichenkette«, damit nur die Endung `oga` gefunden wird und nicht etwa ein Text »oga« mitten im Dateinamen.

Für das Argument `[files]` wird `{}` eingesetzt.

```
Find -exec
```

liefert seine Ergebnisse an dieses Klammersymbol. Somit wird für jede Datei, die `find` findet, der `rename`-Befehl ausgeführt. Das Semikolon be-

endet das `exec`-Kommando und muss mit `\` escaped werden, damit die Shell es ignoriert.

Durch die mächtige Regular Expression kann der `rename`-Befehl noch weitaus mehr leisten als hier dargestellt. Weitere Möglichkeiten finden sich hier [2].

Vorsicht

Da die Kombination der Befehle `find` und `rename` eine Massenmutation bewirkt, ist besondere Vorsicht geboten. Folgende Sicherheitsregeln sollten befolgt werden:

- den Befehl nie mit root-Rechten (`sudo`) ausführen
- den Befehl immer zuerst mit dem Argument `-n` laufen lassen, um die potentiellen Auswirkungen zu erkennen
- bevor der Befehl ausgeführt wird, muss immer in das Verzeichnis gewechselt werden, ab dem der Befehl gelten soll. Nie im root-Verzeichnis ausführen!

Ralf Hersel

rhersel@yalmagazine.org

Informationen

- [1] Perl Regular Expression: <http://www.troubleshooters.com/codecorn/littperl/perlreg.htm>
- [2] Rename: <http://tips.webdesign10.com/how-to-bulk-rename-files-in-linux-in-the-terminal>

Schlussbemerkungen

Yalm ist ein privates, nichtkommerzielles Projekt. Die Zeitschrift erscheint am 15. eines Monats.

Rückmeldungen zu unserem Magazin – seien es Artikelwünsche, Verbesserungsvorschläge, Lob oder auch Kritik – sind herzlich willkommen. Schreibt einfach an redaktion@yalmagazine.org oder postet in unserem Forum auf <http://www.yalmagazine.org/forum>.

Wir suchen immer engagierte und zuverlässige Helfer, die bei unserem Magazin mitarbeiten wollen. Nicht nur Layouter mit guten OpenOffice-Kenntnissen und natürlich Autoren sind gerne gesehen, sondern auch Programmierer und Entwickler sind herzlich eingeladen, bei Yalm mitzumachen. Schreibt uns bei Interesse bitte eine E-Mail an redaktion@yalmagazine.org oder seht euch für weitere Details die Rubrik »Mitmachen« auf unserer Homepage an.

Layout

Yalm wird mit OpenOffice erstellt; die Bearbeitung und Korrektur der Texte erfolgt derzeit noch in GoogleDocs. Die jeweils gültige Dokumentvorlage kann von der Yalm-Homepage (Rubrik »Extras«) [heruntergeladen](#) werden.

Listings und weiterführende Informationen

Layoutbedingte Zeilenumbrüche werden mit einem Pfeil → dargestellt. Eventuell notwendige Leerzeichen stehen vor diesem Pfeil.

Weiterführende Informationen, Listings und Dateien zu einzelnen Artikeln werden in der Rubrik »Extras« der Yalm-Homepage zum Download angeboten.

An dieser Ausgabe haben mitgewirkt:

Bernhard Posselt (Autor, Korrektur)
 Christoph Schmied (Autor, Korrektur)
 Frank Brungräber (Layout, Korrektur)
 Jan Guth (Korrektur)
 Jan Radecker (Autor, Korrektur)
 Jürgen Weidner (Autor, Korrektur)
 Mario Fuest (Autor, Korrektur, Layout)
 Matthias Haupt (Korrektur)
 Peter Majmesku (Korrektur)
 Ralf Hersel (Autor, Korrektur)
 Robert Huckstorf (Korrektur)
 Stefan Zaun (Autor, Korrektur)
 Thomas Rudolph (Autor, Korrektur)

Lizenz

Yalm wird unter der [Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz](#) veröffentlicht.



Kurz: Yalm-Ausgaben oder einzelne Artikel dürfen kopiert, verbreitet und öffentlich zugänglich gemacht werden; die Inhalte dürfen abgewandelt und

bearbeitet werden. Voraussetzung hierfür ist, dass sowohl der Autor als auch Yalm genannt werden und die Weitergabe unter den gleichen Lizenzbedingungen erfolgt.

Redaktion und Homepage

Kontakt: redaktion@yalmagazine.org

Yalm-Homepage: <http://www.yalmagazine.org>

IRC-Channel: [freenode.irc.net / #yalmagazine](irc://freenode.net/#yalmagazine)

V.i.S.d.P.: Tobias Kündig

Sagenblickweg 6

CH-6030 Ebikon

tobias@yalmagazine.org

Bildquellen

Die Inhaber der Bildrechte werden in den Bildunterschriften oder in den Artikelinformationen genannt. Für den Fall, dass die Verwendung eines Bildes nicht zulässig oder gewünscht ist, bitten wir um eine kurze Information; wir werden es dann umgehend entfernen.

Das auf der Titelseite verwendete Symbol einer Festplatte entstammt dem »Phuzion-Icon-Pack«, die Verwendung im Rahmen dieser Ausgabe ist durch den Autor, [kyo-tux](#), gestattet.

*Yalm 03/2009 erscheint
am 15. März 2009*