

Themen dieser Ausgabe sind u. a.

dCore

Die leichtgewichtige Distribution dCore basiert auf Tiny Core und greift auf den Softwarebestand von Ubuntu bzw. Debian zu. Nach einer allgemeinen Einführung wird in dem Artikel eine schrittweise Anleitung für eine Testinstallation auf einem USB-Stick gegeben. ([weiterlesen](#))

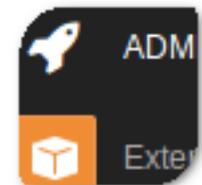
Seite 3



Einführung in Typo3

In dem Artikel wird die Geschichte von Typo3 und die Installation dieses Content Management Systems auf einem lokalen Rechner mit Linux Mint beschrieben. ([weiterlesen](#))

Seite 15



Review: Steam Controller

Mit dem Steam Controller möchte Valve seine Steamboxen mit Steam OS wohnzimmertauglich machen. Ob die außergewöhnlichen Bedienkonzepte auch unter Ubuntu funktionieren, soll in dem Artikel geklärt werden. ([weiterlesen](#))

Seite 26



Editorial

Ubuntu-Tablet

Was iOS und Android mit Universal Apps schon länger kennen, führte letztes Jahr Microsoft mit Continuum ebenfalls ein: Smartphone-Apps, die als Universal App entwickelt wurden, lassen sich mit angeschlossenem Monitor auch auf einem Desktop-PC darstellen und nutzen, wobei das Smartphone als zweites Display dient. Unter Ubuntu wird diese Funktion Convergence heißen und soll für Smartphones und Tablets verfügbar sein. Während der spanische Hersteller BQ schon letztes Jahr zwei Smartphones mit dem Ubuntu-Betriebssystem auf den Markt gebracht hat (siehe auch [freiesMagazin](#) 04/2015 [1] und 08/2015 [2] für ausführliche Reviews), wurde Anfang Februar das Aquaris M10-Tablet mit Ubuntu als Betriebssystem vorgestellt [3]. Auch das Tablet wird wieder von BQ hergestellt und soll im zweiten Quartal 2016 erscheinen. Convergence wird nicht auf Tablets beschränkt bleiben, sondern soll demnächst auch mit den Ubuntu Phones funktionieren.

Interview

Klaus Gürtler, Redakteur und Freier Journalist, hat ein Interview mit Dominik Wagenführ u.a., Chefredakteur von [freiesMagazin](#) geführt, um Einsteigern in die Linux-Welt ein paar Tipps an die Hand zu geben. Die Artikel wurden im Südkurier [4] und in der Saarbrücker Zeitung [5] veröffentlicht.

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

Ihre [freiesMagazin](#)-Redaktion

LINKS

- [1] <https://insights.ubuntu.com/2016/02/04/canonical-reinvents-the-personal-mobile-computing-experience/>
- [2] <http://www.freiesmagazin.de/freiesMagazin-2015-04>
- [3] <http://www.freiesmagazin.de/freiesMagazin-2015-08>
- [4] <http://techniknotizen.de/?p=489>
- [5] <http://www.suedkurier.de/nachrichten/digital/Digitale-Zukunft-mit-Pinguin;art1182640,8440597>

Inhalt

Linux allgemein

dCore	S. 3
Kurztipp: Unerwünschte grep-Ausgaben vermeiden	S. 11
Der Jahreswechsel im Kernelrückblick	S. 12

Anleitungen

Einführung in Typo3	S. 15
GitLab Continuous Integration	S. 21

Hardware

Review: Steam Controller	S. 26
--------------------------	-------

Community

Rezension: Professional Python	S. 32
Rezension: Open Source und Schule – Warum Bildung Offenheit braucht	S. 34
Rezension: Das Sketchnote-Arbeitsbuch	S. 35
Rezension: Linux-Server – Das umfassende Handbuch	S. 37

Magazin

Editorial	S. 2
Leserbriefe	S. 41
Veranstaltungen	S. 42
Vorschau	S. 42
Konventionen	S. 42
Impressum	S. 43



Teilen



Kommentieren

dCore von Stefan Müller

Die leichtgewichtige Distribution dCore basiert auf Tiny Core und greift auf den Softwarebestand von Ubuntu bzw. Debian zu. Nach einer allgemeinen Einführung wird hier eine Anleitung für eine Testinstallation auf einem USB-Stick gegeben.

Einführung

Distributionen, die im RAM laufen (vgl. Anhang „InitRD“), gibt es unter Linux in stattlicher Zahl. Zu den bekanntesten Vertretern dieser Zunft zählen Knoppix sowie die Live-Versionen von diversen Distributionen. Daneben gibt es eine ganze Schar speziell leichtgewichtiger Distributionen wie Puppy Linux oder Tiny Core Linux. Der jüngste Spross in dieser Familie hört auf den Namen dCore und stammt ebenfalls aus dem Hause Tiny Core.

Der Name ist ein Zusammenschluss aus Debian und Tiny Core und deutet auf das neue Konzept hin. Dabei wird das gesamte Grundsystem von dCore von Tiny Core übernommen und dem Benutzer somit eine minimalistische Linux-Umgebung zur Verfügung gestellt, die er nach eigenen Wünschen modular ausbauen kann. Wo Tiny Core jedoch eigene Paketquellen verwendet, greift dCore hingegen auf die Bestände von Debian oder Ubuntu zurück. Dadurch steht dem Benutzer ein außerordentlich schlankes Grundsystem mit einer überwältigenden Auswahl an aktueller Software zur Verfügung.

Für die x86-Architektur existiert dCore derzeit (November 2015) in den Debian- bzw. Ubuntu-basierten Ausprägungen Jessie und Wheezy beziehungsweise Trusty, Utopic, Vivid und Wily mit den entsprechenden Kernel-Versionen. Nach gut zweijähriger Entwicklung ist dCore den Kinderschuhen entwachsen und bietet eine höchst interessante Plattform für fortgeschrittene Linux-Benutzer.

Informationen

Die Installationsdateien [1] befinden sich etwas versteckt im Download-Bereich der Website von Tiny Core [2]. Im Tiny-Core-Wiki existiert ein separater Bereich für dCore [3]. Daneben wird das dCore-Forum [4] sehr aktiv genutzt.

Modularität

dCore lädt den Linux-Kern sowie eine InitRD, die das Wurzeldateisystem umfasst. Diese beiden Dateien sind zusammen weniger als 15 MB groß und stellen ein minimales Linux-System mit Almquist-Shell (Ash) und Busybox zur Verfügung. Zusätzliche Software-Pakete werden in Form von sogenannten Erweiterungen importiert und können nach Bedarf geladen werden. So kann sich der Benutzer sein System nach eigenen Wünschen zusammenstellen.

Erweiterungen

Erweiterungen sind SquashFS-Dateien, d. h. in eine Datei gepackte Dateisysteme, mit Dateien-

ung `.sce`. Darin sind alle zur Erweiterung gehörigen Dateien enthalten. Auf dem System vorhandene Erweiterungen werden durch den Befehl `sce-load` gemountet, der auch alle Dateien als symbolische Links ins Wurzeldateisystem kopiert. Somit stehen sie im normalen Pfad zur Verfügung. Nötigenfalls werden auch Startskripte ausgeführt. Danach kann die importierte Applikation normal gestartet werden.

Erweiterungen importieren

Zuerst müssen die Erweiterungen importiert werden. Der Befehl `sce-import` aktualisiert die Paketverzeichnisse und durchsucht sie nach passenden Namen. Man wählt das gewünschte Paket aus einer Trefferliste, worauf es samt Abhängigkeiten heruntergeladen und in eine sce-Datei umgewandelt wird.

Um Plattenspeicher zu sparen, lässt sich die Größe von Erweiterungen reduzieren, indem sie von anderen abhängig gemacht werden. Das Flag `-d` zu `sce-import` erlaubt die Auswahl von vorausgesetzten Abhängigkeiten. Einerseits werden deren Dateien beim Import ignoriert. Andererseits wird beim Laden der Erweiterung sichergestellt, dass alle Abhängigkeiten vorrangig geladen werden.

Erweiterungslisten

Sollen mehrere Erweiterungen stets zusammen geladen werden, bietet es sich an, sie zu einer ein-

zigen Erweiterung zu kombinieren. Mit dem Flag `-l` liest `sce-import` die Paketnamen aus einer Textdatei und erzeugt daraus eine Erweiterung.

Erweiterungen laden

Neben dem manuellen Laden können Erweiterungen auch beim Systemstart automatisch geladen werden, was für systemnahe Software wie zum Beispiel Wireless-Unterstützung oder Window-Manager beziehungsweise Desktop-Umgebungen sinnvoll ist. Andererseits lassen sich Programme auch bei Bedarf („on demand“) laden, was besonders in graphischen Umgebungen den Bedienungskomfort erhöht.

Persistenz

Das Grundsystem und auch alle Erweiterungen bestehen aus ausschließlich lesbaren Dateisystemen. Deshalb braucht es einen speziellen Mechanismus, um persistente Veränderungen bspw. an Konfigurationsdateien vorzunehmen. Im Gegensatz zu vielen anderen Live-Systemen setzt dCore, wie auch Tiny Core, nicht auf Overlay-Dateisysteme (vgl. Anhang „Persistenz“).

In der einfachsten Installationsvariante werden alle Benutzerdaten – und dazu zählen auch individuelle Systemeinstellungen – ins Archiv `/opt/mydata.tgz` gepackt. Die Datei `/opt/.filetool.lst` regelt, welche Dateien darin aufgenommen werden.

Konfigurationsdateien beispielsweise unter `/etc/` können durch Eintrag in diese Datei persistent



dCore mit LXDE. Das geöffnete Startmenü zeigt Rubriken für geladene Erweiterungen. Rechts im Panel sichtbar: NetworkManager, OwnCloud, Tastaturlayout, sowie Starter für Bereitschafts- und Ruhezustand sowie Herunterfahren. 🔍

gemacht werden. Die Originalversion wird beim Systemstart durch die in `mydata.tgz` gespeicherte überschrieben.

Neben dem Home-Verzeichnis wird in der Grundeinstellung auch `/opt/` persistent gemacht. Es enthält die internen Konfigurationsdateien für dCore. Dazu zählen Listen mit Paketquellen sowie die Dateien `bootlocal` und `bootsync`. Diese Shell-

Skripte werden im Verlauf des Systemstarts ausgeführt und können individuell angepasst werden.

Beispielinstallation

Um ein Gefühl für die Funktionsweise von dCore zu erhalten, wird hier eine LXDE Desktop-Umgebung mit aktuellen Versionen von LibreOffice und Firefox samt deutschsprachiger Lokalisierung installiert. Diese Applikationen werden

aus dem Software-Bestand von Ubuntu Trusty stammen.

USB-Stick vorbereiten

Man nehme einen leeren USB-Stick mit einer Ext-formatierten Partition `/dev/sdXn`, der boot-bar ist. Auf dieser erstellt man mit Root-Berechtigung einen Ordner namens `boot/`.

Falls kein Boot-Loader vorhanden ist, schreibt man einen solchen von einer bestehenden Linux-Installation aus auf den Stick (bevorzugt Extlinux, das an über die Paketverwaltung installiert).

Mit den folgenden Befehlen installiert man als Root den Boot-Loader samt benötigtem Hilfsmodul und passendem „master boot record“ auf dem Stick. Weiter werden der Linux-Kern sowie die InitRD heruntergeladen und überprüft.

```
# mount /dev/sdXn/ /mnt/
# mkdir /mnt/boot/
# extlinux --install /mnt/boot/
# cp /usr/lib/syslinux/menu.c32 /mnt/boot/
# dd if=/usr/lib/extlinux/mbr.bin of=/dev/sdX
# cd /mnt/boot/
# repo="http://tinycorelinux.net/dCore/x86/~
release/dCore-trusty"
# for file in dCore-trusty.gz vmlinuz-trusty ; do
#     wget $repo/$file
#     wget $repo/$file.md5.txt
#     cat $file.md5.txt >> md5.txt
# done
# md5sum -c md5.txt
# rm md5.txt
```

Im Ordner `boot/` erstellt man anschließend die Datei `extlinux.conf` wie im folgenden Listing `extlinux.conf` angegeben. Hierzu wird die UUID der Partition benötigt, welche sich mit dem Befehl `sudo blkid /dev/sdXn` herausfinden lässt.

```
DEFAULT menu.c32
PROMPT 0
LABEL dCore
    KERNEL vmlinuz-trusty
    INITRD dCore-trusty.gz
    APPEND waitusb=10 tce=UUID=<UUID> ~
home=UUID=<UUID> opt=UUID=<UUID> tz="CET~
-1CEST,M3.5.0,M10.5.0/3"
```

Listing 1: `extlinux.conf`

Für Extlinux von Version kleiner als 3.70 müssen die Dateinamen im Format DOS-8.3 angegeben werden, d. h. als `vmlinu~1` bzw. `dcCore~1.gz`. Je nach Reaktionsfreudigkeit des USB-Sticks kann die Sekundenzahl des Boot-Parameters `waitusb` auch verkürzt werden.

Erster Start

Vor dem ersten Start versichere man sich, dass keine andere Installation die Swap-Partition der Festplatte für den Ruhezustand benützt, denn diese Daten könnten überschrieben werden!

Das System bootet in eine Almquist-Shell (Ash), welche zusammen mit den

weiteren grundlegenden Kommandozeilenwerkzeugen durch BusyBox bereitgestellt wird. Die bildet mit InitRD das Wurzeldateisystem und liegt vollständig im RAM. Der einzige Benutzer heißt `tc` und kann mittels `sudo` Root-Berechtigung erlangen.

Besonders viel bietet das System noch nicht. Während des Startvorgangs wurden jedoch diverse Skripte ausgeführt, die u. a. die Boot-Parameter „tce“, „opt“ und „home“ dahingehend interpretierten, dass die entsprechenden Ordner auf der angegebenen Partition angelegt werden sollten.

Im Verzeichnis `/tce/` werden die Erweiterungen abgespeichert. Von der dCore-Installation aus ist dieses Verzeichnis unter `/etc/sysconfig/tcedir` verlinkt.

„Side-loading“ von Erweiterungen

Hat man eine funktionierende Ethernet-Schnittstelle zur Verfügung kann man nun gleich loslegen und Erweiterungen installieren. Andernfalls müssen zuerst geeignete Erweiterungen geladen werden, um die kabellose Netzwerkschnittstelle zu betreiben.

Hierfür muss das System zunächst heruntergefahren werden, damit die passenden Dateien auf den USB-Stick gebracht werden können. Die folgenden Befehle müssen genauso wie die ursprüngliche Installation mit Root-Rechten ausgeführt werden.

```
# mount /dev/sdXn/ /mnt/
# cd /mnt/tce/sce/
# repo="http://tinycorelinux.net/dCore/x86/~
release/dCore-trusty"
# for file in wireless-3.16.6-tinycore wireless ; do
do
#   for ext in sce sce.list sce.md5.txt ; do
#     wget $repo/sce/$file.$ext
#   done
#   cat $file.sce.md5.txt >> md5.txt
# done
# md5sum -c md5.txt
# rm md5.txt
```

Die erste Erweiterung liefert alle verfügbaren Treiber inklusive Firmware, die zweite Werkzeuge für den Umgang mit kabellosen Netzwerken. Zusätzlich stünde auch NdisWrapper zur Verfügung, falls keine Linux-Treiber für die Netzwerkkarte existieren sollten.

Wireless

Beim zweiten Start wird mit dem Befehl **sudo wifi.sh** eine Auswahlliste der erkannten WLAN-Netzwerke angezeigt, worauf eventuell benötigte Passwörter abgefragt werden. Diese werden in der Datei `~/Wifi.db` abgelegt und später von dort automatisch abgerufen.

X11-Desktop

Um eine einfache X11-Umgebung aufzuziehen werden folgende Erweiterungen benötigt: **graphics-3.16.6-tinycore**, **xorg-all** und **flwm**. Sie werden einzeln importiert. Die Erweiterung

graphics-3.16.6-tinycore wird von dCore zur Verfügung gestellt, die anderen von Ubuntu.

Der Befehl **sce-import** fragt nach dem Anfang des Paketnamens, worauf eine Auswahlliste aller passenden Pakete präsentiert wird. Nach der Auswahl werden alle Abhängigkeiten aufgelöst. Die Pakete werden heruntergeladen und zu einer Erweiterung mit Dateieindung **.sce** in `/etc/sysconfig/tcedir` zusammengefasst.

Für den Start des graphischen Desktops, muss die Boot-Loader-Konfiguration angepasst werden. Zu diesem Zweck wird mittels **sce-import nano** der Editor Nano importiert. Alternativ steht die BusyBox-Version von Vi zur Verfügung.

Diese Erweiterung wird mittels **sce-load nano** geladen. Danach kann mittels **sudo nano /mnt/sdXn/boot/extlinux.conf** der Boot-Code **desktop=flwm** zum Eintrag **append** hinzugefügt werden. Da sich erst später um die Internationalisierung gekümmert wird, muss mit der US-amerikanischen Tastaturbelegung gearbeitet werden – insbesondere muss für den Schrägstrich die Bindestrich-Taste gedrückt werden.

Die drei importierten Erweiterungen müssen beim Booten automatisch geladen werden. Hierzu werden sie aufgelistet in der Datei `/etc/sysconfig/tcedir/sceboot.lst`.

Nach einem Neustart zeigt sich ein graphischer Desktop mit dem minimalistischen Fenstermanager FLWM. Dessen Menü ist über einen Rechtsklick auf den Desktop zugänglich. Diese Umgebung ist sehr minimalistisch und wird hier nicht weiter betrachtet. Stattdessen soll die etwas leistungsfähigere Desktop-Umgebung LXDE installiert werden.

LXDE

Mittels Rechtsklick auf den Desktop kehrt man zur Kommandozeile zurück („Exit to Prompt“). Dort importiert man das Paket **lxde** und ersetzt in der Datei `/etc/sysconfig/tcedir/sceboot.lst` den Eintrag **flwm** durch **lxde**.

In `/mnt/sdXn/boot/extlinux.conf` verändert man den den Boot-Parameter zu **desktop=startlxde**. Nach einem Neustart erscheint daraufhin eine funkelneue LXDE-Arbeitsfläche.

Tastaturbelegung

LXDE stellt ein eigenes Verwaltungsprogramm für die Tastaturbelegung zur Verfügung. Mittels Rechtsklick auf die LXDE-Leiste fügt man „Keyboard Layout Handler“ als neues Panel-Applet hinzu.

Mit Rechtsklick auf das Flaggensymbol können daraufhin die Einstellungen angepasst werden. Bevor eine neue Belegung hinzugefügt werden kann, muss die Option „Keep system layouts“ deaktiviert werden.

Applikationen

Applikationen bestehen typischerweise aus einem Grund- sowie ergänzenden Paketen beispielsweise für die Internationalisierung. Diese werden zu einer einzigen Erweiterung zusammengefasst, so dass sie zusammen geladen werden.

Hierzu erstellt man zum Beispiel eine Liste namens **libreoffice.liste**:

```
libreoffice
libreoffice-help-de
libreoffice-l10n-de
```

Listing 2: *libreoffice.liste*

Mittels **sce-import -l libreoffice.liste** werden die zugehörigen Pakete zusammengesucht und zu einer Erweiterung gebündelt. Die Liste kann danach gelöscht werden, denn sie wurde in den Ordner **tce/** übernommen.

Mittels **sce-load libreoffice** wird die Erweiterung geladen und steht danach im Startmenü von LXDE zur Verfügung.

On demand

Eine rein graphische Art Applikationen zu starten, bietet der sogenannte „on demand“-Mechanismus. Mittels **ondemand libreoffice** wird ein Startskript angelegt, das mit einem Menüeintrag verbunden ist.

Leider wird LXDE jedoch nicht direkt unterstützt, weshalb das entsprechende Menü fehlt. dCore un-

terstützt diverse Fenstermanager, darunter Openbox. Da LXDE wiederum auf Openbox aufsetzt, lässt es sich mit etwas Handarbeit passend einrichten.

LXDE & Openbox

Zunächst ist es notwendig in der Konfigurationsdatei **/mnt/sdXn/boot/extlinux.conf** den Boot-Parameter **desktop=openbox** abzuändern. Damit wird die angesprochene Unterstützung durch dCore aktiviert.

Um trotzdem LXDE zu starten, wird in der Datei **~/xinitrc** die Zeile

```
$DESKTOP 2>/tmp/wm_errors &
```

geändert zu:

```
startlxde 2>/tmp/wm_errors &
```

Nach einem Neustart, der mit „*System Tools* → *Exit*“ eingeleitet wird, hat sich scheinbar nichts geändert. Nach wie vor strahlt einem ein LXDE-Desktop entgegen, und von einem Ondemand-Menü ist nichts zu sehen. Hierfür ist es zuerst einmal notwendig, das Openbox-Menü anzeigen zu lassen. Im Abschnitt „*Advanced*“ von „*Preferences* → *Desktop Preferences*“ kann man die Anzeige des Fenstermanagermenüs bei Rechtsklick auf den Desktop erlauben.

In der Standardeinstellung enthält dieses Menü nichts Neues. Weiter muss in der Datei

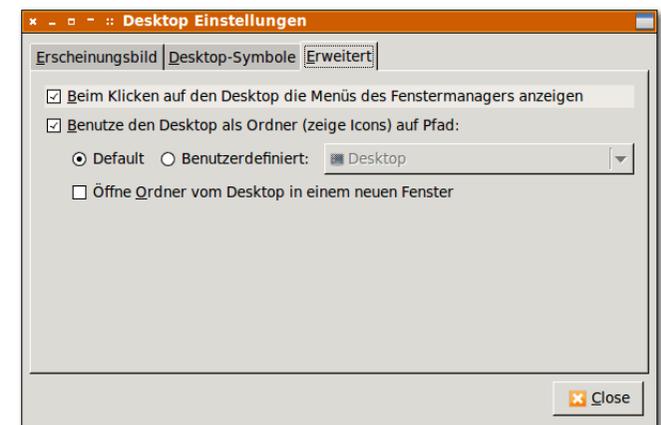
~/config/openbox/lxde-rc.xml die Zeile **<file>/usr/share/lxde/openbox/menu.xml</file>** durch die vier Zeilen des folgenden Listings ersetzt werden. Am schnellsten findet man diese Zeile durch Suche nach **<file>**.

```
<file>/usr/local/tce.openbox.xml</file>
<file>ondemand.xml</file>
<file>system-tools.xml</file>
<file>menu.xml</file>
```

Listing 3: *lxde-rc.xml*

Danach wählt man im Openbox-Menü „*Reconfigure Openbox*“, worauf das „on demand“-Menü mit dem Eintrag „*libreoffice*“ erscheint. Damit kann diese Erweiterung geladen und gestartet werden.

Um das Openbox-Menü stets zugänglich zu halten, lohnt es sich, an einem Rand des Desktops

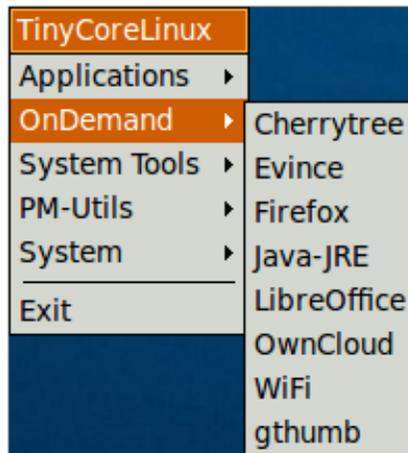


Das Openbox-Menü muss extra aktiviert werden, damit es bei Rechtsklick auf den Desktop angezeigt wird. 🔍

ein Pixel frei zu halten, sodass dort stets ein Rechtsklick auf den Desktop möglich ist. Diese Einstellung kann im Bereich „Margins“ des Openbox Konfigurationsmanagers getätigt werden.



Das Openbox-Menü zeigt den „korrekten“ Exit-Befehl, der für dCore benötigt wird. Insbesondere werden dabei auch die persistenten Dateien gespeichert. 🔍



Vom Untermenü „Ondemand“ aus können Erweiterungen geladen werden, damit sie nachher im LXDE-Startmenü verfügbar sind. 🔍

Individuelle Startskripte

Möchte man zum Beispiel den Browser Firefox nutzen, so ist man auf eine funktionierende Netzwerkverbindung angewiesen. Angenommen, man verwendet WLAN, so muss zuerst diese Verbindung aufgebaut werden, bevor Firefox genutzt werden kann. Somit müssen zwei Erweiterungen geladen und gestartet werden, was von Hand mit der Zeit mühselig erscheint. Mit dem „on demand“-Mechanismus lässt sich dies elegant vereinfachen.

Zunächst erstellt man die Liste `firefox.liste` und erstellt anschließend daraus mittels der Eingabe `sce-import -l firefox.liste` eine Erweiterung.

```
firefox
firefox-locale-de
```

Listing 4: `firefox.liste`

Mit dem Befehl `ondemand firefox` wird unter `/etc/sysconfig/tcedir/ondemand/` das Skript `firefox` angelegt. Dessen Inhalt wird angepasst wie im Listing `firefox` angegeben, um vor dem Start des Browsers die WLAN-Verbindung aufzubauen.

```
#!/bin/sh
sce-load wireless-3.16.6-tinycore
sce-load wireless
cliorex sudo wifi.sh
ondemand -e firefox.sce
```

Listing 5: `firefox`

Einschätzung

Nach Aussage von Hauptentwickler Jason Williams ist dCore „*a fast moving target*“. Mittlerweile bewegt es sich jedoch in recht ruhigen Schritten voran, sodass es als stabil bezeichnet werden darf. Neben dCore wird auch Tiny Core aktiv weiter entwickelt; gegenwärtig ist Version 6.4 aktuell (Stand November 2015).

dCore bietet ein außerordentlich stabiles, leichtgewichtiges System, das nach eigenen Wünschen zusammengestellt und eingerichtet werden kann. Der Kompromiss zwischen Geschwindigkeit und Komfort fällt sehr positiv ins Gewicht. Im Gegensatz zu den großen Distributionen laufen nur diejenigen Programme, die man wirklich benötigt, und im Gegensatz zu anderen leichtgewichtigen Distributionen hat man Zugriff auf dasselbe Angebot an Applikationen wie bei einer großen Distribution.

Upgrade

Das dCore-Basissystem besteht zur Hauptsache aus dem Linux-Kern sowie den Startskripten. Die verschiedenen Versionen unterscheiden sich praktisch nicht in der Installation, und ein nachträgliches Upgrade von Kernel und InitRD ist problemlos. Beim Software-Stand stehen einem jedoch aktuellere Applikationen zur Verfügung.

So gesehen gibt es keinen Grund, die Beispielinstallation auf Ubuntu Trusty ruhen zu lassen. Es sei dem geneigten Leser als Übungsaufgabe überlassen, selber Hand anzulegen. Nach erfolg-

tem Austausch von Kernel und InitRD werden die Applikationen mit **sce-update** aktualisiert.

dCore eignet sich sehr gut zur Installation auf älteren Maschinen. Es ist eine wahre Freude zu sehen, wie flüssig die Arbeit auch auf „ehrenwerten“ Geräten vonstatten geht. Für den alltäglichen Gebrauch wird man die Installation auf einer Festplatte vornehmen. Das Vorgehen ist identisch mit dem hier vorgestellten und ist im Wiki [3] ausführlich beschrieben.

dCore mit LXDE

Der hier behandelte LXDE-Desktop zeigt, was man diesbezüglich erwarten kann. Da es sich um eine Beispielininstallation handelt, wurden etliche Punkte weggelassen, die u. a. im dCore-Wiki [3] detailliert besprochen werden:

- Bereitschafts- oder Ruhezustand
- NetworkManager zur komfortableren Verwaltung der Netzwerkverbindungen
- Angepasste Konfiguration des LXDE-Desktops (insbesondere ein Exit-Button anstatt desjenigen von LXDE, der nicht funktioniert)
- Verwendung weiterer Paketquellen inkl. PPAs
- Java und Flash im Browser
- OwnCloud-Synchronisation

Eine solche Installation entspricht ungefähr einer aktuellen Ubuntu-Installation. Im Gegenteil zu einer solchen lässt sie aber auch ein betagtes Netbook noch respektable Sprünge vollführen. Dank der Modularität von dCore lässt sich der

Software-Stand einer aktuellen Ubuntu-Version erreichen, ohne dass man ein derart schweres System installieren müsste.

dCore mit Window-Manager

Bei der hier vorgestellten Installation handelt es sich um die „Luxusversion“ eines leichtgewichtigen Systems. Anstatt LXDE können selbstverständlich einfachere Window-Manager eingesetzt werden, um das System zu entschlacken und zu optimieren.

Dabei gilt es zu beachten, dass der „on demand“-Mechanismus nur für die folgenden Window-Manager eingerichtet ist: Fluxbox, FLWM, Hackedbox, Icewm, Jwm-snapshot, Jwm und Openbox. Diese Einschränkung stammt von Tiny Core, von dem dCore diese Funktionalität übernimmt.

Standard ist FLWM [5] mit Wbar als Panel, zu dem falls nötig ein Dock hinzugefügt werden kann.

Einschränkung

Das Grundkonzept, einer minimalen Basisinstallation Zugriff auf die riesigen Bestände der Ubuntu-beziehungsweise Debian-Quellen zu geben, entkoppelt die Software vom jeweiligen Grundsystem. Anstatt eines vorkonfigurierten Wolkenkratzers mit Betonsockel baut man sich sein eigenes Mobil-Home, indem man einzelne Wohneinheiten auf einen Laster auflädt. Wie gesehen gewinnt man dabei an Flexibilität. Die Modularität jedoch geht auf Kosten des Kitts, der die Einheiten mit den

Sockel verbindet. Konkret geht es um die Start- und Installationskripte, die teilweise angepasst werden müssen, bevor sie auf den dCore-eigenen Repositorien verfügbar sind. Dies betrifft jedoch längst nicht alle Applikationen, sodass sich der Aufwand für Entwickler und Community in Grenzen hält.

Weitere Informationsquellen

Die Readme-Seiten [6] zur Installation geben konkrete Zusatzhinweise.

Das Wiki [3] enthält weitere Beispielininstallationen sowie ausführliche Informationen zur Arbeit mit Erweiterungen.

Für direkte Anfragen eignet sich das dCore-Forum [4] besonders gut. Man kann mit einer umgehenden Antwort der beteiligten Entwickler rechnen.

Als technische Referenz eignet sich das Core-Book [7] besonders für die mannigfaltigen Boot-Codes. Zwar bezieht es sich auf Tiny Core, dessen Basisinstallation jedoch quasi baugleich mit derjenigen von dCore ist.

Anhang

InitRD

Die meisten „regulären“ Distributionen verwenden eine initiale RAM-Disk (InitRD), um dem Linux-Kern in der Startphase zusätzliche Treibermodule zur Verfügung zu stellen. Damit kann auch ein generischer Kern das Wurzeldateisystem beispiels-

weise auch von LVM, RAID oder verschlüsselten Partitionen laden.

Die InitRD umfasst ein gepacktes Dateisystem, das vom Kern im RAM entpackt und eingehängt wird. Daraufhin wird der darin enthaltene Init-Prozess gestartet. Üblicherweise hängt dieser ein beschreibbares Wurzeldateisystem von einem Datenträger ein, und lässt sich durch den eigentlichen Init-Prozess ablösen.

Ein System, das im RAM läuft, verlässt die InitRD jedoch nicht und belässt das Wurzeldateisystem im RAM. Damit gewinnt man Geschwindigkeit, weil die vielen Systemdateien nicht einzeln von einem Datenträger geladen werden müssen. Andererseits können die Dateien nicht verändert werden, weil die InitRD grundsätzlich nur lesbar ist.

Letzteres bringt zunächst die Einschränkungen mit sich, dass Konfigurationsdateien, zum Beispiel unter `/etc/` nicht dauerhaft verändert werden können. Prinzipiell sind solche Systeme dadurch sozusagen „unkaputtbar“, weil das Grundsystem nach jedem Neustart wieder in den Ursprungszustand zurückkehrt. Methoden, um trotzdem Dateien dauerhaft im System zu speichern, werden unten diskutiert.

Zuletzt muss noch erwähnt werden, dass die Größe der InitRD natürlich gering gehalten werden sollte, damit das Entpacken in annehmbarer Zeit geschieht. Deshalb wird das Grundsystem mini-

mal gehalten, was sich zum Beispiel darin zeigt, dass oftmals Busybox anstelle der GNU Coreutils zum Einsatz kommt. Die InitRD von dCore ist ca. 10 MB groß.

Persistenz

Ein Live-System kann mit verschiedenen Methoden um ein persistentes Dateisystem ergänzt werden. Am gebräuchlichsten ist ein beschreibbares Overlay-Dateisystem [8]. Ein solches kann transparent über das Wurzeldateisystem gelegt werden. Geänderte oder gelöschte Dateien werden punktuell überdeckt, sodass im Gesamten der Eindruck eines beschreibbaren Wurzeldateisystems entsteht.

dCore beschreitet einen anderen Weg. Da alle Dateien in einem RAM-Dateisystem liegen, können sie im Betrieb direkt verändert werden. Um Dateien oder Ordner persistent zu machen, listet der Benutzer ihren Namen (ohne führenden Schrägstrich!) in der Datei `/opt/.filetool.lst`. Beim Herunterfahren werden diese Dateien ins Archiv `/opt/mydata.tgz` gepackt – sofern vor dem eigentlichen Shutdown-Kommando das Skript `backup.sh` aufgerufen wird. Beim grafischen Abmeldedialog ist das der Fall, auf der Kommandozeile muss das Skript jedoch manuell aufgerufen werden.

Beim Hochfahren werden die Dateien wieder ins RAM-Dateisystem geschrieben und stehen so zur Verfügung. In der Grundeinstellung werden `/home/` und `/opt/` persistent gemacht. Letzteres

enthält insbesondere alle importierten Erweiterungen, weshalb beide Verzeichnisse recht umfangreich sein können.

Da das Ver- und Entpacken eines tar-Archivs recht zeitaufwändig ist, wird man beide Verzeichnisse gerne von einem herkömmlichen Dateisystem einbinden wie in der besprochenen Beispielinstallation.

LINKS

- [1] <http://tinycorelinux.net/dCore/x86/release/> 
- [2] <http://tinycorelinux.net/downloads.html> 
- [3] <http://wiki.tinycorelinux.net/dcore:welcome> 
- [4] <http://forum.tinycorelinux.net/index.php/board,66.0.html> 
- [5] <https://en.wikipedia.org/wiki/FLWM>
- [6] <http://tinycorelinux.net/dCore/x86/README/> 
- [7] <http://distro.ibiblio.org/tinycorelinux/corebook.pdf> 
- [8] https://en.wikipedia.org/wiki/Category:Union_file_systems 

Autoreninformation

Stefan Müller verwendet gerne Linux-Installationen auf Tablet PCs. Beruflich versucht er jungen Menschen die Freiheit in der (Informatik-)Welt näher zu bringen.

 [Teilen](#)

 [Kommentieren](#)

Kurztipp: Unerwünschte grep-Ausgaben vermeiden von Hans-Joachim Baader

Selbst über bekannte Programme lässt sich oft noch einiges lernen. Man muss sich nur gründlich mit ihnen auseinandersetzen. Dieser Artikel zeigt wie die eingebauten Hilfsmittel von **grep** dabei helfen unerwünschte Ausgaben zu vermeiden.

Redaktioneller Hinweis: Der Artikel „Unerwünschte grep-Ausgaben vermeiden“ erschien erstmals bei Pro-Linux [1].

Die folgende Situation ist vielen bekannt. Auf die Schnelle möchte man wissen, ob eine bestimmte Zeichenfolge (z.B. ein Name) in einer Datei (ggf. auch in welcher) im Verzeichnis **/etc/** vorkommt. Natürlich benutzt man dafür keinen Dateimanager und erst recht keine Desktop-Suche, sondern **grep**. Die Suche ist so viel einfacher und schneller, aber das Ergebnis versteckt zwischen einem Wust von Ausgaben nach dem Muster:

```
$ grep samba /etc/*
grep: /etc/acpi: Ist ein Verzeichnis
...
```

Wie nicht anders zu erwarten, gibt es eine Reihe von Bordmitteln, um die unerwünschten Ausgaben zu unterdrücken, die den Blick aufs Wesentliche verstellen. Zum Beispiel eine Kommandooption von **grep** oder eine Shell-Funktionalität.

```
$ grep -s samba /etc/*
```

```
$ grep samba /etc/* 2> /dev/null
$ grep -d skip -D skip samba /etc/*
```

Die drei Kommandos liefern, abhängig von der **grep**- bzw. Shell-Version, eine ähnliche Ausgabe. Je nach Implementierung (selbstgeschrieben, traditionelle Unix- oder BSD-Variante) treten Unterschiede auf. Am portabelsten ist das Konstrukt **2>**, welches die **grep**-Warnungen in den virtuellen Schredder umleitet.

Gemäß Handbuch ist die Option **-s** POSIX-konform und wird somit auf entsprechenden Systemen unterstützt. Kein Glück hat man dagegen bei alten **grep**-Versionen. Diese kennen die Option gar nicht. Bei wiederum Anderen wirkt sie wie **-q** (umgekehrt des erwünschten Verhaltens). Die Optionen **-d** und **-D** gibt es dagegen nur bei GNU Grep. (Es könnten noch weitere Fälle auftreten, bei denen **-s** schweigt, die anderen jedoch nicht.)

Auf GNU/Linux-Systemen ist es in der Regel egal, welche Variante man wählt. Die Verzeichnismeldung wird übrigens auch unterdrückt, wenn man die Option **-R/-r** zum rekursiven Suchen (mit bzw. ohne Folgen von symbolischen Links) verwendet.

Bequemer kann die Arbeit werden, wenn die häufig verwendete Art des Aufrufs mit einer speziellen Shell-Funktion umgesetzt ist. Diese ist im eigenen Profil (z.B. **~/ .bashrc**) zu speichern, um möglichst wenig tippen zu müssen.

```
$ lg() { grep -s "$@"; } # Funktion
$ lg samba /etc/*      # Aufruf
```

Und rekursiv wäre das Folgende denkbar:

```
$ rg() { grep -R -s "$@"; }
$ rg samba /etc/*
```

In beiden Fällen werden alle angegebenen Parameter mittels **\$@** an **grep** übergeben, sodass nichts an Flexibilität verloren geht. Der Vollständigkeit halber soll auch noch erwähnt werden, dass man Optionen an GNU Grep mit der Umgebungsvariable **GREP_OPTIONS** übergeben kann. In einem Skript sähe dies so aus:

```
$ export GREP_OPTIONS='-d skip -D skip'
$ grep samba /etc/*
```

LINKS

[1] <http://www.pro-linux.de/kurztipps/2/1809/unerwuenschte-grep-ausgaben-vermeiden.html>

Autoreninformation

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

arbeitet seit 1993 mit Linux und ist einer der Betreiber von Pro-Linux.de.

 [Teilen](#)

 [Kommentieren](#)

Der Jahreswechsel im Kernelrückblick von Mathias Menzer

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

Linux 4.4 entwickelt sich

Zum Nikolaustag wurde die vierte Entwicklerversion von Linux 4.4 vorgelegt [1]. Der Löwenanteil des eingebrachten Codes entfiel auf einen neuen Ethernet-Treiber. Dieser soll NB8800-Controller von Aurora VLSI ansteuern und wurde auf Basis der Treiber für Embedded-Prozessoren der Firma Sigma Designs entwickelt, welche auch über einen eingebauten Netzwerk-Controller verfügen.

Linux 4.4-rc5 [2] kam wieder mit etwas weniger Änderungen daher, von denen das Meiste auf Korrekturen entfiel. Davon konnte unter anderem der dwc2-Treiber profitieren, dem ein Kernel Oops, also ein Fehler des Kernels, der nicht zum vollständigen Systemabsturz führt, behoben wurde. Dieser Treiber unterstützt einen USB-On-the-Go-Controller und ermöglicht es, dass USB-Geräte andere USB-Geräte ansteuern können, wie beispielsweise ein Smartphone eine per USB-OTG angeschlossene Tastatur verwenden kann.

Auch die kurz vor Weihnachten erschienene sechste Entwicklerversion [3] wies keine größeren

Änderungen auf, auch wenn deren Gesamtzahl wieder ein wenig angestiegen war. Am Treiber für die Netzwerk-Virtualisierung VXLAN [4] wurde die Berechnung von Netzwerk-Routen für IPv6-Pakete in eine eigene Funktion verlagert wodurch künftige Optimierungen einfacher umsetzbar sind. Unter den Korrekturen fand sich auch eine, die das Verhalten der WLAN-Infrastruktur beim Aufwachen aus dem Suspend korrigiert. War das System beim „Einschlafen“ gerade mit der Suche nach neuen WLAN-Netzen beschäftigt, so brach dieser ab und das System wartete vergeblich auf dessen Vollendung. Künftig wird nun geprüft, ob ein solcher Zustand vorliegt und der Scan-Auftrag dann gegebenenfalls gelöscht.

Die beiden folgenden Entwicklerversion fielen dann ziemlich klein (-rc7) [5] und extra klein (-rc8) [6] aus – kein Wunder, auch manche Kernel-Entwickler machen über Weihnachten und den Jahreswechsel Urlaub. So hielten sich auch die Korrekturen in einem sehr überschaubaren Rahmen. Unter den hervorstechenden fand sich beispielsweise die Behebung eines Fehlers bei der Speicherbehandlung der Prozessor-eigenen Funktionen der Sparc64-Architektur, indem der Kernel neue Funktionen erhielt, um diese Fehler erkennen und umgehen zu können.

Linux 4.4 [7] wurde eine Woche später freigegeben. Auch hier hielten sich die Änderungen gegenüber der letzten Entwicklerversion in Gren-

zen. Nutzer künftiger Android-Versionen können sich über die Korrektur eines Fehlers im Umgang mit SYSENER freuen. Diese Funktion dient bei x84-Systemen dazu, Funktionsaufrufe in einem geschützten Betriebsmodus (Protected Mode) ohne aufwändige Prüfungen der Rechte des aufrufenden Prozesses durchzuführen und kann somit schneller erfolgen als ein direkter Funktionsaufruf. Ob die korrekte Nutzung von SYSENER die Schwuppdizität auf Systemen mit beschränkter Leistung (sprich: Smartphones) dann tatsächlich spürbar erhöht, ist allerdings fraglich.

Die Veröffentlichung von Linux 4.4

Seit Veröffentlichung von Linux 4.3 Anfang November waren 69 Tage vergangen und knapp über 14.000 Änderungen haben die Linux-Entwickler beige-steuert. Damit zählt der neue Kernel zu den größeren der letzten fünf Jahre. Die meisten der eingebrachten Änderungen zielen auch diesmal wieder darauf, Fehler zu beheben und doch sind ein paar bemerkenswerte Neuerungen mit dabei, wenn auch die meisten davon für den Nutzer eher verborgen bleiben.

„Kopfloser“ Speicher

So wurde nun mit LightNVM [8] die Unterstützung für Open-Channel SSDs aufgenommen. Diese verfügen über keinen vollwertigen Controller für die enthaltenen Flash-Speicherchips sondern überlassen die Speicherzuweisung dem Betriebssystem. Dadurch lassen sich zum einen ge-

nerische Treiber entwickeln, die nicht auf unterschiedliche Controller Rücksicht nehmen müssen, aber es sind auch Optimierungen der Schreib- und Lesevorgänge möglich und letztlich könnten hier solche Dinge wie DAX (Direct Access, siehe „Der April im Kernelrückblick“, [freiesMagazin 05/2015 \[9\]](#)) einfacher umgesetzt werden, die direkten Zugriff auf den Flash-Speicher des Datenträgers erwarten. Letztlich ist es auch nicht allzu weit hergeholt, die Verwaltung von Flash-Speichern dem Betriebssystem zu überlassen, haben dessen Entwickler mit Arbeitsspeicher doch schon Erfahrung.

Schnellerer Aufbau

Verbesserungen am Netzwerkstack hatten direkte Auswirkungen auf den Aufbau von Netzwerkverbindungen. Die Umsetzung des TCP-Protokolls kommt nun ohne Locking für die Bearbeitung eingehender Verbindungen aus. Das bedeutet, dass keine Sperren auf Ressourcen des Kernels gesetzt werden müssen, wobei gegebenenfalls auf die Freigabe einer zuvor von einem anderen Prozess gesetzten Sperren gewartet werden muss. Diese Wartezyklen fallen nun weg, wodurch sich die Menge der Verbindungsanfragen, die nun verarbeitet werden kann, erheblich angestiegen ist. Davon profitieren in erster Linie die Betreiber von Serverdiensten, da nun Verbindungen (etwas) schneller und mit weniger Belastung des Systems aufgebaut werden können. Ob dann natürlich die dahinterstehende Anwendung die Anfragen ebenso schnell bedienen kann, ist eine andere Geschichte.

Erweiterer eBPF

Unter Paketfilter verstehen zumindest Netzwerkaffine Menschen eine der Kernkompetenzen von Firewalls. Im Gegensatz dazu stellt der Berkeley Paket Filter (BPF [10]) beziehungsweise dessen Linux-Umsetzung eBPF eine Möglichkeit zur Fehlersuche innerhalb eines Systems dar. eBPF erlaubt es in den Netzwerk-Stack eines Linux-Systems zu greifen und dort Datenpakete zu analysieren. Um nicht allen Netzwerkverkehr an die anfordernde Anwendung zurückliefern zu müssen, bietet eBPF umfangreiche Filterfunktionen, die als Programme geschrieben und mittels eines eigenen Interpreters [11] ausgeführt werden.

Unter Linux 4.4 wurden nun zwei große Verbesserungen eingeführt. Zum einen können nun auch Programme, die im Benutzerkontext laufen, auf eBPF zugreifen. Und zum anderen können nun in eBPF erzeugte Filter dauerhaft abgelegt werden und bleiben damit auch nach Beendigung des zugehörigen Prozesses verfügbar.

Und auch die Betreuer anderer Komponenten des Kernels haben eBPF als nützliches Tool erkannt. Deshalb haben die Entwickler der Messwerkzeug-Sammlung „perf“ nun Unterstützung für eBPF eingebaut. Wird perf künftig eine C-Quellcode-Datei übergeben, die eBPF-Programmcode enthält, so wird dieser automatisch greift.

Virtuelles 3-D und himbeerige Grafik

Die Kernel-eigene Infrastruktur für den Betrieb virtueller Systeme wurde um die Unterstützung

für 3-D-Beschleunigung erweitert. Dadurch können nun mittels QEMU [12] virtualisierte Systeme ebenfalls in den Genuss eventuell vorhandener 3-D-Kapazitäten der Grafikhardware des Host-Systems kommen. Nutzer, die beispielsweise ISO-Dateien mittels Ubuntu Testdrive [13] ausprobieren, könnten hiervon in Zukunft profitieren.

Nicht vergessen werden sollte der neue Broadcom VC4-Treiber, welcher nun von Haus aus die Grafikeinheit der BCM2835-Prozessoren unterstützt. Den meisten Bastlern dürften diese ARM-Prozessoren aus dem Raspberry Pi 1 und Raspberry Pi Zero sowie dem Compute Module geläufig sein. VC4 erlaubt jedoch nur die Grundfunktionen, 3-D-Beschleunigung ist außen vor. Dennoch werden Systeme auf entsprechenden Prozessoren davon profitieren, da auf zusätzlich einzubindende Treiber verzichtet werden kann. Eine Erweiterung um 3-D-Unterstützung war für Linux 4.5 vorgesehen, wurde aber bis zum jetzigen Zeitpunkt noch nicht aufgenommen.

Eine vollständige Übersicht mit Verlinkungen zu den entsprechenden Commits oder Mailing-List-Beiträgen liefert hierzu auch die englischsprachige Seite [LinuxChanges.org \[14\]](#).

Die Entwicklung von Linux 4.5 startet

Mit der Freigabe von Linux 4.4 startete auch wieder der Entwicklungszyklus des Nachfolgers mit dem Merge Window. In diesem Zeitraum nimmt Torvalds Änderungen in seinen Entwicklungszweig auf, die ihm Entwickler oder die Betreuer

der verschiedenen Bereich des Kernels vorlegen. Einige Entwickler hatten bereits im Vorfeld ihre Merge Requests gestellt, sodass die Zahl der eingebrachten Änderungen rasch anwuchs. Mit Freigabe der ersten Entwicklerversion [15] nach 14 Tagen kamen so 11.000 Änderungen zusammen und platzierte Linux 4.5 damit – rein auf diesen Aspekt bezogen – im Mittelfeld.

Anders sieht es bei den dadurch geänderten Quelltext-Zeilen aus. Nur wenige Kernel-Versionen der letzten Zeit überschritten einen Wert von 1 Million geänderter Zeilen, 4.5-rc1 erreicht sogar 1,9 Millionen. Allerdings liegt der Hauptgrund hierfür wieder einmal im Bestreben der Entwickler, Ordnung in die Struktur des Kernel-Codes zu bringen und diesmal waren die Netzwerk-Treiber dran. Ein großer Teil der WLAN-Treiber wurde umsortiert, sodass die einzelnen Treiber sich nun einfacher dem jeweiligen Hersteller zuordnen lassen. Innerhalb der Versionsverwaltung Git [16] lässt sich dies bequem mittels rename, also einer Umbenennung der Datei und des Dateipfades abbilden, sobald allerdings ein Patch erstellt wird, wie dies auf Kernel.org [17] geschieht werden diese Dateien zuerst an einer Stelle entfernt und danach an anderer Stelle eingefügt. Sie schlagen sich somit sowohl in der Zahl der gelöschten als auch der aufgenommenen Codezeilen nieder und landen damit gleich zweimal im Patch, was diesen dann entsprechend aufbläht. Lange Rede kurzer Sinn: Es wurden sehr viele Codezeilen angefasst, doch mit verhältnismäßig wenig Wirkung dahinter.

Immerhin einige Neuerungen kamen hinzu, so soll PowerPlay [18] künftig für eine effizientere Nutzung von ATI/AMD-Grafik-Prozessoren sorgen und ARM-Freunde können künftig generische Kernel für ARMv6- und ARMv7-Plattformen bauen, die jeweiligen Eigenheiten der Hardware werden dem Kernel dann über Device Tree [19], eine Datenstruktur für die Beschreibung verfügbarer Hardware, bekannt gemacht. Und wer plant, auf einem Surface Pro 4 [20] Windows gegen Linux auszutauschen, wird mit Linux 4.5 auch Unterstützung für die (wenigen) Hardware-Buttons erhalten.

Linux 4.5-rc2 [21] liegt ein wenig über dem Durchschnitt, was die Zahl der Änderungen eines -rc2 betrifft. Immerhin lief der größte Teil unter der Kategorie Korrekturen, wobei insbesondere die Kernel-eigenen Werkzeuge für die Leistungsmessung profitieren konnten. Diese „perf-tools“ können Kernel- und Anwendungsentwicklern bei der Optimierung ihres Codes dienen, indem sie die Auswirkungen von Modulen und Funktionen auf den Ressourcenbedarf des Kernels messen. Hinzugekommen sind dagegen ein paar Testwerkzeuge für virtio, einen Treiber aus dem Virtualisierungsumfeld um QEMU und KVM [22].

LINKS

- [1] <https://lkml.org/lkml/2015/12/6/137> 
- [2] <https://lkml.org/lkml/2015/12/13/228> 
- [3] <https://lkml.org/lkml/2015/12/20/193> 
- [4] https://de.wikipedia.org/wiki/Virtual_Extensible_LAN
- [5] <https://lkml.org/lkml/2015/12/27/215> 

- [6] <https://lkml.org/lkml/2016/1/3/261> 
- [7] <https://lkml.org/lkml/2016/1/10/305> 
- [8] <http://lightnvm.io/> 
- [9] <http://www.freiesmagazin.de/freiesMagazin-2015-05>
- [10] https://de.wikipedia.org/wiki/Berkeley_Paket_Filter
- [11] <https://de.wikipedia.org/wiki/Interpreter>
- [12] <http://wiki.qemu.org/> 
- [13] <https://wiki.ubuntuusers.de/TestDrive/>
- [14] http://kernelnewbies.org/Linux_4.4 
- [15] <https://lkml.org/lkml/2016/1/24/196> 
- [16] <https://de.wikipedia.org/wiki/Git>
- [17] <https://kernel.org/> 
- [18] https://de.wikipedia.org/wiki/ATI_PowerPlay
- [19] https://en.wikipedia.org/wiki/Device_tree 
- [20] https://de.wikipedia.org/wiki/Microsoft_Surface#Vierte_Generation
- [21] <https://lkml.org/lkml/2016/1/31/231> 
- [22] https://de.wikipedia.org/wiki/Kernel-based_Virtual_Machine

Autoreninformation

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



Teilen



Kommentieren

Einführung in Typo3 von Martin Stock

In dem nachfolgenden Artikel wird die Geschichte von Typo3 [1] und die Installation dieses Content Management Systems auf einem lokalen Rechner mit Linux Mint beschrieben.

Content Management Systeme

Ein Content Management System (CMS) ist eine Software, die online auf einem Webspace installiert wird. Mit solch einem System kann man Inhalte erstellen und bearbeiten, um diese dann auf Webseiten online zu stellen. Benutzer können nach dem Anmelden am System in ihren zugehörigen Bereichen arbeiten oder alle Bereiche nutzen. Das liegt an den Zugriffsrechten, die intern vergeben werden können.

Die Bearbeitung von Inhalten geht genauso einfach wie eine E-Mail zu schreiben. Online-Texteditoren unterstützen den Benutzer bei dieser Arbeit. Mit einem CMS kann man auch ohne HTML-Kenntnisse Webseiten erstellen, die modern und barrierefrei gestaltet sind.

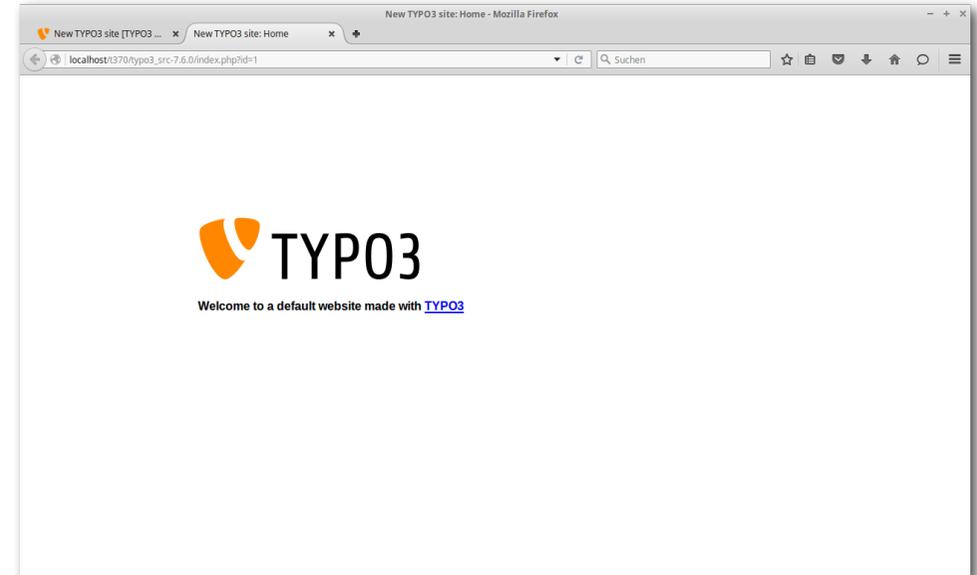
Historisches

Typo3 CMS ist ein von Kasper Skårhøj im Jahr 1997 entwickeltes Open Source CMS, das unter der GNU/GPL Lizenz veröffentlicht wird. Das CMS ist gerade in Deutschland ein sehr beliebtes Werkzeug, das nicht nur bei Unternehmen und politischen Parteien eingesetzt wird, sondern auch für Vereine und Organisationen geeignet ist.

Im Jahr 2000 kam die erste Beta-Version heraus und 2001 wurde dem System der Open-Source-Standard eingeräumt [2]. In den darauffolgenden Jahren gab es immer mehr Meetings, Code-Sprints, Entwicklertreffen und unterstützende Artikel in diversen Fachmagazinen, die dem System zu dem gemacht haben, was es heute ist – ein Content Management System mit Enterprise-Charakter im Open-Source-Bereich!

Finanzielle Unterstützung bekommt das System von der im Jahre 2004 gegründeten Typo3 Association aus der Schweiz. Das ist eine nicht-profitorientierte Organisation, welche sich durch Mitgliedsbeiträge, Spenden und der jährlich stattfindenden Konferenz finanziert. Jeder Nutzer und Freund von Typo3 kann natürlich unabhängig von dieser Institution jederzeit für das Projekt spenden.

Seit 2012 ist Typo3 eine geschützte Marke. Darunter sind weitere Produkte wie zum Beispiel Flow und Neos entwickelt worden.

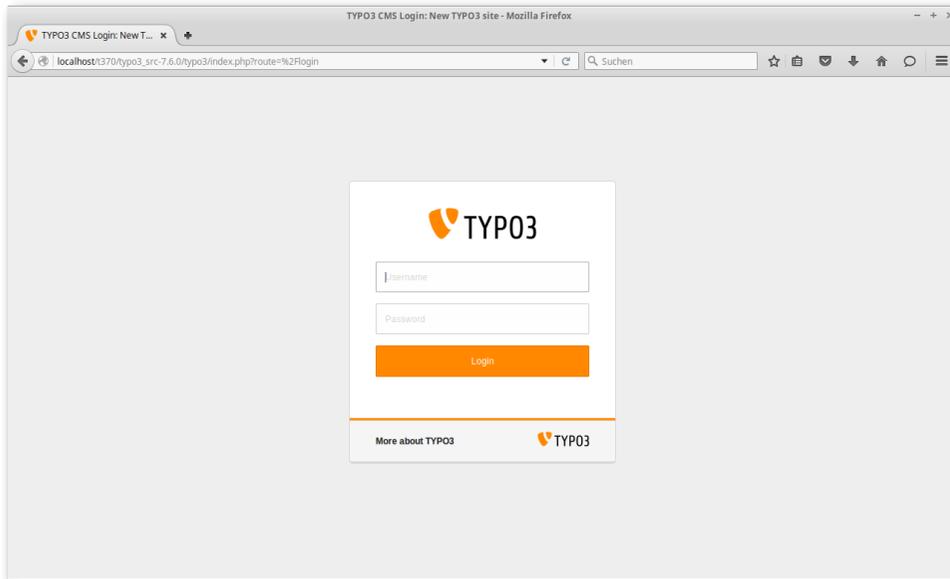


Beispielseite von Typo3 CMS. 

Einsatz von Typo3

Typo3 CMS genießt weltweit einen sehr guten Ruf. Die Community ist online sehr aktiv und unterstützt sich gegenseitig. Verschiedene Agenturen und freie Entwickler steuern zusätzlich zur allgemeinen Hilfe auch Erweiterungen bei, die dem System Leben einhauchen und kostenlos zur Verfügung stehen. Man geht davon aus, dass das System bisher über 500.000 Mal zum Einsatz gekommen ist [3].

Das System hat hohe Ansprüche an den verwendeten Webserver. Nicht alle Anbieter von Webservern können Typo3-Webseiten hosten. Bevor man



Login-Bereich zum Backend von Typo3 CMS 7.6.0. 

sich für einen Anbieter entscheidet, sollte man diesen genau fragen, ob es möglich ist, ohne Einschränkungen Typo3 auf deren Webspace zu nutzen. Allerdings kann man das System auch lokal auf dem heimischen Computer in einem bereits installierten Webserver testen.

Aufbau von Typo3

Das CMS basiert auf PHP. Die Dateien werden auf einem Webserver abgelegt. Inhalte, die auf den Webseiten angezeigt werden sollen, werden in Datenbanken gespeichert.

Typo3 CMS ist in zwei Hauptteile aufgeteilt. Das sogenannte Backend ist der Bereich, in dem sich der Administrator oder Redakteur anmeldet, um

Ausblenden von Inhalten, wie zum Beispiel Statistiken, und vieles mehr. Mit den sogenannten Erweiterungen werden z. B. Gästebücher, Nachrichtenbereiche, Community-Funktionen, Newsletter oder Foren erstellt. Es sind einem fast keine Grenzen gesetzt, bei dem was man machen und darstellen möchte. Von einfachen bis hin zu komplexen Inhalten ist dieses CMS perfekt für die Darstellung von Webseiten geeignet.

Die Inhalte der Webseite können in Typo3 getrennt vom Design erstellt werden. Wie bei fast allen modernen Systemen kann das Design angepasst werden, ohne den Inhalt zu verändern. Jede einzelne Seite kann ein eigenständiges Template erhalten.

an der Seite zu arbeiten und Konfigurationen im Hintergrund vorzunehmen. Der Frontend-Bereich ist die veröffentlichte Webseite.

Im Backend werden durch die jeweiligen Verantwortlichen unter anderem folgende Aufgaben durchgeführt: das Erstellen von Inhalten (z. B. Texte, Bilder, Videos, Tabellen, Links zu anderen Webseiten), das Setzen von Zugriffsrechten, das Ein- und

Installation des Systems

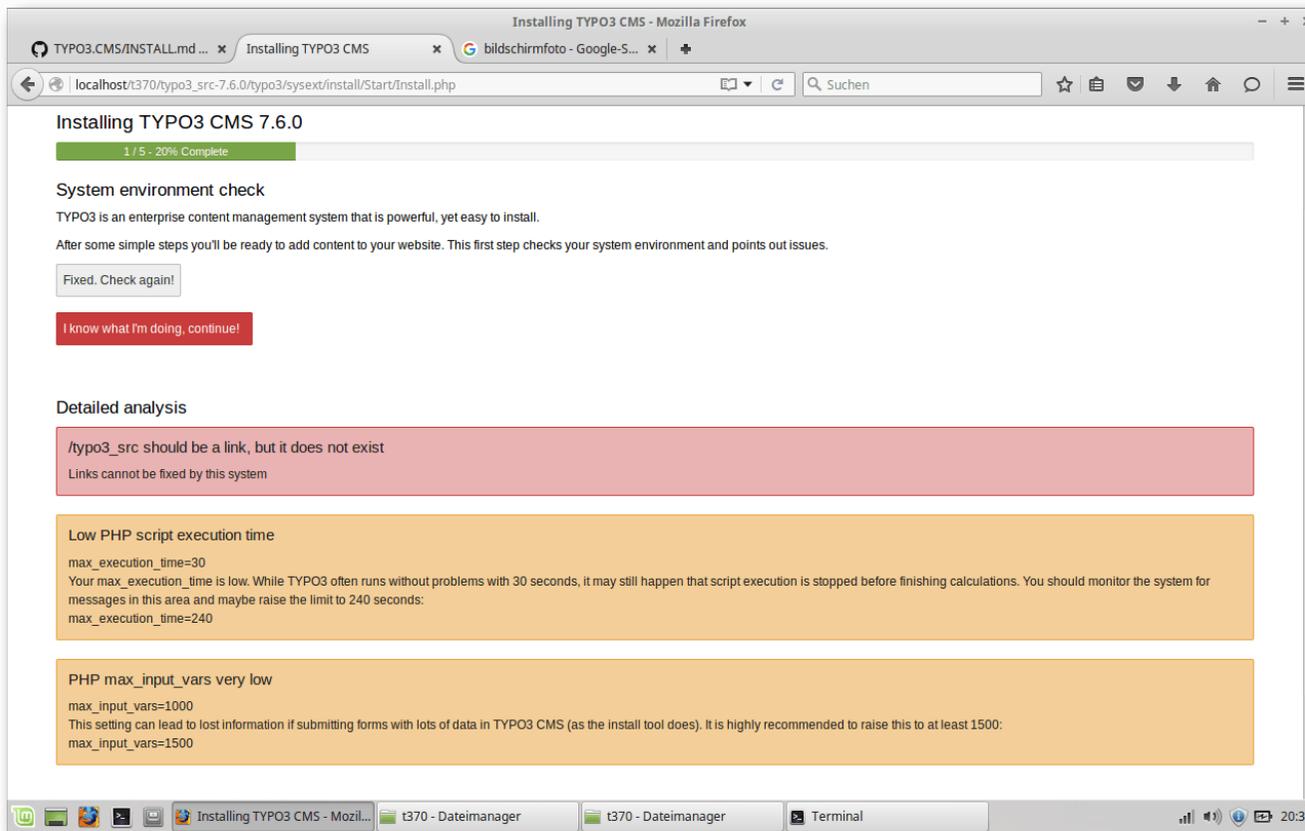
Da die Dateien des Systems lokal oder online auf einem Webserver gespeichert werden und über den Webbrowser abrufbar sind, ist vor einer Installation zu prüfen, welche Systemvoraussetzungen der Server haben muss, um mit Typo3 CMS ohne Fehler arbeiten zu können. Diese sind wie folgt (Stand Dezember 2015): PHP in den Versionen 5.5.x – 5.6.x, PHP memory_limit von 128M, PHP max_execution_time von 240 s und Internetbrowser wie der Internet Explorer 9 oder höher sowie aktuelle Browser anderer Hersteller.

Im Verzeichnis `/var/` der lokalen Verzeichnisstruktur des GNU/Linux-Betriebssystems befindet sich der Ordner `www`. In diesem Ordner kann ein Benutzer mit entsprechenden Zugriffsberechtigungen (Achtung!! Sinnvoll einsetzen!) nun den Ordner `typo3` erstellen, der alle Dateien des CMS speichert. Das kann man über den Dateimanager machen oder über die Konsole.

Für das Herunterladen und Installieren der CMS-Dateien benötigt man nun die Konsole. Am besten man öffnet diese mit Root-Rechten und wechselt in den Ordner `typo3`:

```
$ cd /var/www/typo3
/var/www/typo3
```

Da Typo3 CMS regelmäßig Sicherheitsupdates und Fehlerbehebungen bekommt, ist es ratsam, die herunterzuladenen Dateien direkt von der Downloadseite von Typo3 [4] zu speichern.



Systemcheck vor der Typo3-Installation. 🔍

Die folgende Anweisung holt den Code vom Server und speichert die zu entpackenden Dateien im aktuellen Arbeitsverzeichnis (**/var/www/typo3**).

```
# wget get.typo3.org/6.2 -O typo3_src~
-6.2.x.tar.gz
```

Damit Typo3 einwandfrei funktioniert, benötigt das System Symlinks auf das Verzeichnis **typo3**,

die **index.php** und den Ordner **typo3_src**.

Die folgenden vier Kommandos erledigen dies am Beispiel der Version Typo3 CMS 7.6.0.

```
# cd htdocs
# ln -s ../typo3_src-7.6.0 typo3_src
# ln -s typo3_src/index.php index.php
# ln -s typo3_src/typo3 typo3
```

Danach muss im Root-Verzeichnis der Installation eine leere Datei mit dem Namen **FIRST_INSTALL** gespeichert werden. Typo3 CMS benötigt dies für die Installation des Systems und die Freigabe des Installationswerkzeuges.

Zusätzlich wird die **.htaccess**-Datei in das Hauptverzeichnis der Typo3-Installation kopiert.

```
# cp typo3_src/.htaccess .htaccess
```

Die Struktur der Installation sollte dann so sein.

```
typo3_src-7.6.0/
typo3/typo3_src -> ../typo3_src-7.6.0/
typo3/typo3 -> typo3_src/typo3/
typo3/index.php -> typo3_src/index.php
typo3/.htaccess
```

Tipp: Mit dem Kommando **tree** (aus dem gleichnamigen Paket) lässt sich die Ordnerstruktur einfach und übersichtlich (in Farbe) auf der Kommandozeile überprüfen – auch misslungene Links.

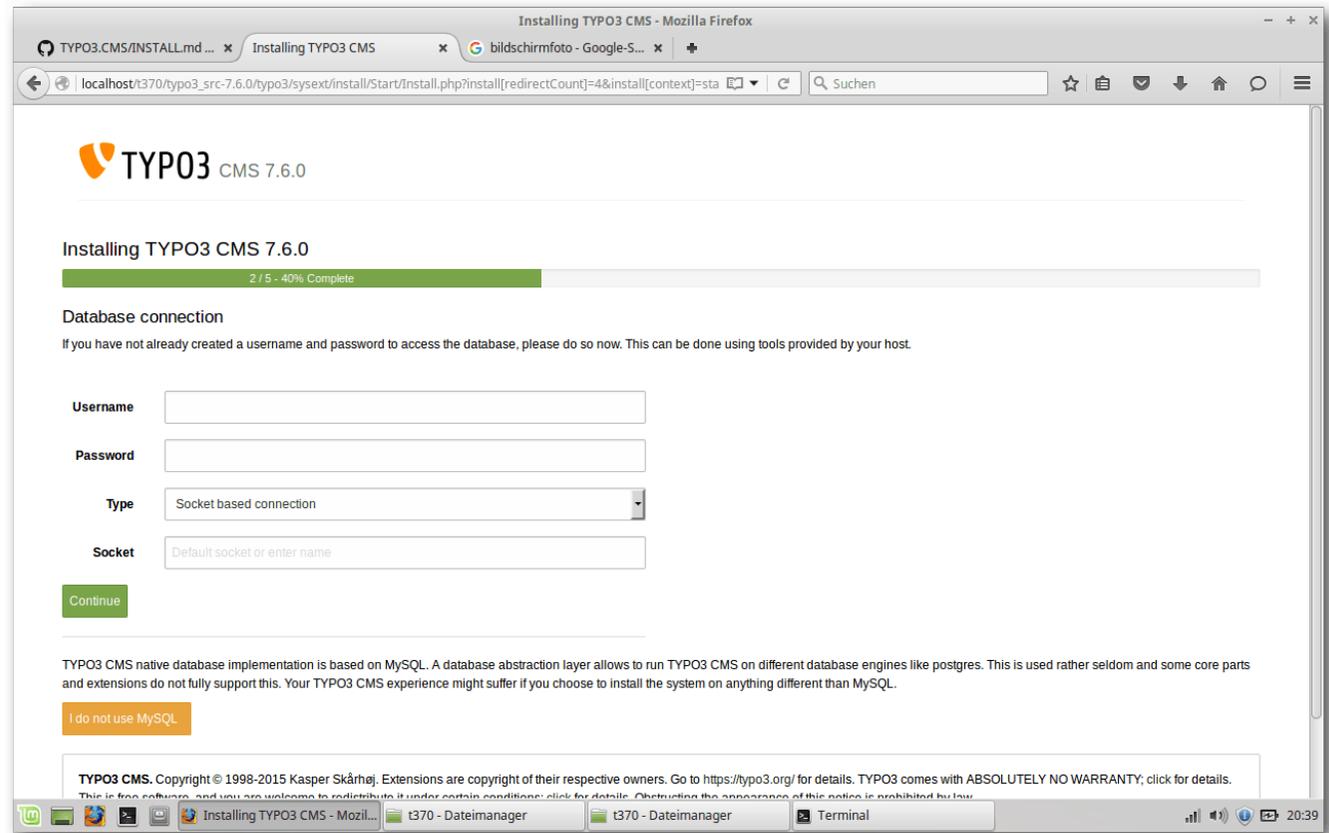
Nun kann man die lokale Typo3-Installation durchführen, indem man folgende URL im Browser aufruft: http://localhost/typo3/typo3_src-7.6.0/

Daraufhin sieht man hoffentlich, dass das System bisher richtig installiert ist und es keine Fehler gab. Dann kann man weiterklicken. Es kann auch vorkommen, dass es zu Notizen kommt, die auf fehlende PHP-Module oder -Einstellungen hinweisen.

Wenn Fehlermeldungen angezeigt werden, so ist das meistens eine der Folgenden:

- „*PHP OpenSSL extension not working*“: Diese Fehlermeldung sagt, dass das Modul OpenSSL nicht konfiguriert ist. Man sollte daraufhin nachschauen, ob das Apache-Modul „ssl_module“ aktiviert ist. In der **php.ini** muss die Funktion „extension=php_openssl.dll“ aktiviert werden. Ist das auch nicht die Ursache des Fehlers, dann müsste man zum Schluss den Ordner **sysex** nach den Unterordnern **rsaauth** und **saltedpasswords** absuchen, denn diese sind ebenso wichtig.
- „*PHP Maximum upload filesize too small upload_max=2M*“: In der **php.ini** sind folgende Änderungen der Upload-Größe vorzunehmen: `upload_max_filesize=12M` und `post_max_size=12M`. Man sollte diese Werte nicht zu niedrig halten, sonst wird diese Meldung immer wieder erscheinen [5].
- „*Low PHP script execution time*“: Da steht dann meistens ein Wert von „`max_execution_time=30`“. Ein höherer Wert ist besser, da Typo3 durch die vielen Ressourcen eine längere Zeit zum Laden von Skripten benötigt. Ein guter Wert ist zum Beispiel „240“.
- „*PHP extension fileinfo not loaded*“: In der **php.ini** Datei sollte man dann das Semikolon vor „extension=php_fileinfo.dll“ entfernen.

Obwohl die Fehlermeldungen angezeigt werden, kann man die Installation trotzdem fortführen. Diese Fehler werden auch im sogenannten „Install-



Datenbankverbindungsdaten eingeben. 🔍

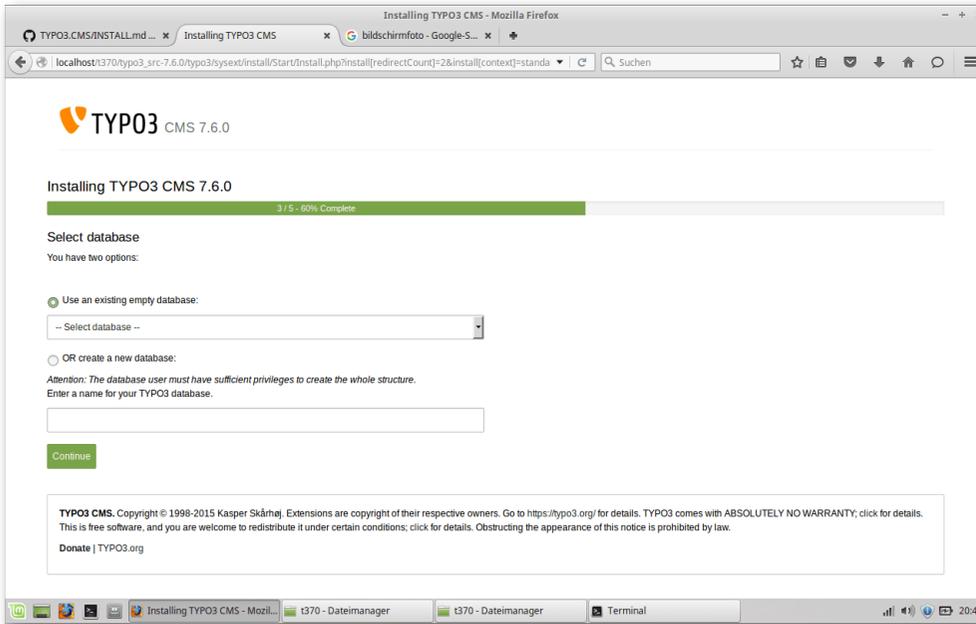
Tool“ im Backend angezeigt und könnten auch dort bearbeitet werden.

Hinweis: Nach jeder Änderung ist es wichtig, den Server neu zu starten.

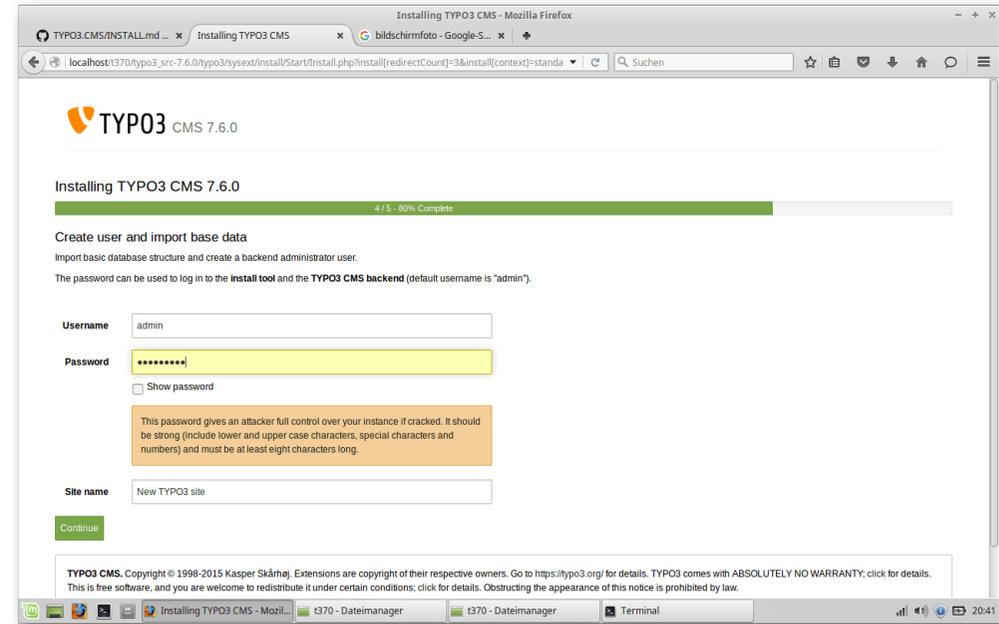
In der nächsten Maske wählt man die Datenbankverbindung aus. Hierzu benötigt man einen Datenbank-Benutzernamen und das

dazugehörige Passwort. Diese Daten sollte man unbedingt lokal wie online einrichten, das erhöht die Sicherheit des Systems.

Wenn man die richtigen Daten eingegeben hat, wird man zur nächsten Maske weitergeleitet, die die Datenbankauswahl zulässt. Dort kann man eine schon existierende Datenbank auswählen oder eine neue Datenbank auswählen.



Datenbankauswahl oder neue Datenbank auswählen. 🔍



Backend-Benutzer erstellen und Webseitenamen vergeben. 🔍

Ist das erledigt, wird in der folgenden Einstellungs-
maske der erste Backend-Benutzer festgelegt.
Hier sollte auf sichere Authentifizierungsmerkmale
geachtet werden.

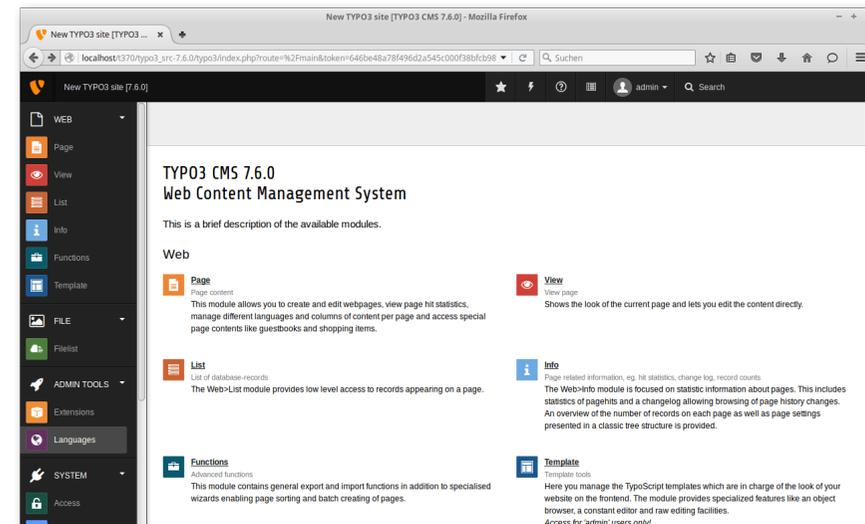
Zusätzlich kann ein Seitenname der Typo3-
Installation gewählt werden. Standard ist immer
„New TYPO3 site“. Diese Einstellung lässt sich
aber jederzeit ändern.

Danach wird dem Benutzer die letzte Maske mit
Installationsschritten angezeigt. Auf ihr werden
drei verschiedene Optionen, von denen eine aus-
zuwählen ist, angezeigt. Man kann das Backend
von Typo3 CMS mit einem Weiterklick erreichen,

eine einfache Testseite automa-
tisch vom System erstellen oder
eine komplette und größere Bei-
spielseite mit Inhalten und Ein-
stellungen einrichten lassen.

Egal, welche Option man wählt,
der Benutzer wird dadurch sofort
im Backend des Systems ange-
meldet und auf dessen Interface
weitergeleitet.

Typo3 einrichten
Im Backend gibt es eine Menü-
leiste, die folgende Funktionen



Startseite des Backends von Typo3 CMS 7.6.0. 🔍

enthält: „*Caches löschen*“, „*Hilfe*“, „*bereits bearbeitete Dokumente öffnen*“, „*Benutzerkontoeinstellungen*“, „*Logout*“ und „*Suchfunktion*“.

In der linken Spalte befinden sich verschiedene Funktionen des Systems, um Inhalte verwalten und Einstellungen vornehmen zu können.

Die Standardsprache des Systems ist Englisch. Da Typo3 CMS weltweit eingesetzt wird und auch Benutzer der verschiedenen Sprachen daran arbeiten können, kann man im Backendbereich die eigene Sprache festlegen. Das geht ganz einfach. Im Bereich „*Admin Tools*“ in der linken Menüspalte findet man den Punkt „*Language*“. Klickt man darauf, lädt eine Übersichtsseite mit den vielen Sprachen, die das System anbietet. Dort sucht man zum Beispiel „*German*“ für Deutsch aus und klickt auf das Plus-Symbol. Nun wird die Seite ins Deutsche übersetzt.

Doch dadurch ist die deutsche Anzeige noch nicht sichtbar. In der „*Benutzerkontoeinstellung*“, im oberen Menü, kann man auf den „*Benutzernamen*“ und auf „*Einstellungen*“ klicken. Es öffnet sich die Einstellungsseite, auf der man weiter unten die Sprache auf Deutsch einstellen kann. Erst nach dem Abspeichern wird die Seite im Backend komplett in deutscher Sprache angezeigt.

TypoScript

Neben der Konfiguration und Einstellung von Inhaltselementen auf der Webseite durch die verschiedenen Erweiterungen kann man auch mit

der Konfigurationssprache TypoScript arbeiten. TypoScript ist ein mächtiges Werkzeug und wird die eingegebenen Konfigurationen in ein PHP-Array umwandeln. Dieses wird PHP-Befehle ausführen, die bestimmte Elemente auf der Seite auch global darstellen können.

TypoScript wird in einer objektorientierten Syntax geschrieben. Solche Objekte werden durch diverse Code-Wörter und Zeilen dargestellt, die man zum Beispiel verwenden kann, um einen Satz wie „Hello World!“ auszugeben.

```
page = PAGE
page.10 = TEXT
page.10.value = Hello World
page.10.wrap = <h1>|</h1>
```

Mit dem **page**-Objekt wird definiert, was und wie Inhalte auf der Webseite dargestellt werden. In der zweiten Zeile des Codeblocks sieht man die Angabe **TEXT**. Der auszugebende Text steht eine Zeile darunter und heißt **Hello World**. Die letzte Zeile konfiguriert die Ausgabe des HTML-Quellcodes, der um den Satz herum platziert wird.

Das ist allerdings nur ein einfaches Beispiel, um die Ausgabe eines einfachen Textes zu veranlassen. Es sind mit TypoScript natürlich auch wesentlich komplexere Darstellungen möglich.

TypoScript wird auch dazu verwendet, Benutzereinstellungen im Backend vorzunehmen oder die weiteren Konfigurationen der Seite einzustellen.

Mit „*User TSConfig*“ werden Backendbenutzereinstellungen vorgenommen und mit „*Page TSConfig*“ die Einstellungen der Seiten und Unterseiten.

Fazit

Als Typo3-Nutzer sollte man sich nicht darauf verlassen, alle Einstellungen und Darstellungen mit Erweiterungen zu erledigen. Nicht nur die allgemeine Handhabung des Systems muss man verstehen, sondern auch die Art und Weise, wie man mit TypoScript arbeitet.

Typo3 CMS ist ein System, das man vollständig ausreizen kann. Wenn man dazu bereit ist, ein neues und umfassendes CMS kennenzulernen, keine sehr lange Einarbeitungszeit scheut und viele Erfolge damit erleben möchte, sollte man dieses Produkt ausprobieren.

LINKS

- [1] <http://www.typo3.org/> 
- [2] <http://typo3.org/about/the-history-of-typo3/> 
- [3] <https://typo3.org/home/typo3-in-numbers/> 
- [4] <http://www.typo3.org/download/> 
- [5] <http://haraldwingerter.de/?p=1>

Autoreninformation

Martin Stock beschäftigt sich privat mit der Erstellung von Webseiten und seit 2007 mit Typo3 CMS.



Teilen



Kommentieren

GitLab Continuous Integration von Christian Stankowic

Mit der neuen Hauptversion 8 wurde GitLab [1] offiziell um eine eigene Continuous Integration (CI [2]) Komponente ergänzt, die bisher als dediziertes Projekt gepflegt wurde. Während man früher externe Lösungen wie beispielsweise Jenkins [3] verwenden musste, kann man nun gepflegte Projekte leicht automatisiert übersetzen, testen, etc.

Doch wozu dient eine CI? CI findet hauptsächlich in der professionellen Anwendungsentwicklung Verwendung und automatisiert maßgeblich Teilschritte zur Programmübersetzung und Erhöhung der Quellcode-Qualität. Um mal einige praktikable Beispiele zu nennen:

- Übersetzung von Teilprogrammen
- Zusammensetzen des Gesamtprogramms
- Tests von Teilprogrammen anhand Spezifikationen (Test-Units)
- Datei-Replikation auf andere Systeme (z. B. andere Entwicklungssysteme)

Dabei integriert sich die CI in eine Versionsverwaltung, um sich beispielsweise auf die neuesten Commits zu beziehen oder automatisiert auf Code-Änderungen zu reagieren.

Ich habe bisher GitLab in Kombination mit Jenkins verwendet, um automatisiert RPM-Pakete für mehrere Enterprise-Linux-Architekturen und

The screenshot shows the GitLab CI Runner configuration interface. On the left is a sidebar with navigation options like 'Project Settings', 'Deploy Keys', 'Web Hooks', 'Services', 'Protected Branches', 'Runners', 'Variables', 'Triggers', 'CI Web Hooks', 'CI Settings', and 'CI Services'. The main content area is titled 'Christian Stankowic / ci-test · Settings'. It provides instructions on how to set up a new project-specific runner, including steps for installing software, specifying a URL, and using a registration token. Below this, there are two sections: 'Runners activated for this project' and 'Available specific runners'. The first section shows a runner with ID '3c0b983c' (st-devel02.stankowic.loc) which is currently disabled for this project. The second section shows a runner with ID '2162f531' (st-mon02.stankowic.loc) which is available and can be enabled for this project. There is also a section for 'Shared runners' with a warning that they do not offer secure isolation and a button to 'DISABLE SHARED RUNNERS' for this project.

CI Runner-Konfiguration. 🔍

-Versionen zu erstellen. Jenkins bietet dabei gegenüber GitLab CI eindeutig mehr Flexibilität und die größere Feature-Auswahl, ist aber auch deutlich komplexer zu konfigurieren. Bei der Implementation seinerzeit gab es viel

Nachlesebedarf. GitLab CI ist simpler und dürfte sich deswegen vermutlich nicht für alle Kunden-größen eignen; für das unten stehende Szenario war es aber die bessere Wahl und ließ sich binnen weniger Stunden implementieren und testen.

Die meisten CI-Lösungen verwenden auf einem oder mehreren Build-Systemen entsprechende Agenten bzw. Runner-Prozesse, um die Prozesse (Code kompilieren, Dateien verteilen ...) zu steuern. Während Jenkins eine Java-Anwendung nutzt, bietet GitLab CI mehrere Runner an.

Vorbereitungen

Um GitLab CI zu verwenden, muss GitLab in der Version 8.0 oder höher installiert sein. Das Feature wird projektweise aktiviert und konfiguriert. Für die Steuerung der einzelnen Prozesse wird eine simple YAML-Datei mit dem Namen `.gitlab-ci.yml` im Hauptordner des Projekts erstellt – dazu gleich mehr.

Projekt-Einstellungen

Um CI für ein GitLab-Projekt zu aktivieren, genügen die folgenden Schritte:

1. Auswahl des Projekts, Anklicken von „*Settings*“ und „*Project Settings*“.
2. Anklicken von „*Builds*“ und „*Save Changes*“.
3. Auswahl von „*CI Settings*“, um erweiterte Parameter zu setzen. Dazu zählen u. a. Timeouts und automatische Builds.
4. Anklicken von „*Save Changes*“.
5. Auswahl von „*Runners*“, Notieren/Kopieren der „*CI-URL*“ und dem „*CI-Token*“ – diese Informationen werden später benötigt, um die Runner zu registrieren.

Zur Steuerung der einzelnen Schritte auf dem Runner muss die YAML-Konfiguration erstellt wer-

den. In dieser werden einzelne Teilschritte, Skripte und eventuelle Abhängigkeiten zu Runnern anhand von Tags definiert.

Ein einfaches Beispiel sieht wie folgt aus:

```
before_script:
  - hostname

build:
  script:
    - "javac *.java"

clean:
  script:
    - "find . -type f -name '*.tmp' -exec rm {} \;"
```

In diesem Beispiel werden zwei Jobs definiert:

- **build** – Skript zur Kompilierung aller Java-Quellcodes im aktuellen Verzeichnis
- **clean** – Entfernen aller Dateien, die mit `.tmp` enden

Vor dem Ausführen der Jobs wird das `hostname`-Kommando ausgeführt. Der Job wird auf einem System aus dem Pool für das Projekt verfügbarer Runner ausgeführt. Durch die Verwendung von Tags kann man dieses Verhalten einschränken:

```
before_script:
  - hostname

build_stable_jars:
```

```
  script:
    - "javac *.java"
  tags:
    - java
  only:
    - master

build_hipster_swag:
  script:
    - "bundle exec swag"
```

In diesem Beispiel gibt es erneut zwei Jobs:

- **build_stable_jars** – Übersetzen von Java-Quellcodes aus der master-Branch des Repositories; wird lediglich auf Runnern mit dem `java`-Tag ausgeführt.
- **build_hipster_swag** – Paketieren von Ruby-Anwendungen; kann auf allen Runnern ausgeführt werden und beschränkt sich auf keinen Branch

Auf der GitLab-Seite gibt es weitere Beispiele [\[4\]](#).

Runner-Einstellungen

Für die Runner-Anwendung empfiehlt sich die Erstellung eines dedizierten System-Benutzers. Für diesen Benutzer müssen auch ein Passwort gesetzt und ein SSH-Key generiert werden. Beim SSH-Key ist wichtig, dass keine Passphrase angegeben wird, um ein passwortloses Steuern zu ermöglichen. Diese Informationen werden nachher in GitLab hinterlegt, um den Runner-Prozess zu kontrollieren:

```
# useradd --comment "GitLab service user"
" --system -m su-gitlab-ci
# passwd su-gitlab-ci
# su - su-gitlab-ci
$ ssh-keygen
```

Für GitLab CI stehen mehrere Clients [5] zur Verfügung. Neben dem offiziellen, in Go entwickelten Client gibt es auch einen in Scala/Java [6].

Der offizielle Client unterstützt neben Windows, OS X und FreeBSD auch Debian- und Red-Hat-Enterprise-Linux-basierende Distributionen [7]. Alternativ gibt es auch distributionsunabhängige Binärpakete [8]. Für unterstützte Linux-Distributionen empfiehlt GitLab die Verwendung eines Skripts zur automatischen Konfiguration der Paketquellen. Hier im Artikel wird diese Konfiguration selbst vorgenommen und die in der Tabelle aufgeführten Repository-URLs zur Konfiguration der Paketquellen verwendet.

Der GPG-Key für die Paketsignierung ist unter <https://packages.gitlab.com/gpg.key> zu finden.

```
# gitlab-runner register
Please enter the gitlab-ci coordinator URL (e.g. https://gitlab.com/ci):
http://gitlab.localdomain.loc/ci
Please enter the gitlab-ci token for this runner:
xxx
Please enter the gitlab-ci description for this runner:
[gitlab.localdomain.loc]:
Please enter the gitlab-ci tags for this runner (comma separated):
rpm764,generic
INFO[0035] 7ab95543 Registering runner... succeeded
Please enter the executor: ssh, shell, parallels, docker, docker-ssh:
ssh
Please enter the SSH server address (eg. my.server.com):
gitlab.localdomain.loc
Please enter the SSH server port (eg. 22):
22
Please enter the SSH user (eg. root):
su-gitlab-ci
Please enter the SSH password (eg. docker.io):
myPassword
Please enter path to SSH identity file (eg. /home/user/.ssh/id_rsa):
/home/su-gitlab-ci/.ssh/id_rsa
INFO[0143] Runner registered successfully. Feel free to start it, but if
it's running already the config should be automatically reloaded!
```

Repository-URLs zur Konfiguration der Paketquellen

Release/Architektur	URL
Enterprise Linux 7, x86_64	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/el/7/x86_64
Enterprise Linux 6, x86_64	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/el/6/x86_64
Enterprise Linux 6, i686	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/el/6/i686
Fedora 23, x86_64	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/fedora/23/x86_64
Fedora 23, i686	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/fedora/23/i686
Fedora 22, x86_64	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/fedora/22/x86_64
Fedora 22, i686	https://packages.gitlab.com/runner/gitlab-ci-multi-runner/fedora/22/i686

Zur Installation genügt der folgende Yum-Aufruf:

```
# yum install gitlab-ci-multi-runner
```

Nachdem alle Runner installiert wurden, müssen diese mit GitLab registriert werden. Dazu wird das **gitlab-runner**-Kommando auf dem Build-System ausgeführt. Während der Konfiguration werden die CI-URL und der dazugehörige Token

angegeben. Darüber hinaus können pro System Tags vergeben werden. Insbesondere bei Projekten, die auf unterschiedlichen Systemen erstellt werden sollen (z. B. RPM-Paketebau) ist das äußerst hilfreich. Zur Steuerung des Runners stehen lokale Shells, SSH, Parallels (für VMs) oder Docker zur Verfügung. Bei der Verwendung von SSH müssen gültige Login-Daten und der zuvor generierte SSH-Key angegeben werden.

Der erste Build

Zeit, mit etwas Einfachem zu beginnen. Als Beispiel dient eine einfache kleine C-Anwendung namens `test.c`:

```
#include<stdio.h>

int main() {
    printf("GitLab CI rocks!\n");
    return 0;
}
```

Listing 1: `test.c`

Die CI-Konfigurationsdatei `.gitlab-ci.yml` sieht wie folgt aus:

```
build:
  script:
    - "rm *.out || true"
    - "gcc *.c"
```

In diesem Beispiel wird ein Job `build` definiert, der zwei Kommandos ausführt. Das erste Kommando löscht eventuell vorher übersetzte

Status	Build ID	Commit	Ref	Stage	Name	Duration	Finished at
success	Build #39	7b4c6b42	master	test	build	0 seconds	7 minutes ago
failed	Build #38	7b4c6b42	master	test	build	1 second	9 minutes ago
failed	Build #37	1f570091	master	test	build	1 second	11 minutes ago
failed	Build #36	13175997	master	test	build	2 seconds	13 minutes ago

Übersicht aktueller und vergangener Builds. 🔍

C-Programme, während der zweite Aufruf den Quellcode übersetzt.

Nach einem erfolgten Commit wird automatisch die CI ausgelöst. Ersichtlich ist das durch einen Eintrag links im „Builds“-Menü – hier ist im Idealfall eine 1 zu sehen (für einen aktuell laufenden

Build-Prozess). Ein Klick auf das Menü öffnet die Übersicht aktueller und vergangener Builds. Mit einem Klick auf einzelne Builds lassen sich Konsolenausgaben und Git-Informationen (Commit-Nummer, Branch, Autor) anzeigen. Per Mausklick auf das entsprechende Symbol lassen sich Builds wiederholen.

Einbindung der **README.md**-Datei des Repositorys. 🔍

GitLab bietet auch die Möglichkeit, Build zeitgesteuert auszuführen – z. B. spätestens alle fünf Stunden. In diesem Fall würde nach einer gewissen Zeit ein Build ausgelöst werden, sofern kein Commit erfolgt. Dieses Verhalten kann in den Projekt-Einstellungen unterhalb „CI Settings“ konfiguriert werden („Build Schedule“).

Ausblick

Mit meinem Jenkins gab es häufig Probleme – nicht zuletzt wegen Java. Häufige Updates und Re-Konfigurationen waren die Folge. GitLab CI integriert sich wunderbar in die Oberfläche, die ohnehin schon zum Entwickeln benutzt wird. Das bedeutet weniger Administrations- und Verwal-

tungsaufwand. Die integrierte CI-Lösung ist ggf. nicht allen Anforderungen gewachsen, aber vor allem für kleinere Setups ist sie ideal.

Auf der GitLab-Webseite gibt es weitere Infos [9].

LINKS

- [1] <https://about.gitlab.com/> 🇬🇧
- [2] https://de.wikipedia.org/wiki/Continuous_Integration
- [3] <https://jenkins-ci.org/> 🇬🇧
- [4] <http://doc.gitlab.com/ce/ci/yaml/> 🇬🇧
- [5] <https://about.gitlab.com/gitlab-ci/#gitlab-runner> 🇬🇧
- [6] <https://github.com/nafg/gitlab-ci-runner-scala> 🇬🇧
- [7] <https://gitlab.com/gitlab-org/gitlab-ci-multi-runner/blob/master/docs/install/linux-repository.md> 🇬🇧
- [8] <https://gitlab.com/gitlab-org/gitlab-ci-multi-runner/blob/master/docs/install/linux-manually.md> 🇬🇧
- [9] http://doc.gitlab.com/ce/ci/quick_start/ 🇬🇧

Autoreninformation

Christian Stankowic (Webseite) beschäftigt sich seit 2006 mit Linux. Als Fachinformatiker widmet er sich seit 2009 u. a. RHEL/CentOS, Spacewalk und Icinga.



Teilen



Kommentieren

Review: Steam Controller von Robert Kurz

Mit dem Steam Controller [1] möchte Valve seine Steamboxen mit Steam OS wohnzimmertauglich machen. Ob die außergewöhnlichen Bedienkonzepte auch unter Ubuntu funktionieren, soll in diesem Artikel geklärt werden.

Redaktioneller Hinweis: Der Artikel „Review: Steam Controller“ erschien erstmals in *lkhaya* auf ubuntuusers.de [2].

Alles was Valve [3] anfasst wird zu Gold, so war zumindest die langläufige Meinung. Denn egal worum es geht, ein DRM-System wie Steam, Mikrotransaktionen in Spielen wie Dota 2, Counterstrike und Team Fortress 2, aber auch E-Sport-Veranstaltungen mit Millionenpreisgeldern wie „The International“ werden immer zu Erfolgen. Doch mit Steam OS und den zugehörigen Steamboxen sowie dem Controller könnte sich Valve etwas übernommen haben.

Obwohl mittlerweile viele AAA-Spiele für Linux erscheinen – was zweifelsohne ein Erfolg des Unternehmens aus Seattle ist – wird ihr Betriebssystem noch lange nicht von der breiten Masse angenommen.

Valve verfolgt mit dem neuen Steam Controller ebenfalls ein ambitioniertes Ziel: Er soll als erster Controller so präzise wie Maus und Tastatur sein. Kann er diese Anforderungen wirklich erfüllen?



Der Steam-Controller. 🔍

Wie schwierig ist es, ihn unter Linux zu betreiben?

Aufbau

Für den Aufbau geht Valve einen unkonventionellen Weg, denn statt der Analogsticks sind in dem Controller zwei Touch-/Trackpads verbaut. Damit

kann entweder eine Mausbewegung durchgeführt oder eine Controllerbedienung wie ein Analogstick oder „Digipad“ emuliert werden. Für das nötige Feedback sorgen zwei Motoren unter den Trackpads, die mit Vibrationen ein haptisches Feedback erzeugen. Die Stärke der Vibration lässt sich in

den Einstellungen verändern und alternativ auch komplett abschalten. Eine auf den ersten Blick unsichtbare Komponente stellt der Gyrosensor dar. Mit ihm sollen sich vor allem First-Person-Spiele besser steuern lassen.

Neben dieser auffälligen Neuerung hat Valve auch an anderen Stellen das konventionelle

Controllerdesign über den Haufen geworfen. Auf der Rückseite verbergen sich zwei große Tasten, die mit den Ring- oder Zeigefingern bedient werden können.

Ganz nett, aber nicht weltbewegend, ist zudem die Möglichkeit, den Controller sowohl kabellos als auch über ein MicroUSB-Kabel mit dem Com-

puter zu verbinden. Das erspart die Notwendigkeit, ein eigenes Adapterkabel für den Controller kaufen zu müssen, wie es bei einigen Konkurrenten der Fall ist.

Bei allem anderen hat sich Valve auf bewährte Konzepte verlassen. Die Schultertasten haben einen angenehmen Widerstand und sind ergonomisch an die Form eines gekrümmten Fingers angepasst. Auch die vier üblichen Aktionstasten sind vorhanden und mit den Buchstaben A, B, X und Y beschriftet. Mittig angeordnet hat der Controller zwei Tasten. Mit der rechten lässt sich das jeweilige Spielmenü öffnen, die linke ist meist frei belegbar. Zwischen ihnen befindet sich der An-/Aus-Schalter mit einem beleuchteten Steam-Logo, der in Spielen das Steam-Overlay öffnet.

Zu guter Letzt hat auch der Steam Controller einen Analogstick spendiert bekommen, der sich leicht versetzt unter dem linken Trackpad befindet.

Hinter den auf der Rückseite befindlichen Tasten steckt auf jeder Seite ein Batteriefach für jeweils eine AA-Batterie. Diese werden mit einem durchdachten Mechanismus an ihrem Platz gehalten. Nett wäre es gewesen, anstatt der Batterien zwei Akkus mitzuliefern, was bei dem Preis möglich gewesen sein sollte.

Generell macht die Verarbeitung des Controllers einen sehr guten Eindruck; lediglich die drei mittleren Tasten könnten einen etwas festeren Druckpunkt haben.



Das Trackpad des Controllers. 🔍



Die Rückseite des Steam Controllers. 

Verbindung

Der Steam Controller verbindet sich wahlweise über einen Micro-USB-Anschluss oder kabellos mit dem PC. Ein USB-Empfänger sowie ein Verstärker für längere Verbindungen werden mitgeliefert. Die Reichweite des Controllers ist vollkommen ausreichend; er funkt sogar durch Wände noch sehr stabil. Das enthaltene USB-Kabel ist

allerdings unbrauchbar, da es mit 1,5 Meter Länge vor allem für den Wohnzimmereinsatz viel zu kurz ist. Als Micro-USB-Kabel findet es aber auch bei anderen Geräten eine neue Verwendung.

Ob man gleichzeitig mehrere Controller an einem Gerät benutzen kann, konnte nicht getestet werden. Zumindest bei der Benutzung von mehreren

unterschiedlichen Controllern an einem System kann es zu Problemen bei unterschiedlichen Tastenbelegungen kommen.

Big-Picture-Modus ist Pflicht

Der „Big-Picture-Modus“ ist eine Bedienoberfläche im Vollbild, die besonders auf die Steuerung mit Controllern ausgelegt ist. Sie ähnelt der Bedienoberfläche des KODI-Mediencenters [4] (ehemals XBMC). In diesem Modus sind sowohl der Steam Store, die eigene Spielebibliothek, der Community-Hub, als auch ein eigener rudimentärer Browser enthalten. Der Big-Picture-Modus lässt sich sowohl mit dem Steam Controller als auch mit jedem anderen gängigen Controller bedienen. An vielen Stellen fühlt sich die Bedienoberfläche für PC-Nutzer des Steam-Clients noch sehr beschränkt an. Auch dass der Zugriff auf eine Kommandozeile nicht möglich ist, ist in dem Fall wohl selbstredend.

Der Controller lässt sich seit einem Update vor einigen Wochen zwar indirekt auch ohne Steam benutzen. Allerdings wird er außerhalb von Steam kaum von Spielen nativ unterstützt. Zudem kann die Tastenbelegung nur über Steam im Big-Picture-Modus umgestellt werden. Deshalb ist der Einsatz des Controllers ohne den Wohnzimmermodus des Steam-Clients nur eingeschränkt möglich.

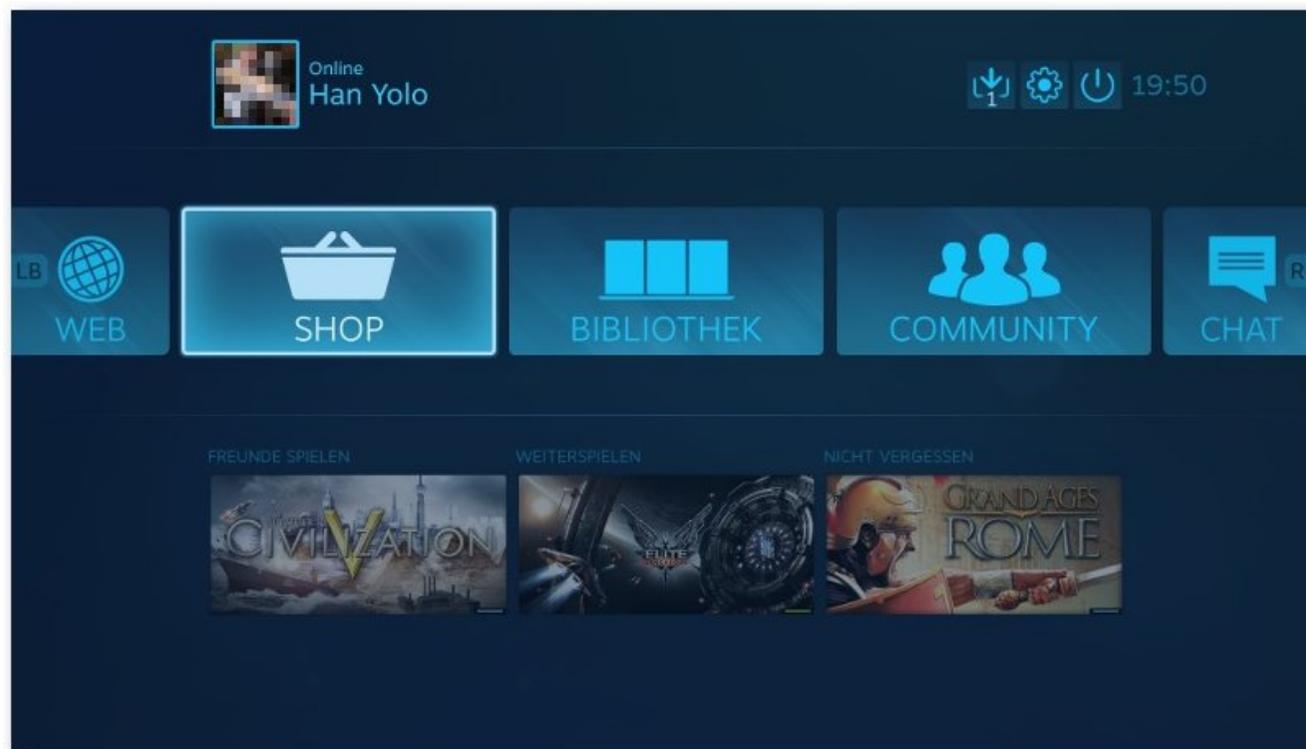
Um ein steamfremdes Spiel auch mit dem Steam Controller spielen zu können, muss man es im Menü über den Punkt „Ein steamfremdes Spiel

„hinzufügen“ verknüpfen. Danach kann es auch über den Big-Picture-Modus gespielt und die Tastenbelegung über das Menü umgestellt werden.

Hat man den Modus erst einmal aktiviert, entfaltet der Steam Controller sein volles Potenzial. Die Navigation mit dem rechten Trackpad und der rechten Schultertaste geht leicht von der Hand. Das ist nach einer kurzen Eingewöhnungszeit deutlich komfortabler als mit einem herkömmlichen Controller durch lange Listen zu navigieren. Die Genauigkeit der Maussteuerung erreicht der Controller bisher noch nicht.

Durchdacht ist auch das Schreiben mit dem Steam Controller. Dazu wird die Tastatur auf beide Touchpads aufgeteilt und die Buchstaben werden mit einem einfachen Druck auf die entsprechende Stelle des Trackpads ausgeführt. Auch das ist deutlich komfortabler als das Auswählen der einzelnen Buchstaben mit dem Analogstick wie es bisher bei handelsüblichen Controllern der Fall war. Mit etwas Übung geht das Schreiben von Nachrichten recht schnell von der Hand. Sobald es sich dabei allerdings um mehr als ein paar Sätze handelt ist ein Griff zur Tastatur trotzdem sehr verlockend. Das Wechseln zwischen Tastatur und Controller ist immer möglich und auch die Maus bleibt die ganze Zeit aktiviert. Für den normalen Gebrauch reicht der Controller allerdings vollkommen aus.

Doch nun zu einem der laut Valve weiteren Alleinstellungsmerkmale. In der Präsentation des



Der Big-Picture-Modus. 

Controllers wird mit dem hochtrabenden Slogan „fully customizable control schemes“ geworben – und zumindest damit behält die Marketingabteilung der Spieleplattform recht. Jeder Button lässt sich beliebig belegen und auch für die beiden Trackpads gibt es unzählige Konfigurationsmöglichkeiten. Ob das alles sinnvoll ist, ist natürlich eine ganz andere Frage. Doch alleine die Möglichkeit, so viele Einstellungen zu haben, ist für die „Frickler“ unter den Spielern ein wahrer Segen. Ungeduldige Spieler können über den Big-Picture-Modus auch die Konfigurationen anderer

Spieler übernehmen. Leider lassen sich diese jedoch nicht nach Bewertungen sortieren. So müssen Spieler auch hier viel ausprobieren, welche Einstellungen am besten zu ihnen passen. Seit einem Update wird bei der Auswahl der Community-Profilen immerhin die Anzahl der Downloads eines bestimmten Profils angezeigt.

Linux-Kompatibilität

Egal ob Windows oder Linux, der Steam Controller lässt sich unter beiden Systemen problemlos via Plug&Play verwenden. Sobald der ka-

bellose Adapter eingesteckt ist, lässt sich der Steam Controller über den Valve-Button anschalten. Wenn der Big-Picture-Modus gestartet ist, lässt sich mit dem Controller potenziell jedes der aktuell 3352 [5] linuxkompatiblen Spiele steuern. Wer auf Nummer sicher gehen möchte, greift lieber nur zu Spielen, die bereits eine native Controller-Unterstützung anbieten. Unter diesem Kriterium bleiben noch knapp 1100 [6] Spiele zur Auswahl. Die Auswahl der AAA-Titel mit Linux-Unterstützung ist allerdings immer noch sehr gering.

Allgemeine Erfahrungen

Mit einem Preis von etwa 55 Euro plus Versandkosten ist der Steam Controller definitiv kein Schnäppchen, auch wenn Microsoft mit seinem neuen „Elite-Controller“ [7] in ganz andere Preissphären vorstößt. Für den Preis fühlt sich Valves Controller ordentlich verarbeitet an. Einige einzelne Buttons könnten allerdings einen etwas festeren Druckpunkt haben.

Die verbauten Touchpads überraschen sowohl in positiver als auch negativer Hinsicht. Negativ ist, dass der Steam Controller in First-Person-Spielen in einigen Fällen sogar noch schlechter zu bedienen ist als ein herkömmlicher Controller. Das war beispielsweise bei Bioshock Infinite der Fall. In Portal 2 schlug sich der Controller recht wacker, alles andere wäre für ein hauseigenes Spiel auch ziemlich peinlich. Der im Controller verbaute Bewegungssensor kann bei einigen Shootern zwar hilfreich sein, um kleine Korrekturbewegungen

zum genaueren Zielen zu machen, gerade gegenüber dem Zielen mit der Maus ist diese Methode trotzdem recht langsam.

Seine Stärken konnte der Controller vor allem in Rundenstrategiespielen und anderen eher langsamen Titeln entfalten. Paradebeispiele hierbei sind vor allem „Civilisation 5“ und das Indiespiel „Papers, Please“ (siehe freiesMagazin 06/2014 [8]). Erstaunlich gut funktionierte außerdem die Weltraumsimulation „Elite Dangerous“, auch wenn dieser Titel leider nicht für Linux erhältlich ist.

Fazit

Der Steam Controller funktioniert unter Linux besser als gedacht. Das Bedienkonzept mit den zwei Trackpads ist ein sehr guter Ansatz und funktioniert vor allem in der Bedienung des Big-Picture-Modus sehr gut. Weniger gut ist die Implementierung in den Spielen. Auch wenn die Bedienung etwas besser funktioniert als mit einem herkömmlichen Controller, gibt es immer noch zu viele Totalausfälle à la Dota 2, die sich mit einem Controller etwa so gut spielen wie auf einem USB-Toaster. Doch für das Genre der MOBAs [9] ist



Die Einstellungsmöglichkeiten. 

der Controller nicht gemacht. Definitiv auch nicht für reaktionsschnelle Shooter ohne „aim assist“ im Stil von Battlefield, Counterstrike oder Quake. Eine zum Test gespielte Runde Counterstrike im Matchmaking endete mit zwei Kills und einem Downrank. Selbst mit dem seit einiger Zeit verbesserten Gyrosensor zieht man mit dem Controller gegen gleichwertige Spieler mit Maus und Tastatur meistens den kürzeren.

Seine vollen Fähigkeiten entfaltet der Controller erst bei controlleroptimierten Spielen oder Rundenstrategietiteln wie Civilisation 5. Maus und Tastatur bleiben trotzdem immer unangefochten die besten Eingabegeräte. Das Eingabegerät von Valve eignet sich dafür gut als Ergänzung für das Sofa oder zurückgelehntes Spielen. Dass der

Controller auch dafür noch nicht perfekt ist, merkt man an den wöchentlich mehrmals erscheinenden Patches. In ihnen nimmt Valve teilweise noch große Änderungen an den Bedienkonzepten vor. Löblich ist, dass auf Feedback der Community sehr schnell reagiert wird.

LINKS

- [1] <http://store.steampowered.com/universe/controller>
- [2] <https://ikhaya.ubuntuusers.de/2015/12/29/review-steam-controller/>
- [3] <http://valvesoftware.com/> 
- [4] <http://kodi.tv/> 
- [5] http://store.steampowered.com/search/?sort_by=Name&sort_order=ASC&os=linux 
- [6] http://store.steampowered.com/search/?sort_by=Name&sort_order=ASC&os=linux#

[sort_by=Name&category2=28&os=linux&page=1](http://store.steampowered.com/search/?sort_by=Name&category2=28&os=linux&page=1) 

[7] <http://www.heise.de/-2721411>

[8] <http://www.freiesmagazin.de/freiesMagazin-2014-06>

[9] https://de.wikipedia.org/wiki/Multiplayer_Online_Battle_Arena

Autoreninformation

Robert Kurz spielt auf seinem Linux-PC schon seit er als Kind herausgefunden hat, wie man SuperTux installiert.

 [Teilen](#)

 [Kommentieren](#)



“Wisdom Teeth” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/861/>

Rezension: Professional Python von Jochen Schnelle

Python gilt als Programmiersprache, welche sich besonders gut für Programmieranfänger eignet. Allerdings bietet Python noch weit mehr, auch für professionelle Programmierer beziehungsweise „große“ Projekte. Diesen fortgeschrittenen Themen widmet sich das vorliegende, englischsprachige Buch „Professional Python“ [1].

Redaktioneller Hinweis: Wir danken Wiley-VCH für die Bereitstellung eines Rezensionsexemplares.

Gemäß einleitendem Text auf der ersten Seite des Buchs ist das Buch für Python-Programmierer gedacht, welche bereits Erfahrung mit der Sprache haben und jetzt einen Blick auf die fortgeschrittenen Konzepte und Möglichkeiten der Programmiersprache werfen möchten.

Was steht drin?

In der ersten Hälfte des Buchs werden in sieben Kapiteln verschiedene fortgeschrittene Möglichkeiten von Python vorgestellt, namentlich Dekoratoren, Context Manager, Generatoren, „Magic Methods“, Metaklassen, „Class Factories“ und das Konzept der „Abstract Base Classes“.

In der zweiten Hälfte des Buchs widmet sich der Autor dann Themen wie den Unterschieden zwischen Python 2 und 3, regulären Ausdrücken, Unit

Testing, den Bibliotheken `optparse` und `argparse` zum Parsen von Kommandozeilenoptionen sowie der relativ neuen Bibliothek `asyncio`. Den Abschluss des Buchs machen dann ein paar Seiten mit allgemeinen Tipps zum Schreiben von les- und wartbaren Python-Programmen sowie zu gutem Programmierstil.

Das komplette Inhaltsverzeichnis des Buchs ist auch auf der Webseite des Verlags [1] einzusehen.

Gute Abdeckung

Gut ist, dass das Buch auf alle Python-Versionen eingeht, welche zum Zeitpunkt des Schreibens „in der freien Wildbahn“ vorzufinden sind, also Python 2.6 und 2.7 sowie die Version 3.2, 3.3 und 3.4. Auf das Thema „Python 2 vs. Python 3“ wird in zwei eigenen Kapiteln eingegangen, nämlich denen zu den Datentypen `String`, `Byte` und `Unicode` sowie in einem Kapitel, was sonstige Änderungen und Unterschiede zwischen den Versionen aufzeigt und erläutert. In letzterem wird auch häufig auf die externe Bibliothek `six` [2] eingegangen, welche das Portieren von Python 2.x auf Python 3 erleichtert.

Wie liest es sich?

Insgesamt liest sich das Buch recht flüssig, wobei das verwendete Englisch schon etwas anspruchsvoller ist und über das Niveau von Schulenglisch hinausgeht.

Der Autor verwendet viele Beispiele, um die vorgestellten Möglichkeiten und Konzepte zu zeigen und dann zu erläutern. Alle Listings sind komplett im Buch abgedruckt. Leider sind einige Codebeispiele recht kurz gehalten und wirken dadurch etwas abstrakt bzw. nicht besonders praxisnah – was letztendlich das Nachvollziehen erschwert.

Kritik

Dem selbst gestellten Anspruch, in fortgeschrittene Sprachmittel von Python einzuführen, wird das Buch meiner Meinung nach nur in der ersten Hälfte gerecht. Die in der zweiten Hälfte behandelten Themen wie z. B. die Datentypen `String`, `Byte` und `Unicode` oder reguläre Ausdrücke werden in ähnlichem Umfang und in ähnlicher Tiefe auch in manchen (umfangreicheren) Einsteigerbüchern behandelt.

Hier wäre es meiner Meinung nach deutlich besser gewesen, die ersten Hälfte des Buchs zu vertiefen, z. B. durch ausführlichere und mehr Beispiele. Das Konzept der „klassischen“ Co-Routinen, welche Python ebenfalls kennt, wird

Buchinformationen

Titel	Professional Python [1]
Autor	Luke Sneeringer
Verlag	Wiley-VCH
Umfang	Umfang: 260 Seiten
ISBN	978-1-119-07085-6
Preis	45,50 € (Taschenbuch), 34,99 € (E-Book)

gar nicht behandelt, beziehungsweise nur kurz im Rahmen der asyncio-Co-Routinen.

Fazit

Das Buch „Professional Python“ führt in der ersten Hälfte kompakt in einige fortgeschrittene Möglichkeiten von Python ein, in der zweiten Hälfte fällt der Anspruch dann aber ab. Bei einem Umfang von ca. 250 Seiten und einem Preis von 45,50 € (für die gedruckte Ausgabe) ist das Buch selbst für ein IT-Fachbuch sehr teuer. Vor dem Kauf sollte man also für sich selber prüfen, ob der Inhalt den Preis rechtfertigt.

Eine Alternative zu dem Buch wäre „Fluent Python“ aus dem Verlag O'Reilly [3], welches die gleiche Zielgruppe wie das vorliegende Buch hat,

bei dreifachem Umfang aber deutlich mehr Themen bei einer besseren Abdeckung und Tiefe bietet.

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

„Wie im Artikel erwähnt ist das Python-Modul `asyncio` relativ neu. Seit welcher Python-Version befindet es sich in der Standardbibliothek?“

Die Antwort kann bis zum **14. Februar 2016, 23:59 Uhr** über die Kommentarfunktion oder per E-Mail an redaktion@freiesMagazin.de geschickt werden. Die Kommentare werden bis zum Ende der Verlosung nicht freigeschaltet. Das

Buch wird unter allen Einsendern, die die Frage richtig beantworten konnten, verlost.

LINKS

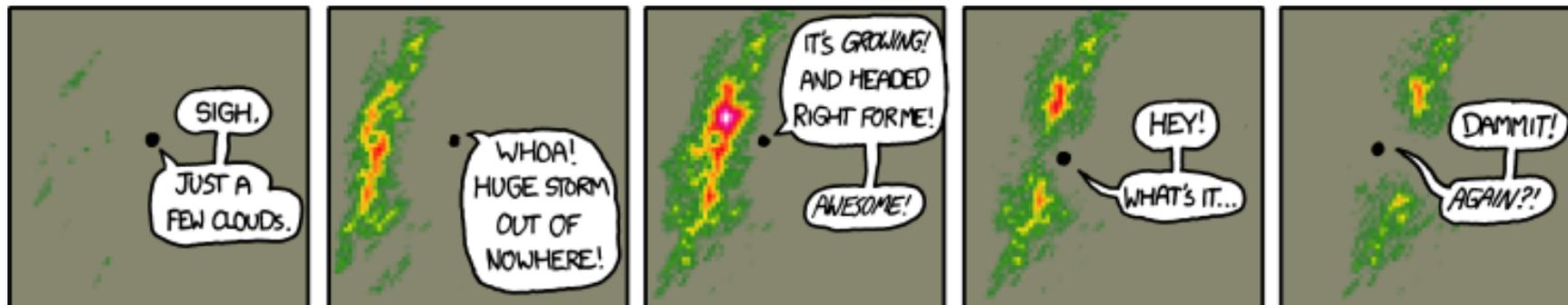
- [1] <http://www.wiley-vch.de/publish/dt/books/ISBN1-119-07085-6/>
- [2] <https://pypi.python.org/pypi/six/> 
- [3] <http://shop.oreilly.com/product/0636920032519.do> 

Autoreninformation

Jochen Schnelle ([Webseite](#)) programmiert bevorzugt in Python.

 [Teilen](#)

 [Kommentieren](#)



WHEN THE FOLKS AT THE WEATHER OFFICES
SEE YOU REFRESHING THE RADAR TOO OFTEN,
THEY START TEASING YOU.

“Weather Radar” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/831/>

Rezension: Open Source und Schule – Warum Bildung Offenheit braucht von Dominik Wagenführ

Die meisten Menschen wurden in ihrer Schulzeit hauptsächlich mit proprietären Programmen konfrontiert: Microsoft Windows, MS Office, Adobe, Oracle und Co. Selbst im Studium setzt sich das meist fort. Dabei gibt es inzwischen sehr oft freie Alternativen, die zum einen meistens kostenlos sind, zum anderen aber auch noch frei verteilt werden dürfen. Das Buch „Open Source und Schule“ [1] zeigt einige der Anwendungsgebiete und welche Hürden auftreten können.

Inhalt

Wichtig zu wissen ist, dass dieses Buch nicht aus einer Feder stammt, sondern mehrere Autoren beteiligt waren. Der Herausgeber, Sebastian Seitz hat insgesamt sieben weitere Autoren gefunden, die etwas beizutragen hatten.

Das Buch gliedert sich in drei Bereiche: „Primar- und Sekundarstufe“, „Aus- und Weiterbildung“ sowie „Administration“. Die IT-Betreuer und Lehrer berichten über den Einsatz diverser Open-Source-Programme in ihrem jeweiligen Umfeld. Meist zeigt sich, dass für derlei Betreuung kein Geld zur Verfügung steht. Woraufhin die betroffenen Lehrkräfte ihr Hobby mit an den Arbeitsplatz bringen, um Möglichkeiten aufzuzeigen. Im besten Fall werden ihnen dabei keine Steine in den Weg gelegt. Sehr oft behindern auch äußere Vorgaben (wie zum Beispiel die eingesetzten Betriebssysteme in den privaten Haushalten) den Umstieg.

Dennoch zeigt das Buch neben den Hindernissen auch Erfolge. Dabei werden die vorgestellten Lösungen weder schöneredet, noch Nachteile verschwiegen und die oft überwiegenden Vorteile motivieren dazu, sich aufgetretenen Problemen etwas später zu widmen.

Zielgruppe

Zielgruppe sind ganz klar Pädagogen und Techniker, die Open-Source-Software in ihrem Umfeld einsetzen wollen. Da die Autoren aus den gleichen Berufen kommen, liest sich das Buch entsprechend wissenschaftlich bzw. technisch, was das Verständnis nicht immer erleichtert.

Sehr gut ist, dass auch für Außenstehende ohne Open-Source-Erfahrung entsprechende Verweise als Fußnote genau erklärt werden oder zumindest mit einem Link versehen sind.

Fazit

Das Buch lässt sich angenehm lesen, da die verschiedenen Beiträge interessant geschrieben sind, auch wenn natürlich der Schreibstil je nach Autor variiert. Inhaltlich wirken die Artikel sehr fundiert als Erfahrungsbericht mit zahlreichen Fakten und nicht nur als bloße Meinungsdarstellung. Einzig der Beitrag von Wolf-Dieter Zimmermann fällt etwas ab, da nicht immer ersichtlich ist, was nun Fakt und was nur bloße Meinung des Autors ist. Dennoch hat Herausgeber Sebastian Seitz eine sehr gute Mischung an Beiträgen gefunden.

Das Buch unterliegt der Creative-Commons-Lizenz Attribution-ShareAlike [2], sodass die Texte mit Angabe der Autoren entsprechend frei weiter verteilt werden können. Auf der Webseite des Herausgebers [1] findet man das Buch im PDF- oder E-Book-Format als kostenlosen Download, wobei die E-Book-Version einige, kleinere Darstellungsfehler hat. Auch den $\text{T}_\text{E}\text{X}$ -Quellcode, mit dem das Buch gesetzt wurde, findet man dort.

Buchinformationen

Titel	Open Source und Schule – Warum Bildung Offenheit braucht [1]
Autor	Sebastian Seitz und weitere
Verlag	Liberio GmbH
Umfang	139 Seiten
ISBN	978-3-86373-034-5
Preis	kostenlos (PDF und EPUB)

LINKS

- [1] <http://s-seitz.de/index.php/buch/>
- [2] <http://creativecommons.org/licenses/by-sa/3.0/de/>

Autoreninformation

Dominik Wagenführ ([Webseite](#)) freut sich über den Einsatz Freie Software im Bildungsbereich.



Teilen



Kommentieren

Rezension: Das Sketchnote-Arbeitsbuch von Dominik Wagenführ

Nachdem Mike Rohde [1] in seinem Handbuch erklärt hat, was die Essenz einer Sketchnote ist (siehe freiesMagazin 09/2015 [2]), geht es im Sketchnote-Arbeitsbuch [3] an die harte Arbeit. Das, was im Handbuch gefehlt hat, wird diesmal erklärt – nämlich die Anwendungsgebiete für Sketchnotes.

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

Anwendungen von Sketchnotes

Nach einer (sehr) kurzen Einführung, die einige der wichtigsten Aspekte aus dem Sketchnote-Handbuch noch einmal zusammenfasst, stellt Mike Rohde in den weiteren Kapitel dar, wozu Sketchnotes nützlich sein können außer für das bloße Aufzeichnen von Meetings oder Vorträgen.

Zu den Anwendungsgebieten zählen Ideenfindung/Brainstorming, Urlaubs- und Auftragsplanung, Dokumentation von Ereignissen und Reisen, Festhalten von Eindrücken aus Film, TV und anderen Medien.

Zum Abschluss stellt der Autor noch einige erweiterte Techniken vor, die man bei der Erstellung von Sketchnotes anwenden kann.

Das Buch ist dabei wie das Sketchnote-Handbuch

mit Übungen gespickt, die zum Mitmachen einladen, und enthält wieder zahlreiche Sketchnote-Beispiele von bekannten Sketchnoters aus aller Welt.

Wie liest es sich?

Im Gegensatz zum Sketchnote-Handbuch muss man das Sketchnote-Arbeitsbuch nicht zwingend von vorne nach hinten durchlesen. Man kann sich z. B. kurz vor der Urlaubsvorbereitung auch einfach noch einmal das Kapitel zur Urlaubsplanung durchlesen und die Tipps entsprechend anwenden.

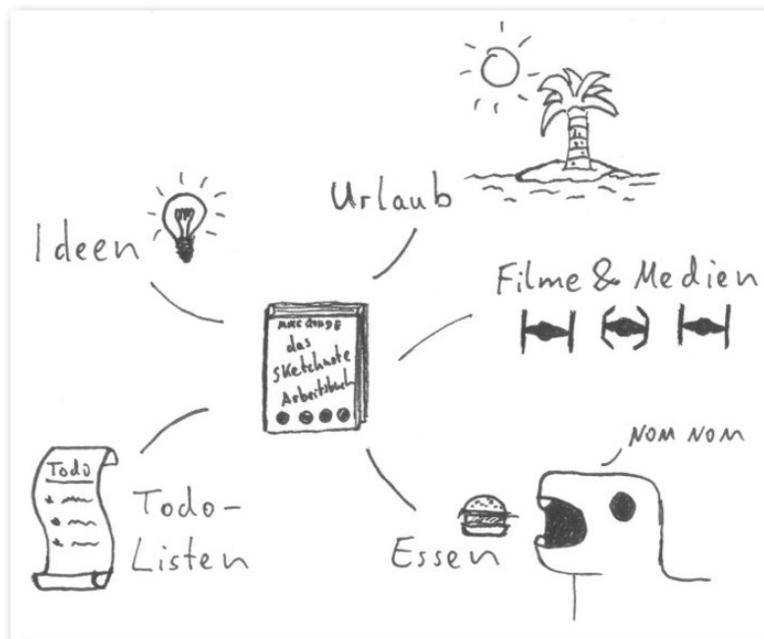
Wichtig ist, dass man ohne Sketchnote-Kenntnisse vermutlich überfordert ist, die Hinweise umzusetzen. Die Einführung fasst auf 10 Seiten zusammen, was das Sketchnote-Handbuch auf 210 Seiten erklärte. Es sollte klar sein, dass die Einführung daher wirklich nur der kurzen Wiederholung dient.

Auch die Übungen sind um einiges schwieriger umzusetzen. Sollte man zuvor noch simple Grafiken, Icons oder Schriften zeichnen, verlangt das Buch jetzt von einem, mit einer kompletten Geschäftsidee für einen Limo-Stand oder einem Baumhaus aufzuwarten.

Fazit

Das Sketchnote-Arbeitsbuch zeigt das, was ich in der Rezension zum Sketchnote-Handbuch noch bemängelte: Es zeigt die Anwendungsfälle, wo man Sketchnotes einsetzen kann und gibt auch einige Tipps, worauf man dabei achten sollte. Damit ist es eine gute Ergänzung zum Sketchnote-Handbuch und sollte sogar nicht alleine ohne dieses gelesen werden, um die verschiedenen Techniken anwenden zu können.

Die Übungen sind recht schwer, denn selbst wer die Kreativität hat, sich beispielsweise auszudenken, wie ein



Inhalte des Sketchnote-Arbeitsbuches. 

Buchinformationen

Titel	Das Sketchnote-Arbeitsbuch [3]
Autor	Mike Rohde
Verlag	mitp, 1. Auflage 2014
Umfang	208 Seiten
ISBN	978-3-8266-8493-7
Preis	24,99 Euro (Druck), 21,99 Euro (PDF)

Baumhaus aussehen könnte, wird ggf. an der kreativen Umsetzung mit Stift und Papier scheitern. Natürlich gibt es auch wieder „leichtere“ Übungen, bei denen man kleine Bildchen zeichnen soll. Aber wie stellt man „Missachtung des Gerichts“ oder „Opportunitätskosten“ dar? Hier ist die Messlatte also deutlich höher.

Insgesamt bleibt aber ein positiver Eindruck zurück, auch wenn man vielleicht nicht alles aus

dem Buch im Alltag oder Arbeitsleben anwenden kann.

Redaktioneller Hinweis: Da es schade wäre, wenn das Buch bei Dominik Wagenführ nur im Regal steht, wird es verlost. Die Gewinnfrage lautet:

„Welches Tier war in der Sketchnote-Rezension zum Sketchnote-Handbuch abgebildet?“

Die Antwort kann bis zum **14. Februar 2016, 23:59 Uhr** über die Kommentarfunktion oder per E-Mail an redaktion@freiesMagazin.de geschickt werden. Die Kommentare werden bis zum Ende der Verlosung nicht freigeschaltet. Das Buch wird unter allen Einsendern, die die Frage richtig beantworten konnten, verlost.

LINKS

- [1] <http://rohdesign.com/> 
- [2] <http://www.freiesmagazin.de/freiesMagazin-2015-09>
- [3] <http://www.mitp.de/Business-Marketing/Sketchnotes/Das-Sketchnote-Arbeitsbuch.html>

Autoreninformation 

Dominik Wagenführ ([Webseite](#))

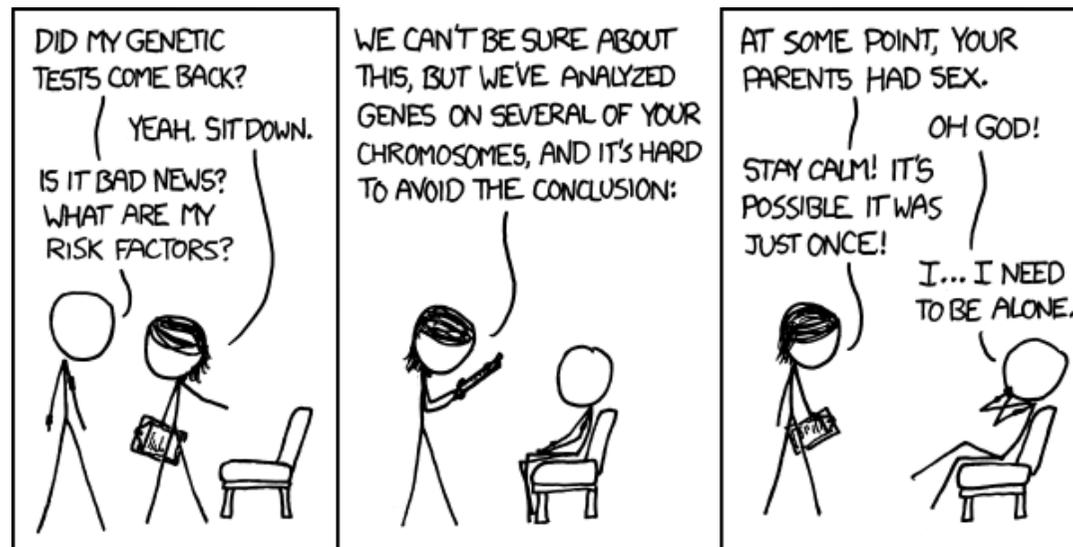
nimmt an vielen (meist technischen) Meetings teil und versucht die Inhalte inzwischen mehr mitzuzeichnen als zu schreiben.



Teilen



Kommentieren



“Genetic Analysis” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/830/>

Rezension: Linux-Server – Das umfassende Handbuch von Matthias Sitte

Linux-Server erfreuen sich immer größerer Beliebtheit, sowohl im High-End-Bereich, in dem sie immer weiter Mainframes ablösen, als auch in kleinen und mittleren Unternehmen (KMU) und bei Privatanwendern. Das vorliegende Buch „Linux-Server – Das umfassende Handbuch“ [1] (in der dritten Auflage) hat sich dabei zum Ziel gesetzt, fortgeschrittene Administratoren dabei zu unterstützen, Linux eben noch besser und effizienter einzusetzen.

Redaktioneller Hinweis: Wir danken dem Rheinwerk Verlag (vormals Galileo Press) für die Bereitstellung eines Rezensionsexemplares.

Einige Bemerkungen vorab

Das vorliegende Buch „Linux-Server“ ist in vielerlei Hinsicht facettenreich: Jedes der 27 Kapitel ist von einem der fünf Autoren (Dirk Deimeke, Stefan Kania, Charly Kühnast, Daniel van Soest und, neu hinzugekommen, Peer Heinlein) geschrieben worden, wodurch sich die Kapitel sowohl in Gestaltung, Stil und Umfang unterscheiden.

Einige Kapitel erscheinen zwar eher knapp, sind aber trotzdem sehr aussagekräftig, weil der Autor sich stets auf die wesentlichen Informationen beschränkt und diese eben sehr gut vorstellt. Andere Kapitel sind sehr ausführlich und enthalten viele und umfangreiche Informationen, die man aber nicht unbedingt alle auf einmal braucht.

Ebenfalls wichtig bei einem Buch über „Linux-Server“ ist die Wahl der Distribution. Hier haben die Autoren die folgenden Distributionen ausgewählt: Debian Wheezy, SUSE Linux Enterprise Server 11 und Ubuntu-Server 14.04 LTS. Diese Distributionen sind allesamt sehr verbreitet, kommen oft auch in Unternehmen zum Einsatz und nutzen allesamt RPM- oder DEB-Pakete für die Software-Paketverwaltung.

Nach Einleitung und Vorwort erläutert zunächst das einleitende Kapitel 1 („Der Administrator“, 20 Seiten) aus Sicht der Autoren den Beruf des Administrators, seine Aufgaben im Unternehmen sowie sein Verhältnis zu Nutzern. Den Abschluß dieses Kapitels bildet ein Überblick über den „System Administrators' Code of Ethics“ der LISA („Special Interest Group for System Administrators“ innerhalb der „Usenix Association“) – ein Verhaltenskodex, der eigentlich nicht nur für Administratoren, sondern für jeden Arbeitnehmer gilt.

Die folgenden Kapitel sind thematisch in sieben verschiedene Teile gegliedert: Grundlagen, Aufgaben, Dienste, Infrastruktur, Kommunikation, Automatisierung und Sicherheit.

Inhalt Grundlagen

Kapitel 2 („Bootvorgang“, 28 Seiten) gibt einen allgemeinen Einblick in den Startvorgang eines Linux-Systems vom Bootloader (GRUB, GRUB2)

über die initiale Ramdisk („initrd“) bis hin zum Start der eigentlichen Dienste mittels Init-Skripten. Eine kurze Einführung zum Thema „eventgesteuertes Starten und Stoppen von Diensten“ mittels Upstart schließt das Kapitel ab.

In Kapitel 3 („Festplatten und andere Devices“, 46 Seiten) geht es um „Geräte“ (engl. „devices“) im Allgemeinen. Neben RAID-Systemen wird auch erklärt, wie der „Logical Volume Manager“ (LVM) funktioniert. Weitere Inhalte des Kapitels sind das `/proc`-Dateisystem sowie `udev`.

Kapitel 4 („Dateisysteme“, 14 Seiten) gibt als Erstes eine Einführung in Dateisysteme und beschreibt, was diese eigentlich genau machen. Danach werden verschiedene Dateisysteme (Ext2/3, ReiserFS, XFS, BtrFS) im Detail besprochen und sowohl Vor- als auch Nachteile angerissen.

Kapitel 5 („Berechtigungen“, 36 Seiten) rundet Teil I ab. Hier geht es aber nicht nur um Benutzer, Gruppen und Berechtigungen auf dem Level des Dateisystems, sondern auch um den Einsatz von erweiterten „Access Control Lists“ (ACL). Weitere Themen sind Quotas und „Pluggable Authentication Modules“ (PAM) sowie die Beschränkung von Systemressourcen mittels `ulimit`.

Aufgaben

Teil II des Buches besteht nur aus zwei Kapiteln. Das erste der beiden, Kapitel 6 („Paketmanage-

ment“, 42 Seiten) beschreibt sehr ausführlich den Umgang mit Paketmanagern und Quellcode. Hier erfährt man eigentlich alles, was man braucht, um Software vom Quellcode in ein Paket (rpm oder deb) zu packen, Update-Pakete zu erstellen, oder zwischen verschiedenen Paketformaten mit **alien** zu konvertieren. Darüber hinaus werden die Paketmanager der gängigen Distributionen (**yum**, **yast**, **apt**) beschrieben. Den Abschluss des Kapitels bildet ein Überblick über verschiedene Proxy-, Cache- und Mirror-Lösungen.

In Kapitel 7 („Backup und Recovery“, 48 Seiten) werden verschiedene Backup-Strategien ausführlich beschrieben. Neben Tipps zum Erstellen von effektiven Backups mit einfachen Bordmitteln (**tar**, **rsync**, **dd**) wird auch das Open-Source-Programm „ReaR“ vorgestellt – ein Recovery-Programm, mit dem sich das komplette Grundsystem des Servers automatisch wiederherstellen lässt. Zum Schluss werden noch „Bacula“ und „Bareos“ als Enterprise-taugliche Backupssysteme vorgestellt.

Dienste

Teil III ist der längste und umfassendste Teil des Buches, der die gängigen Serverdienste beschreibt. In Kapitel 8 („Webserver“, 26 Seiten) geht es zunächst um die klassischen Webserver Apache, LightHttpd und Nginx sowie die Auswertung der Logfiles.

Kapitel 9 („FTP-Server“, 8 Seiten) beschreibt zunächst das FTP-Protokoll mitsamt Vor- und

Nachteilen, Schwächen sowie seinen Erweiterungen. Als FTP-Server wird „vsftpd“ verwendet, für den im Folgenden Beispielkonfigurationen als Download-Server oder für den Zugriff der User auf Home-Verzeichnisse beschrieben werden. Den Abschluss des Kapitels bilden „FTP over SSL“ (FTPS) sowie die Anbindung an LDAP.

In Kapitel 10 („Mailserver“, 44 Seiten) wird ein klassischer Mailserver mit Postfix als SMTP-Server und Dovecot als POP3/IMAP-Server eingerichtet, der von einigen weiteren Programmen wie Amavisd-new, ClamAV und Spamassassin zur Abwehr von Viren und Spam unterstützt wird. Monitoring und Logfiles runden das Kapitel ab.

Kapitel 11 („Datenbank“, 26 Seiten) beschreibt Basisstrategien für Datenbanken und Aufgaben, auf die man bei Datenbanken regelmäßig trifft. Dies umfasst neben Tuning auch das Backup sowie „Point-In-Time Recovery“ von Datenbanken.

Kapitel 12 („Syslog“, 18 Seiten) beschreibt den allgemeinen Aufbau von Syslog-Nachrichten und stellt die drei Syslog-Server „SyslogD“, „Syslog-ng“, „Rsyslog“ und deren Konfiguration vor.

In Kapitel 13 („Proxy-Server“, 56 Seiten) erfährt man alles über Proxy-Server, womit aber nicht nur HTTP/HTTPS-Proxies, sondern auch andere Dienste wie z. B. FTP, NTP, NNTP oder SMTP gemeint sind. Nach einer kurzen Einführung werden am Beispiel von Squid werden verschiedene Herangehensweisen und Vorüberlegungen sowie die

Grundkonfiguration mitsamt Authentifizierung beschrieben. Hilfsprogramme zur Log-Auswertung sowie fortgeschrittene Themen wie „transparent proxy“, „delay_pools“ und „Sibling/Parent“ runden das Kapitel ab.

In Kapitel 14 („Kerberos“, 42 Seiten) wird Kerberos als Single-Sign-On-Dienst (SSO) zur Authentifizierung an verschiedenen Diensten beschrieben. Nach einer Beschreibung der generellen Funktionsweise folgen Installation und Konfiguration des Kerberos-Servers sowie die Verbindung von Kerberos und PAM. Anschließend kann man Benutzer, Hosts und Dienste einrichten sowie Kerberos-Clients konfigurieren. Die fortgeschritteneren Themen „Replikation“, „Kerberos Policies“ und „Kerberos und LDAP“ bilden den Abschluss des Kapitels.

Wer kennt nicht sie nicht – heterogene Unternehmensnetzwerke, in denen sowohl Windows-Server als auch Linux-Server gemeinsam verschiedene Aufgaben übernehmen. Kapitel 15 („Samba“, 96 Seiten) ist das längste Kapitel des Buches und führt ausführlich in die Welt von SMB und NetBIOS sowie Samba als Fileserver ein. Nicht fehlen dürfen natürlich die Themen Benutzerverwaltung, Passwort-Backends („passdb backends“), Domänencontroller, Winbind, Printserver und Samba/Kerberos.

Ein eigenes Kapitel („Samba 4“, 54 Seiten) bekommen die neuen Funktionen von Samba 4. Hier geht es nicht nur um den Aufbau einer Active-Directory-Struktur mit Domaincontroller und deren

Verwaltung, sondern auch um die Benutzer- und Gruppenverwaltung mitsamt Gruppenrichtlinien.

Kapitel 17 („NFS“, 20 Seiten) beschreibt detailliert, wie man mit „NFSv4“ einen Dienst einrichtet, der Daten für Linux-Clients auf einem Dateiserver bereitstellen kann. Hier wird zuerst auf die Unterschiede zwischen „NFSv3“ und „NFSv4“ eingegangen, woran sich eine Übersicht über die Funktionsweise des NFS-Servers anschließt. Die Einrichtung des Servers sowie die Konfiguration der Clients mitsamt Optimierung und Anbindung an Kerberos sind weitere Inhalte dieses Kapitels.

Im zweitlängsten Kapitel 18 („LDAP“, 90 Seiten) wird der Verzeichnisdienst „OpenLDAP“ vorgestellt, der es ermöglicht, die Verwaltung der Ressourcen zentral zu steuern und an mehrere Stellen zu replizieren. OpenLDAP ist ein Dienst, der alle Ressourcen in einer hierarchischen Struktur abgelegt, die der Struktur eines Unternehmens entspricht und somit die Verwaltung und Kontrolle der Ressourcen effizient und einfach gestaltet. Daher werden zunächst die Grundlagen des Dienstes sowie die Unterschiede in den verschiedenen Distributionen diskutiert. Die Konfiguration des LDAP-Clients sowie grafische Werkzeuge für die LDAP-Verwaltung sind neben der Absicherung der Verbindung über TLS nur einige weitere Themen, die im Kapitel diskutiert werden.

Kapitel 19 („Druckserver“, 18 Seiten) schließt Teil III des Buches ab. Hier wird das Drucksystem rund um „CUPS“ betrachtet. Neben der Einrich-

tung des Printservers, Druckern, Klassen und Policies werden ebenfalls PPD-Treiberdateien, Druckquotas und die Anbindung an Kerberos erklärt.

Infrastruktur

Teil IV besteht ebenfalls nur aus zwei Kapiteln. Zunächst geht es in Kapitel 20 („Hochverfügbarkeit“, 26 Seiten) um Hochverfügbarkeitscluster. Hier wird ein Beispiel-Setup beschrieben, in dem ein Aktiv/Passiv-Cluster mit hochverfügbarem Plattenplatz erstellt und betrieben wird. Neben der Grundkonfiguration von Clusterkomponenten ist daher „Shared Storage mit DRBD“ ein wesentlicher Teil des Kapitels.

Kapitel 21 („Virtualisierung“, 34 Seiten) beschreibt zuerst die Unterschiede zwischen Gast- und Wirt-Systemen sowie verschiedene Ansätze zur Virtualisierung (KVM, Xen). Anschließend werden Installation, Management, Netzwerkgrundlagen und Live Migration von Desktop- und Server-Virtualisierungen beschrieben.

Kommunikation

Kapitel 22 („Netzwerk“, 84 Seiten) beleuchtet im Detail die verschiedenen Aspekte der Netzwerkkonfiguration und -einrichtung unter Linux. Neben der Netzwerkkonfiguration mit **ip** werden die Unterschiede zwischen **ifconfig**, **route** und **arp**, das „Bonding“ von Netzwerkkarten sowie „Policy Based Routing“ (PBR) und IPv6 erläutert. Firewalls mit **netfilter** und **iptables** werden ebenfalls erörtert. Den Abschluss des Kapitels bilden die beiden wichtigen Dienste DHCP und DNS.

Kapitel 23 („OpenSSH“, 12 Seiten) beschreibt den fortgeschrittenen Umgang mit dem OpenSSH-Client sowie dem OpenSSH-Server. Neben Public Key Authorization werden auch X11-Forwarding, Portweiterleitung und Tunneling beschrieben.

In Kapitel 24 („Administrationstools“, 34 Seiten) werden verschiedene, für die Administration sehr nützliche Programme beschrieben, die sonst eher stiefmütterlich behandelt werden. Dies umfasst u. a. **rsync**, **screen**, **nmap**, **netstat**, **lsof**, **top**, **strace**, **traceroute** und **ipcalc**.

Kapitel 25 („Versionskontrolle“, 16 Seiten) beschreibt Versionskontrollsysteme (engl. „version control system“, VCS), allerdings nur als Mittel zur Verwaltung von Konfigurationsdateien. Das Thema rund um Konflikte beim Zusammenführen („Mergen“) von unterschiedlichen Versionsständen wird dabei bewusst ausgespart. Nach einer Einführung in die allgemeine Philosophie von Versionskontrollsystemen werden die verschiedenen gängigen Programme wie CVS, Subversion, Bazaar, Mercurial und Git beschrieben. Eine tabellarische Übersicht stellt die Kommandos der einzelnen Versionskontrollsysteme gegenüber und rundet das Kapitel ab.

Automatisierung

Kapitel 26 („Scripting“, 22 Seiten) ist das erste von zwei Kapiteln in Teil IV des Buches. Hier findet man einige wertvolle Tipps, Tricks und Hinweise zum Umgang mit der Kommandozeile und zum Scripting.

Kapitel 27 („Monitoring“, 32 Seiten) führt in die Installation und Konfiguration von Nagios ein. Am Beispiel eines Webservers wird die Überwachung von Serverdiensten und freiem Festplattenplatz gezeigt, wobei Messwerte durch PNP4Nagios graphisch aufbereitet werden. Den Abschluss bildet ein Ausblick zum Monitoring mit Munin.

Sicherheit

Der siebte und letzte Teil des Buches widmet sich dem Thema „Sicherheit“. In Kapitel 28 („Sicherheit“, 54 Seiten) werden Maßnahmen vorgestellt, wie man den Linux-Server weiter absichern kann. Dies umfasst vor allem **chroot**, AppArmor, Einmalpasswörter, OpenVPN sowie „Intrusion Detection Systems“ (IDS) und „Intrusion Prevention Systems“ (IPS).

Im letzten Kapitel des Buches („Verschlüsselung und Zertifikate“, 36 Seiten) geht es um elementare

sicherheitsrelevante Themen. Nach einer Einführung in die Geschichte der Kryptologie wird umfassend die Verwendung einer eigenen Certificate Authority (CA) bis hin zu vollständig verschlüsselten Systemen mitsamt „Public Key Infrastructure“ (PKI) und X.509 beschrieben. Den Abschluss des Kapitels bildet ein Abschnitt „Rechtliches“, in dem auf die aktuellere Gesetzgebung rund um das Signaturgesetz und Signaturverordnung der BRD eingegangen wird.

Wie liest es sich?

Das 1163 Seiten starke Buch gibt einen wirklich umfassenden und sehr guten Einblick in das Thema „Linux-Server“, bei dem kaum ein Thema außen vor bleibt. Die einzelnen Kapitel lesen sich allesamt sehr gut, wobei der Grat zwischen zu wenig Informationen und zu vielen Details recht schmal ist. Obwohl man den einzelnen Kapiteln anmerkt, dass sie von verschiedenen Autoren ge-

schrieben wurden, stört dies den Lesefluss nicht. Erfahrene Administratoren finden eher wenig Neues in diesem Buch. Will man sich in eines der hier vorgestellten Programme tiefer einarbeiten, muss man ohnehin zu Spezialliteratur greifen.

LINKS

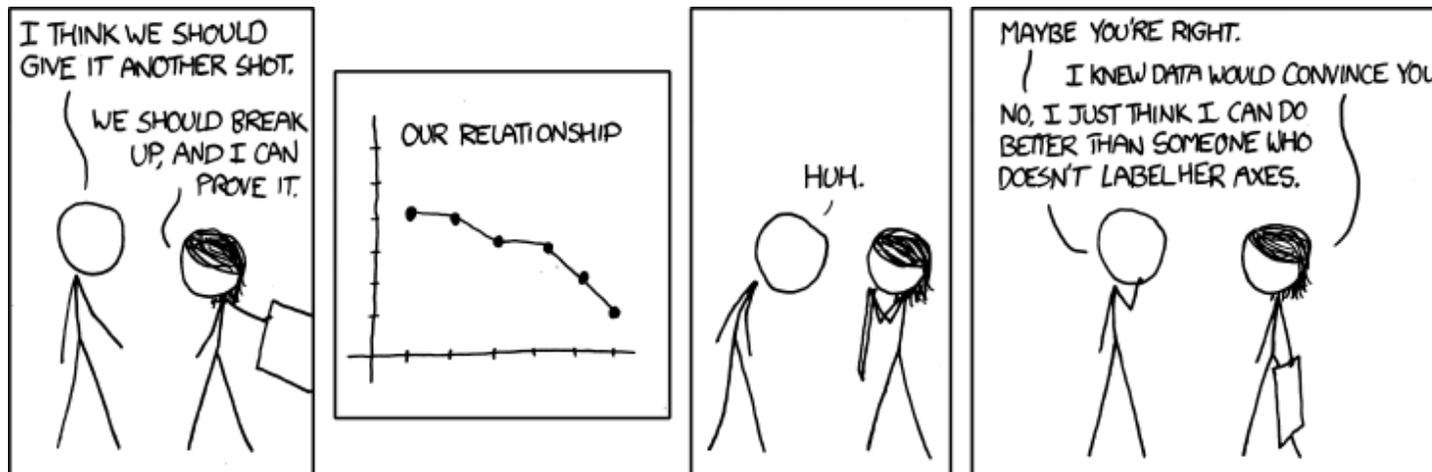
[1] https://www.rheinwerk-verlag.de/linux-server_3685/

Autoreninformation

Matthias Sitte ([Webseite](#)) ist Redakteur bei [freiesMagazin](#) und beschäftigt sich sowohl beruflich als auch privat seit vielen Jahren mit Linux

 [Teilen](#)

 [Kommentieren](#)



“Convincing” © by Randall Munroe (CC-BY-NC-2.5), <http://xkcd.com/833/>

Leserbriefe

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

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

Leserbriefe und Anmerkungen

openSUSE 42.1

❖ Nicht das erste, eher abschreckende Review für openSUSE 42.1, sodass ich vermutlich bis zum Schluss bei openSUSE 13.2 mit KDE 4 bleiben werde. Ich hoffe, die KDE-Entwickler geben nicht auf. Die Screenshots von Plasma 5 sehen vielversprechend aus. **Matthias** (Kommentar)

Kernelrückblick

❖ Vielen Dank für die neue Ausgabe. Ist mal wieder Topp. Danke an die Schreiber, die ihre Texte für [freiesMagazin](#) schreiben bzw. zur Verfügung stellen. Etwas enttäuscht war ich, dass kein Kernelrückblick von Herrn Menzer dabei ist. Hoffe, er ist in der nächsten Ausgabe wieder drin.

Anonym (Kommentar)

❖ *Ja, der Kernelrückblick für den Januar verspricht auch etwas umfangreicher zu werden als nur „RC drei, vier, fünf, sechs und sieben sind erschienen“ – Es ist auch die Acht dabei. ;-)*

Mathias Menzer

Let's Encrypt

❖ Danke für diesen tollen Beitrag! Besonders toll fand ich die Abschnitte mit einer beispielhaften Konfiguration und die Erklärung der einzelnen Konfigurationsoptionen. :) **Gast** (Kommentar)

❖ Ich wollte zu dem Artikel noch anmerken, dass der auf Seite 15 genannte Rewrite der URL von den Apache-Entwicklern nicht empfohlen wird [1]. Empfohlen ist dies über die Redirect-Direktive [2].

Christian

Linux Presentation Day

❖ Ein Kommentar aus Chemnitz. Weder der lokale Chaostreff noch die lokale LUG hat sich für die Teilnahme am LPD entschieden. Der Hauptgrund ist die Frage warum man gerade den LPD und nicht den Software Freedom Day [3], den Document Freedom Day [4] oder den 14. Februar (I love free software day [5]) unterstützen sollte. Ich persönlich veranstalte lieber zu einem der anderen Tage etwas. Das finde ich unterstützenswerter,

da nicht ein spezielles Softwareprodukt sondern eine Idee / Einstellung beworben wird.

txt.file (Kommentar)

Vi für typische nano-Nutzer

❖ Vielen Dank für diesen Artikel, denn er stimmt: Ich nutze nano und schrecke jedes mal auf, wenn auf einer fremden Shell kein nano sondern einzig vi(m) zu finden ist. Dank Eurem Artikel hat der/das Schrecken bald ein Ende: Ich will auch mit wachsen. **Matthias** (Kommentar)

LINKS

- [1] <https://wiki.apache.org/httpd/RewriteHTTPToHTTPS> 
- [2] <https://wiki.apache.org/httpd/RedirectSSL> 
- [3] https://de.wikipedia.org/wiki/Software_Freedom_Day
- [4] <http://documentfreedom.org/index.de.html>
- [5] <https://fsfe.org/campaigns/ilovefs/ilovefs.de.html>

 [Teilen](#)

 [Kommentieren](#)

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

Veranstaltungskalender

Messen

Veranstaltung	Ort	Datum	Eintritt	Link
Hack Im Pott	Essen	26.02.2016	13,37 EUR	https://events.ccc.de/congress/2015
Chemnitzer Linux-Tage	Chemnitz	19.03.–20.03.2016	–	http://chemnitzer.linux-tage.de/
Easterhegg	Salzburg	25.03.–29.03.2016	42 EUR	https://eh16.easterhegg.eu/
Linuxwochen Wien	Wien	28.04.–30.04.2016	frei	http://linuxwochen.at/
Grazer Linuextage	Graz	29.04.–30.04.2016	frei	http://www.linuxtage.at/
VCF Europa	München	30.04.–01.05.2016	–	http://www.vcfe.org/
LUG-Camp	Bruchsal	05.05.–08.05.2016	75 EUR	https://lugcamp.lugbruchsal.de/
Linuxwochen Eisenstadt	Eisenstadt	20.05.–21.05.2016	frei	http://linuxwochen.at/

(Alle Angaben ohne Gewähr!)

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

Vorschau

freiesMagazin erscheint am ersten Sonntag eines Monats. Die März-Ausgabe wird voraussichtlich am 6. März u. a. mit folgenden Themen veröffentlicht:

- Kurztipp: Freie Welten
- Rezension: GIMP 2.8 – Für digitale Fotografie und Webdesign

Konventionen

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

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

Impressum

freiesMagazin erscheint als PDF, EPUB und HTML einmal monatlich.

Kontakt

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

Autoren dieser Ausgabe

Hans-Joachim Baader **S. 11**
Robert Kurz **S. 26**
Mathias Menzer **S. 12**
Stefan Müller **S. 3**
Jochen Schnelle **S. 32**
Matthias Sitte **S. 37**
Christian Stankowic **S. 21**
Martin Stock **S. 15**
Dominik Wagenführ **S. 34, S. 35**

ISSN 1867-7991

Erscheinungsdatum: 7. Februar 2016

Redaktion

Christian Schnell
Matthias Sitte
Dominik Wagenführ (Verantwortlicher Redakteur)

Satz und Layout

Benedict Leskovar
Kai Welke
Thomas Schwemm

Korrektur

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

Veranstaltungen

Ronny Fischer

Logo-Design

Arne Weinberg ([CC-BY-SA 4.0 Unported](https://creativecommons.org/licenses/by-sa/4.0/))

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

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