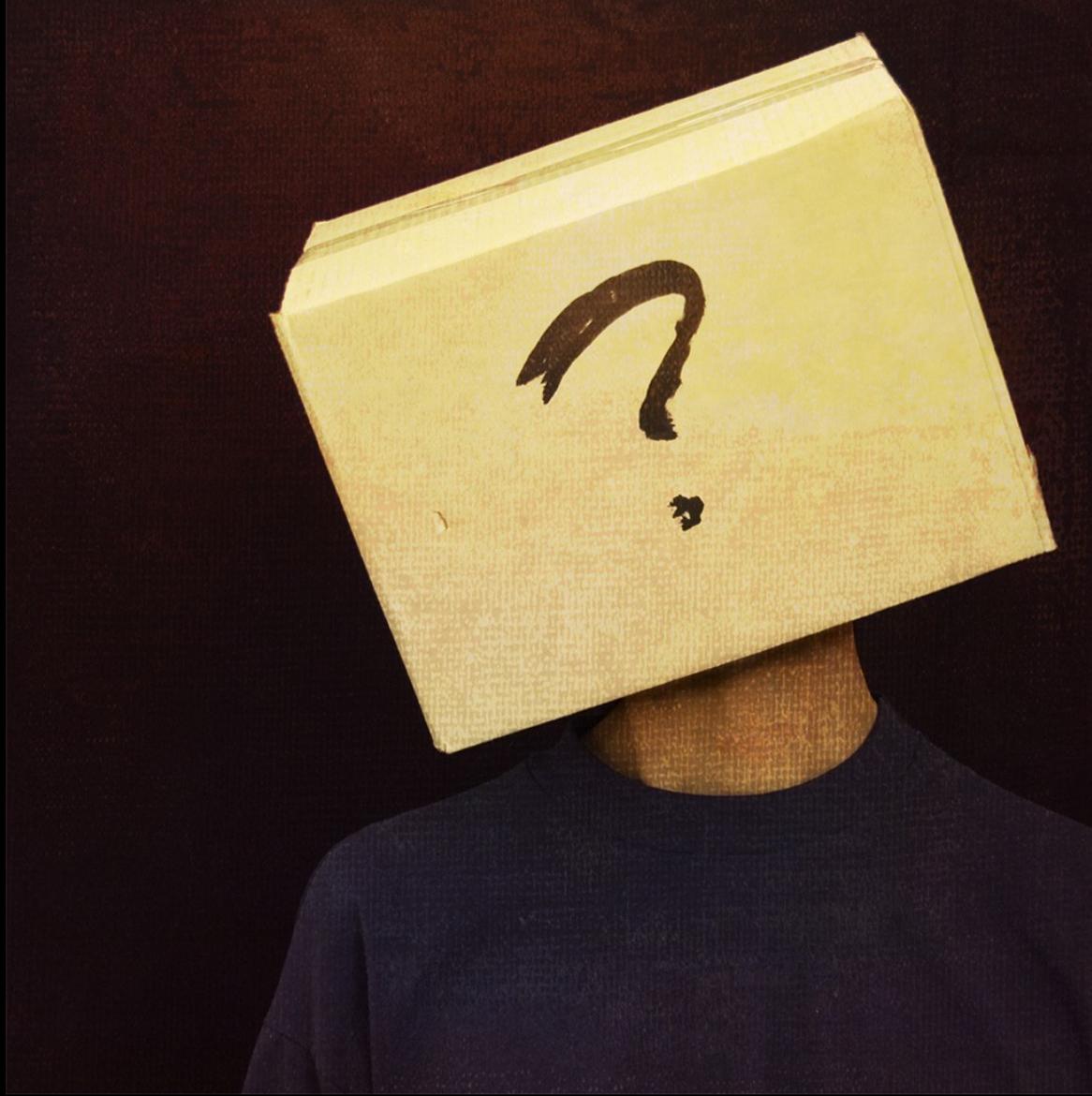


Thorsten Leemhuis

**Pakete und
Container: Was uns
Flatpak und Snap
einbringen**

%pre

%pre | whoami



%pre | whoami @Fedora



Thl [talk](#) [preferences](#) [watchlist](#) [contributions](#) [log out](#)

[user page](#)

[discussion](#)

[edit](#)

[history](#)

[move](#)

[unwatch](#)

WIKI

- [Fedora Project Wiki](#)
- [News](#)
- [Events](#)
- [Features](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

NAVIGATION

- [Home](#)
- [Get Fedora](#)
- [Join Fedora](#)

SUB-PROJECTS

- [Ambassadors](#)
- [Community Operations](#)
- [Design](#)

User:Thl

Thorsten Leemhuis

[\[edit\]](#)

Email: <fedora@leemhuis.info>

GPG Key: [4C583D2D](#) (run **gpg --recv-key 4C583D2D** to get it)

Jabber: thl_at_home@jabber.ccc.de

Google+: <https://plus.google.com/+ThorstenLeemhuis/>

Facebook: <https://www.facebook.com/thleemhuis>

Supported languages : de, en Living in: Hannover, Germany;

I was the first FESCo's chairman not working for Red Hat; that was about in 2006 till Fedora Core and Extras merged in early 2007, when I had to step down from the position due to conflicting interests with my day job. I then helped getting the EPEL effort running and served as the EPEL Steering Committee chairmen till February 2008. In parallel to my Fedora contributions within the Fedora project I did a lot of work for a well know 3rd-party add-on package repository and helped getting the predecessor running, which quickly become the unofficial official Add-On repository for Fedora in late 2008/early 2009.

These days mostly inactive in Fedora due to time constrains; I'm maintaining just a few Fedora packages that I need for work and maintain repositories with Fedora packages that contain



%pre | whoami @Fedora

- [Documentation](#)
- [EPEL](#)
- [Infrastructure](#)
- [Internationalization](#)
- [Localization](#)
- [Marketing](#)
- [Magazine](#)
- [Package Maintainers](#)
- [Quality Assurance](#)
- [Websites](#)
- [All projects](#)

SEARCH

Go

Search

TOOLS

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)

Package repository with Linux vanilla kernels for Fedora [\[edit\]](#)

This page contains information about a [set of repositories](#) which contain RPM packages with Linux vanilla kernels built for Fedora. 'Vanilla' in this scope means 'unmodified'. In other words: the sources used to compile those kernels come straight from kernel.org and do not contain any of the enhancements which the official Fedora kernels contain.

How to use these repos [\[edit\]](#)

How to use, the quick (aka TLDR) verison [\[edit\]](#)

Download the definitions for the Kernel vanilla repositories:

```
curl -s https://repos.fedorapeople.org/repos/thl/kernel-vanilla.repo | sudo
```

Run this to get the latest development kernel:

```
sudo dnf --enablerepo=kernel-vanilla-mainline update
```

You don't want to run a development kernel and want the latest stable kernel instead? Then run this:

```
sudo dnf --enablerepo=kernel-vanilla-stable update
```

Reboot. That's it - at least most of the time, as sometimes it's not that easy:

- Is UEFI Secure Boot active on your system? Then you have to disable it in your BIOS Setup to run kernels from these repos, as they are not signed with a key that a default Secure Boot setup



%pre | whoami @work



ct magazin für computer technik

28.12.2015

1



Smartwatches im Test

Richtig einstellen mit wenig Aufwand

Die Sicherheits-Checkliste

Handy, Router, NAS, Smart-TV, Server, Browser, Facebook ...

Upgrade auf Windows 10 stoppen
VR-Brille Samsung Gear
USB-Sticks mit Typ C
Fairphone 2 im Test
Spiele-Highlights 2015

Audi macht Ernst mit VR

Die Kabel-Deutschland-Lücke
Web-Typographie
Linux-Prozessmanagement
GUI mit Python
Skylake übertakten

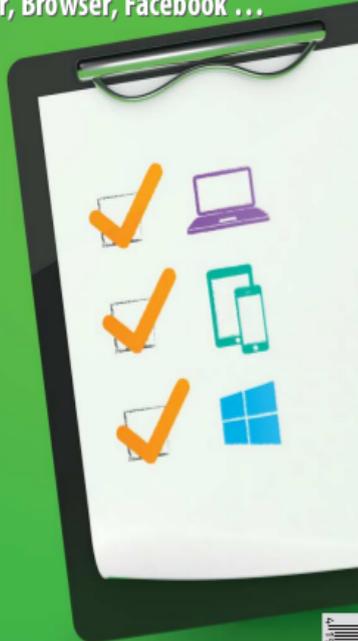
Alte und billige Tablets sinnvoll nutzen

Cooler Tablet-Projekte

Second Screen, Notenständer, Info-Display, Auto-Infotainment ...

€ 4,50

ISSN 1430-2013



Kernel-Log | Linux 4.4

Thorsten Leemhuis

Linux-Kernel 4.4

3D-Grafiktreiber für virtuelle Maschinen und mehr Sicherheit bei RAID 5

Der Linux-Kernel 4.4 wird einen Grafiktreiber für den Raspi mitbringen. Performance-Verbesserungen im Netzwerksystem sollen DDoS-Attacken erschweren. Zwei neue Ansätze versprechen High-End-SSDs mehr Leistung zu entlocken.

Das in der ersten Jahreshälfte erwartete Linux 4.4 erweitert die Fähigkeiten des Treibers Virtio-GPU. Dadurch sollen Linux-Distributionen, die in einer mit KVM betriebenen virtuellen Maschine (VM) laufen, in Zukunft die 3D-Beschleunigung des Wirtsystems nutzen können. Bei diesem über mehrere Jahre als „Virgl 3D“ entwickelten Ansatz reicht ein Gast-3D-Treiber die OpenGL-Befehle zur Ausführung an den Host weiter. Das Verfahren erfordert neben Linux 4.4 die Grafikbibliothek Mesa 11.1 und den Systememulator Qemu 2.5; beide sollen noch im Dezember erscheinen.

Virtualisierungs-Software von VMware oder VirtualBox ermöglicht schon länger eine Nutzung der 3D-Beschleunigung in Linux-VMs. Die VMware-Produkte sind allerdings proprietär und VirtualBox erfordert oft die mühsame Einrichtung passender 3D-Gasttreiber. Anders als Virgl 3D funktioniert der 3D-Support dieser Virtualisierungslosungen allerdings auch in VMs mit Windows. Selbiges gilt auch für das noch unfertige „KVMGT“ von Intel, bei dem die Gast-Treiber direkt auf Funktionen von Intel-GPUs zugreifen.

Raspi-Treiber

Linux bringt mit Version 4.4 erstmals einen Kernel-Grafiktreiber für die Broadcom-Prozessoren mit, die auf den verschiedenen Ausführungen des Raspberry Pi sitzen. Dieser Treiber funktioniert weitgehend autark und ist nicht auf den Grafiktreiber in der proprietären Firmware angewiesen, wie es bei derzeit meist eingesetzten Treibern der Fall ist. Der in 4.4 enthaltene und von Broadcom selbst vorangegangene Treiber beherrscht aber

bislang keine 3D-Beschleunigung; die sollen Verbesserungen ermöglichen, die in Linux 4.5 einfließen sollen.

Der MD-RAID-Code wird bei Software-RAIDs der Level 4, 5 und 6 ein Log führen können, das auf einem weiteren Datenträger liegt und Datenverfälschungen bei Systemabstürzen verhindert. Das zugrundeliegende Verfahren ähnelt dem von Journaling-Dateisystemen wie Ext4. Der Kernel schreibt jede Änderung zuerst in das Log und erst danach auf die am RAID beteiligten Datenträger. Falls die Stromversorgung beim Schreiben auf die RAID-Datenträger ausfällt, kann der Kernel die im Log hinterlegten Daten beim nächsten Start nutzen, um die Integrität innerhalb kurzer Zeit wiederherzustellen.

Das Log kann auch die Geschwindigkeit klein wenig steigern, da es Änderungen kurz puffert. Die Log-Funktion für MD-RAID stammt von Facebook-Mitarbeitern, die bereits an Erweiterungen arbeiten, die das Log zu einem vollwertigen Writeback-Cache machen. Dabei puffert das Log länger und mehr, was der Geschwindigkeit zugute kommt.

Offene SSDs

Neu ist auch Unterstützung für ein LightNVM genanntes Framework, das für „Open-Channel SSDs“ gedacht ist. Mit diesem Begriff bezeichnen die LightNVM-Entwickler einige vornehmlich für Server gedachte SSDs, bei denen das Betriebssystem einige Arbeiten übernehmen kann, die normalerweise der Flash Translation Layer (FTL) oder das Bad Block Management der SSD-Firmware erledigen. Das Delegieren ans Betriebssystem



soll die Geschwindigkeit steigern, denn das vermeidet nicht nur Overhead, sondern auch störende Wechselwirkungen zwischen SSD-Firmware und Betriebssystem. Derzeit gibt es aber nur eine Handvoll SSDs, mit denen das Ganze gelingt. Geschwindigkeitssteigerungen bei High-End-SSDs für Server verspricht auch eine neue, noch experimentelle Infrastruktur. Bei ihr nutzt der Kernel Polling, wenn es große Datenmengen mit besonders schnellen Datenträgern austauscht. Diesen Trick nutzen viele Netzwerk-Treiber im Kernel schon länger, denn bei der Verarbeitung riesiger Datenmengen macht das regelmäßige Abrufen neuer Daten beim Controller weniger Arbeit als die Abarbeitung der vielen Interrupts, die sonst auflaufen.

Linux 4.4 wird TCP-Handshakes schneller verarbeiten. Das reduziert Latenzen und erschwert zugleich DDoS-Attacken, denn der Kernel kann nun mehr Anfragen bedienen, bevor er unter hoher Last ins Straucheln gerät.

Die bessere Performance ist unter anderem einigen Optimierungen der Locking-Mechanismen im TCP-Code zu verdanken. Bei Tests durch den zuständigen Entwickler steigerten diese Änderungen die Zahl der per SYNACK hergestellten TCP-Verbindungen um das Zwei- bis Dreifache. Der Entwickler hat zudem noch einige Umbauten an Codepfaden für das SO_REUSEPORT-Flag vorgenommen, über das mehrere Anwendungen auf einem Port lauschen können; das konnte die Zahl der TCP-Handshakes noch mal nahezu verdoppeln.

Der neue Package-Loss-Algorithmus RACK (Recently ACK) soll die Geschwindigkeit von TCP-Verbindungen steigern, bei denen häufiger Netzwerkpakete verloren gehen. Dazu versucht RACK etwaige Paketverluste anhand der Übertragungszeiten anderer Pakete zu erkennen, und nicht anhand der Reihenfolge, in der sie eintreffen, wie es bisher

ge Algorithmen meist tun. RACK ist vorerst experimentell und stammt von Google. Das Unternehmen setzt den Algorithmus offenbar schon eine Weile ein und hat ihn bei der IETF zur Standardisierung eingereicht.

Unprivilegierte Anwendungen können erstmals mit dem eBPF (extensible Berkeley Packet Filter) ausgeführte Programme in den Kernel laden, um damit Datenströme zu verarbeiten, die durch den Kernel fließen. Dadurch kann beispielsweise ein nicht von Root ausgeführtes Tcpdump in Zukunft eBPF-Filter beim Kernel hinterlegen, damit der nur die Netzwerkpakete an den Sniffer weitergibt, die der Nutzer untersuchen will. Von unprivilegierten Anwendungen stammende eBPF-Programme unterliegen allerdings einigen Einschränkungen, damit Angreifer den eBPF-Interpreter nicht missbrauchen.

Unabhängig davon haben die Entwickler das Performance-Analyse-Werkzeug perf erweitert, damit es eBPF-Programme automatisch bauen, prüfen und in den Kernel laden kann. Der Kernel kann mit solchen Programmen interessante Events frühzeitig auffiltern, um Overhead und den Störinfluss der Analyse zu reduzieren.

Langzeit-Kernel

Facebook-Entwickler haben die Prozesslast beim Einsatz der Btrfs-Mount-Option ssd,spread reduziert. Zuvor hatten sie festgestellt, dass die darüber aktivierte Datenverteilungsmethode die Performance bei ihren Hardware-RAIDs der Level 5 und 6 ethelich verbessert.

Zu den neu zum Kernel stöbenden Treibern gehört einer für USB-WLAN-Chips von Realtek, für die es bislang nur einen Staging-Treiber gab, der größere Qualitätsmängel aufweist. Der Kernel 4.4 wird zudem einige per Firmware angepasene Sound-Chips besser unterstützen und spricht per I2S angebundene Audio-Chips von Skylake-Notebooks an. Der Wacom-Treiber steuert jetzt vier weitere Grafiktablets der Intuos-Serie an. Linux 4.4 wurde zudem vorab zu einem Longterm-Kernel erklärt. Daher soll es nicht nur knapp drei Monate, sondern bis mindestens Januar 2018 mit Fehlerkorrekturen und kleineren Verbesserungen versorgt werden. (thl@ct.de)

%pre | whoami @work

INFOS ZUM ARTIKEL

Kapitel

- 01 Geschwindigkeitswahl wird optimiert
- 02 Wartezeiten bei starkem Netzwerkverkehr reduzieren
- 03 EFI-Firmware spielt BIOS-Updates selbst ein
- 04 Parallele Verzeichnisabfrage
- 05 Kopierbeschleuniger für NFS
- 06 Knautschzone
- 07 Kernel abdichten
- 08 Unterstützung für rund fünfhundert weitere Geräte
- 09 Treiber für Thunderbolt, Xbox-Controller, Audio-Codexs, ...
- 10 Farbmanagement beim Intel-Treiber
- 12 Grafktreiber: GeForce 830M, Raspi, 5K-Display, ...
- 13 Unterstützung für Platten mit Shingled Magnetic Recording (SMR)
- 14 Mehrere Dateisystem-Bäume bei Ceph
- 15 Direktzugriff auf schnelle

Die Neuerungen von Linux 4.7 UPDATE



TRENDS & NEWS | KERNEL-LOG

 [Thorsten Leemhuis](#)

 30.05.2016

 [Kernel-Log](#), [Linux](#), [Linux 4.7](#), [Linux-Kernel](#)

Die neue Kernel-Version unterstützt AMDs neue Grafichips. Ferner soll Linux 4.7 das Stromsparerpotenzial moderner Prozessoren stärker ausschöpfen und Wartezeiten vermeiden, die bislang bei hoher Netzwerklast auftraten.

Linus Torvalds hat Linux 4.7 freigegeben. Eine der wichtigsten Verbesserungen der neuen Kernel-Version: Der Amdgpu-Treiber unterstützt jetzt die Polaris-Grafikprozessoren von AMD (u. a. [1](#), [2](#)).

www.heise.de/thema/Kernel_Log

Universalpakete

Flatpak und Snap sollen App-Installationen unter Linux revolutionieren



Linux-Distributoren versorgen ihre Anwender mit einer umfassenden Software-Ausstattung. Immer mal wieder fehlen aber Programme oder sind schrecklich veraltet. Flatpak und Snap versprechen dieses Manko mit Distributionsübergreifenden Paketen aus der Welt zu schaffen. Das soll Linux auch attraktiver für Entwickler machen.

Von Thorsten Leemhuis

Anwendungen unter Linux einzurichten kann kinderleicht sein: Software-Verwaltung öffnen, Programm installieren und starten – fertig. Aber wehe, dem eingesetzten Linux fehlt die gesuchte Anwendung, dann wird es schnell kompliziert. Dasselbe gilt, wenn die Paket-Repositories eine veraltete Version des Programms enthalten, man aber eine neuere braucht.

Flatpak und Snap versprechen, solche Probleme aus der Welt zu schaffen, indem sie Software mit allem zusammenpacken, was diese zur Ausführung benötigt. Die resultierenden Pakete laufen so unter verschiedenen Distributionen. Das

soll auch Entwickler anlocken: Sie können quell-offene oder proprietäre Linux-Programme jetzt selbst verteilen, ohne Pakete für Dutzende von Distributionen bauen zu müssen. Darüber hinaus versprechen die neuen Paketformate mehr Sicherheit, da sie Programme besser vom Betriebssystem und anderen Anwendungen abschotten.

Noch ist aber ungewiss, ob Flatpak oder Snap zum universellen Standard für „Linux Apps“ werden, wie es sich die Entwickler der beiden Ansätze vorstellen. Ohnehin unterscheiden sich die Herangehensweisen in wichtigen Punkten. Das ist einer von mehreren Gründen, warum wo-

%pre | this talk

**BTW, ja, das geht
in dem Tempo weiter
*(das hier ist Slide #10)***

%pre | this talk

**Daher: Fragen
aufschreiben!**

Wir haben viel vor uns

%pre | this talk

Vorweg ein Wort zum
Wort "Container" im
Vortragstitel...

%pre | this talk

Google

container



All Images Shopping News More

Settings Tools

View saved SafeSearch

home

kitchen

greenhouse

garage

tower

office

storage

water storage

shipping

packaging

interior

living

waste



%pre | this talk



container



All Shopping Images News Videos More Settings Tools

View saved SafeSearch

kitchen

plastic

glass

plastic bottle

metal

food waste

storage

shipping

water storage

recycling

square

jar

box



%pre | this talk



container



All Images Shopping News More

Settings Tools

View saved SafeSearch

packaging

plastic

glass

tin

aluminum

metal

composite

storage

shipping

food storage

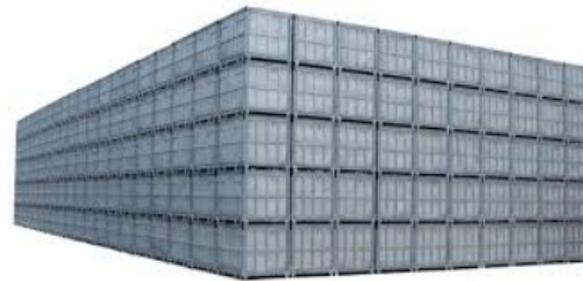
honey

spice

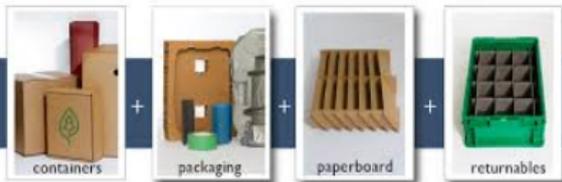
ice cream



450 x 341 - thomascontainer.com



+++ ALL IT TAKES TO GET IT THERE +++



Regular RSC



Rapid Packing Container



%pre | this talk

im Englischen ein
ganz unterschiedlich
verwendetes Wort

%pre | this talk

Google

container



All Shopping Images News More

Settings Tools

View saved SafeSearch

waste

plastic

steel

metal

glass

stainless steel

cardboard

chemical

liquid

solid

flammable

asbestos

solvent



%pre | this talk

Linux-Containern
ähnlich vielseitig
genutzter Begriff

%pre | this talk

**IOW: Denkt nicht zu
viel an Docker-
Container!**

Los jetzt: Was soll das
mit Flatpak & Snap

Motivation

revolutionieren, wie
Anwender Programme
einrichten

Motivation

Warum dieser
Revolutionsversuch?

Motivation

Programme einfacher
beschaffen

Motivation

Linux attraktiver für
Nutzer machen

Motivation

Programme einfacher
vertreiben

Motivation

Linux attraktiver für
Entwickler machen

Motivation

"Was soll das,
Distributionen
machen doch einen
guten Job"

Motivation

So einfach ist das
nicht; Beispiel:
LibreOffice 5.3

Motivation

Am 1. Februar
erschiene

Motivation

Unter Windows:
Herunterladen,
installieren, fertig

Motivation

Unter Linux? Tja...

Motivation

Update der Distribution
abwarten?

Quellcode kompilieren?

Pakete aus Copr/PPAs, Backports,
Entwicklungszweige?

DEB/RPM-Pakete der Software-
Herstellers?

Motivation

Wege, mit denen
unsereins sich oft zu
helfen weiß

Motivation

nicht simpel &
nicht universell

Motivation

abschreckend für
Neulinge & Windows-
Umsteiger

Motivation

und kosten unsereins
wertvolle Zeit

Motivation

Flatpak & Snap sollen
Software-Installation
so einfach wie bei
Windows machen

Motivation

Entwickler können
selbst Pakete
anbieten, die unter
verschiedensten
Distributionen laufen

Motivation



Google+



Linus Torvalds ▶ Public

Nov 25, 2015



This is just very cool.

I finally got around to play with the "ApplImage" version of [+Subsurface](#), and it really does seem to "just work".

It not only allows for a project to create a complex Linux application (in the case of subsurface, one that uses very recent library versions that many distributions don't even have available yet) that works on multiple distributions, you don't even need to really even install it. Download a file, mark it executable, and run it.

It comes with its own little embedded compressed ISO filesystem that gets mounted and contains all the required libraries.

Sure, it means that the end result is much bigger than a distro-native binary would be, but if you want a way to build applications for your users without limiting them to a particular distribution, or having to build fifteen different images, it really looks like it works very well.

Motivation

Nicht mehr Pakete für
Dutzende von
Distributionen bauen

Motivation

oder sich auf die
Paketbetreuer der
Distributionen zu
verlassen

*(die dann manchmal an alten
Versionen mit großen Fehlern
festhalten)*

Motivation

funktioniert auch mit
proprietärer Software

Motivation

erleichtert
Distributoren
die Arbeit

Motivation

Pakete für weniger
gängige
Anwendungen

*reminder: selbst Debian fehlt
hin und wieder mal wieder eine
Software*

Motivation

Sicherheitsgewinn

Motivation

Anwendungen und OS
trennen

Anwendungen
voreinander
abschotten

Motivation

Kein Ersatz für DEB
und RPM

Motivation

Flatpak und Snap sind
(derzeit) voll optional

Motivation

So weit zur Motivation
*Details sowie Vor- und Nachteile
folgen ;-)*

Herangehensweise von Flatpak und Snap

Herangehensweise

Software wird mit
allem
zusammengepackt,
was sie zur
Ausführung braucht
(außer dem Kernel)

Herangehensweise

ja, wie bei Docker
*(aber keine mehrschichtigen
Images)*

Herangehensweise

Flatpaks und Snaps
typischerweise via
Repositories/Stores

Herangehensweise

Kontakt mit einzelnen
Paketen möglich, aber
eher ungewöhnlich
(*"Flatpaks" bzw. "Snaps"*)

Herangehensweise

Downgrade im
Problemfall möglich

Herangehensweise

Isolation wichtiges
Design-Feature

*bei beiden aber noch
unvollständig!*

Herangehensweise

Beide brauchen
Client-Werkzeuge

im Unterschiede zu Applmage

Herangehensweise

das waren die
gemeinsamer Nenner

Herangehensweise

die wichtigsten
Unterschiede

Herangehensweise

Flatpak: Vorwiegend
für Desktop-
Anwendungen

Herangehensweise

Snap: Auch für
Kommandozeilen- und
Server-Programme
konkurriert also auch mit Docker

Herangehensweise

Flatpak:
System- oder User-
Install

Herangehensweise

Snap:
Immer Systemweit

Herangehensweise

Flatpak:
variabler Unterbau
(Runtime)

Herangehensweise

Snap:

Ubuntu Core als
Unterbau; optional:
"Platform Snaps"

Herangehensweise

Flatpak:
Jeder kann ein
Repository aufsetzen

Herangehensweise

Snap:

Sehr starke

Ausrichtung auf

zentralen Store

(ähnlich wie bei Android)

Herangehensweise

Flatpak:
von vornherein auf
Coss-Distri ausgelegt

Herangehensweise

Snap:
eher ein Ubuntu-Ding

Herangehensweise

Flatpak: Maßgeblich
von Red Hat/Fedora,
Endless und Gnome
vorangetrieben

Herangehensweise

Snap:
Von Canonical
dominiert

Herangehensweise

Flatpak:

in aktuellen Versionen
gängigen Distros oft
enthalten

Herangehensweise

Flatpak:

Verfügbar für Ubuntu

Herangehensweise

Snap:
so richtig nur in
Ubuntu

Herangehensweise

Verfügbar für weitere,
aber mit
Einschränkungen
(teilw. erheblichen)

Herangehensweise

Das waren die
wichtigen Eckdaten im
Groben

Details

Flatpak

Details Flatpak

Flat Pack Furniture

(denkt an Möbel von IKEA & Co.)

Details Flatpak

Älter als Snap &
früher mal "Xdg-App"

Details Flatpak

```
zthost@zthost:~$  
$ wget --quiet https://sdk.gnome.org/keys/gnome-sdk.gpg  
$ flatpak remote-add --user --gpg-import=gnome-sdk.gpg gnome https://sdk.gnome.org/repo/  
$ flatpak install --user gnome org.gnome.Platform 3.20  
9 delta parts, 71 loose fetched; 167139 KiB transferred in 37 seconds  
$ flatpak install --user gnome org.gnome.Platform.Locale 3.20  
4 delta parts, 55 loose fetched; 29195 KiB transferred in 11 seconds  
$ flatpak remote-add --user --gpg-import=gnome-sdk.gpg gnome-apps https://sdk.gnome.org/repo-apps/  
$ flatpak install --user gnome-apps org.gnome.Weather stable  
1 delta parts, 3 loose fetched; 5831 KiB transferred in 6 seconds  
$ flatpak run org.gnome.Weather  
Gtk-Message: Failed to load module "pk-gtk-module"  
Gtk-Message: Failed to load module "pk-gtk-module"  
(org.gnome.Weather.Application.2): Gtk-CRITICAL: glib_gio.c:2111: g_io_channel_unix_new: assertion 'fd >= 0' failed
```

Place

Hannover
Germany

Dienstag

Details Flatpak

Verifikations-
Schlüssel
herunterladen

Details Flatpak

Repository mit
Runtime hinzufügen

Details Flatpak

bindet vom Gnome-
Projekt betriebenes
Flatpak-Repository
lokal als "gnome" ein

Details Flatpak

Flatpak Runtime
installieren

Details Flatpak

installiert daraus die
Runtime
„org.gnome.Platform
3.20“

Details Flatpak

stellt eine Basisumgebung bereit, die von Gnome-3.20-Anwendungen typischerweise verwendete Bibliotheken (Glibc, OpenSSL, GTK+ ...) und Interpreter (Javascript, Python ...) enthält

Details Flatpak

passende
Sprachdateien
nachholen

Details Flatpak

Repository mit Apps
hinzufügen

Details Flatpak

Flatpak App
nachinstallieren

Details Flatpak

App aufrufen

Details Flatpak

Desktop-Dateien
abstrahieren
Startkommando

Details Flatpak

Grafisch noch
einfacher, aber noch
verbesserungswürdig

Details Flatpak

Support in Gnome-
Software

Details Flatpak

es gibt einfachere
Kommandos
mittlerweile

Details Flatpak

Alternative: "Bundles"

*wie klassische Pakete, die die
App enthalten*

Details Flatpak

ermöglicht Weitergabe
auf USB-Sticks & Co.

Details Flatpak

Runtime womöglich
trotzdem nötig

Details Flatpak

```
$ wget
```

```
https://download.documentfound  
ation.org/libreoffice/flatpak/latest  
/LibreOffice.flatpak
```

```
$ flatpak install --user --bundle  
LibreOffice.flatpak
```

```
$ flatpak run  
org.libreoffice.LibreOffice
```

Details Flatpak

Updates genau wie
sonst via Repository:
flatpak update

Details Flatpak

Ostree reduziert mit
Objekten (Git-Like)
Download-Umfang

Details zu Snap

Details Snap

Weiterentwicklung von
"snappy", dem
Paketsystem von
"Snappy Ubuntu Core"

Details Snap

Snappy \neq Snap

Details Snap

Snapcraft: Werkzeug
zu Bauen von Snaps

Details Snap

snapd: Daemon &
Software-Sammlung
(*die snap, snapcraft, snapd &
Co. enthält*)

Details Snap

Snapd: CLA

Details Snap

```
sudo snap install krita;  
krita
```

Details Snap

Install-Kommando lädt
Krita-Snap aus dem
Snap-Store herunter

Details Snap

Snap: kein
gesondertes

Kommando zum Start
der Anwendung nötig;

*Pfad entscheidet, ob via DEB oder
Snap installierte App startet*

Details Snap

Snapshots als Datei: nur
im unsicheren
Entwickler-Modus

Details Snap

Gnome-Software (aka
Ubuntu Software)
kann auch Snaps

Details Snap

Delta-Updates
reduzieren Download-
Umfang

Details Snap

Store: Von Canonical
betrieben und
kontrolliert;
proprietärer Code

Details Snap

Snap: Keys und Repo-URLs sind praktisch hardcodiert;

snap daher einfacher, obwohl ganz ähnliche Dinge passieren

Details Snap

den Firmen Red Hat,
Suse, GOG.com,
Valce, ... dürfte das gar
nicht gefallen
vielen in der Community
aber auch nicht

Details Snap

Snap attraktiver für
kommerzielle Anbieter

Details Snap

Store frei Haus

brauchen keinen

WebSPACE einrichten

Details Snap

Bezahlkram frei Haus

*(aber: Canoncial hält
die Hand auf)*

Details Snap

Ähnlich wie Apple und
Google will Canonical
hier einen Teil des
Umsatzes abgreifen

Details Snap

Apps müssen ein
Review über sich
ergehen lassen
(ähnlich Apple/Google)

Details Snap

Apps lassen sich von
Usern installieren

Details Snap

rudimentärer

Storeersatz

veröffentlicht

*unpraktikabel, weil snap nicht
darauf ausgelegt ist,
verschiedene Stores gleichzeitig
zu nutzen*

mehr zu den
Unterschieden von
Flatpack und Snap

Vergleich

Snap auch einfacher,
weil man keine
Runtime installieren
muss

Vergleich

Grundausstattung:

Das "OS Snap"

namens "Ubuntu Core"

*Standard-C-Library, OpenSSL und
andere Kern-Komponenten, die
Linux-Anwendungen in aller
Regel nutzen*

Vergleich

wieder: keine
Flexibilität; man muss
das nehmen, was
Canonical vorgibt

*theoretisch andere denkbar, mit
offiziellen Snap Store keine Option*

Vergleich

neuerdings: "Plattform
Snap" – Middlelayer,
das weitere Libs stellt
etwa KDE Frameworks

Bibliotheken

Bibliotheken

Runtimes/OS Snap:

Alarmglocken bei
alten Hasen!

Bibliotheken

Libc, libssl, zlib, ...
mehrfach auf dem
System!!1!1!!!

Bibliotheken

alles außer einem
modernen Kernel und
ein paar Kleinigkeiten
im Flatpak/Snap
genau wie bei Containern

Bibliotheken

eben diesen Bundling
ermöglicht die
Distributions-
Unabhängigkeit

Bibliotheken

genau wie bei
Containern: hat halt
auch Vorteile

Bibliotheken

ja, da kann man viel
drüber streiten

Bibliotheken

in einer besseren Welt
mit weniger
Distributionen ginge es
auch ohne
fürchte aber, von der
sind wir weit entfernt

Bibliotheken

mir scheint, wir sind
dieser Welt in den
letzten 10 Jahren auch
kein bisschen näher
gekommen (im
Gegenteil)

Bibliotheken

Diskussion dazu gerne
am Ende

Bibliotheken

Flatpak versucht das
Problem von Anfang
an zu entschärfen

Bibliotheken

Gnome Runtime

KDE Runtime

beide basieren auf der

Freedesktop-Runtime

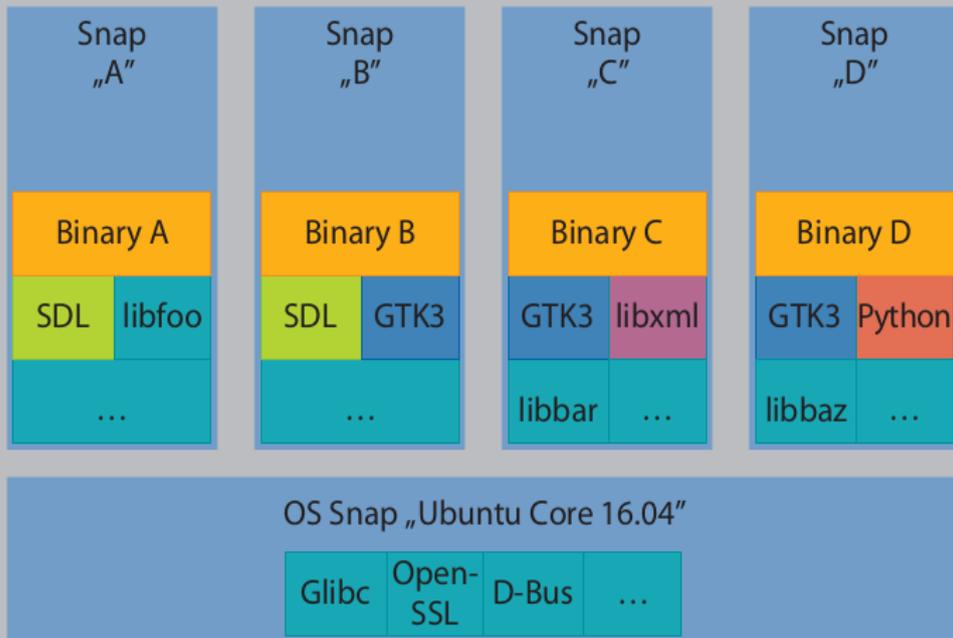
weitere denkbar...

Bibliotheken

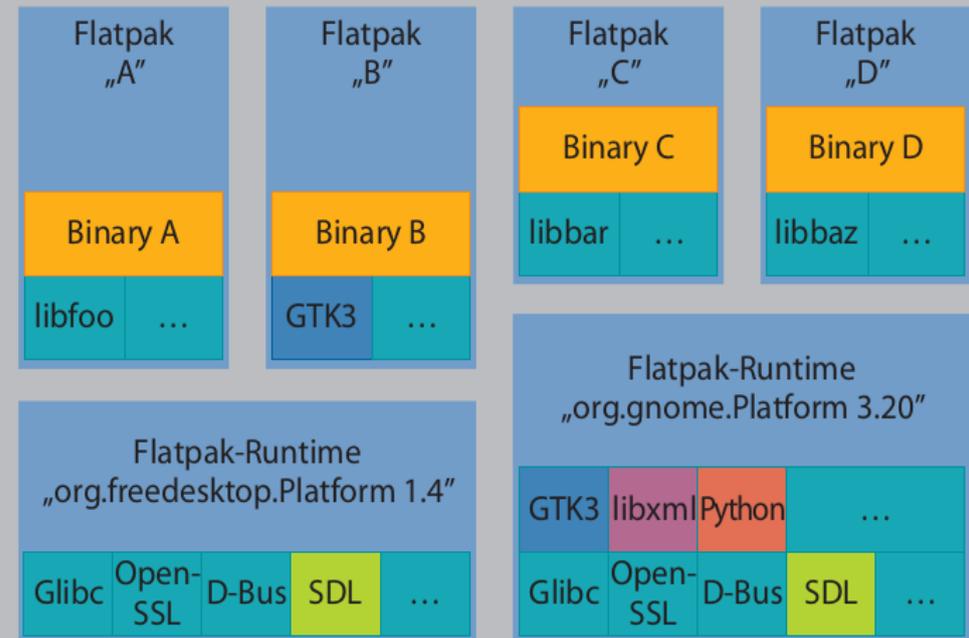
Alles Nötige dabei

Als Snap oder Flatpak verteilte Anwendungen bringen fast alles mit, was zur Ausführung unter modernen Linux-Kerneln erforderlich ist. In Kombination mit den Paketwerkzeugen laufen die Programme so unter verschiedenen Distributionen. Beide Paketformate verwenden typischerweise eine Laufzeitumgebung als Paketfundament, um Paketbau und Pflege zu erleichtern. Snaps nutzen in der Praxis immer ein Mini-Userland von Ubuntu als Basis, wohingegen jedermann Runtimes für Flatpaks erstellen kann.

Snap



Flatpak



Linux-Kernel

Bibliotheken

Plattform Snap
ermöglichen jetzt
ähnliches

Distributions- Unabhängigkeit

Distributionsunabhängigkeit

Client braucht
Werkzeuge

Distributionsunabhängigkeit

Flatpak in Arch Linux,
Debian Testing,
Fedora, openSUSE
Tumbleweed
enthalten

Distributionsunabhängigkeit

Flatpak bei openSUSE
Leap und Ubuntu
leicht nachrüstbar

Distributionsunabhängigkeit

Snap-Client in Arch
Linux, Debian Testing
und Ubuntu enthalten

Distributionsunabhängigkeit

Snap bei Fedora und
openSUSE
Tumbleweed
nachrüstbar

Distributionsunabhängigkeit

Ungefähr
Gleichstand?

Nur auf den ersten
Blick...

Distributionsunabhängigkeit

Snap braucht u.a.

gepatchetes

AppArmor für vollen

Funktionsumfang

Distributionsunabhängigkeit

Operating System	Status	Version	Notes
Ubuntu 14.04 LTS	Supported	2.21	
Ubuntu 16.04 LTS	Supported	2.21	
Debian (testing)	Supported	2.21	<i>devmode</i>
Debian (unstable)	Supported	2.21	<i>devmode</i>
Fedora 25	In progress	N/A	<i>devmode, no-classic</i>
CentOS 7	In progress	N/A	<i>devmode, no-classic</i>
RHEL 7.3	Unsupported	N/A	N/A
Arch	Outdated	2.16	<i>devmode</i>
Gentoo	Outdated	2.15	<i>devmode</i>
OpenSUSE Leap 42.2	Unsupported	TBD	N/A
OpenSUSE Tumbleweed	Unsupported	TBD	N/A
Yocto	Unsupported	2.21	<i>devmode</i>

<https://github.com/snapcore/snapd/wiki/Distributions>

Distributionsunabhängigkeit

Debian

Essential	Data
Version	2.21 (testing), 2.21 (unstable)
Confinement	none
Vendorized	no
Automatic CI	no
Bug reports	snapd

Snapd is maintained by the Debian packaging community. The primary contacts are *Steve Langasek* and *Michael Hudson-Doyle*. Snapd is available as the [snapd](#) package (together with some helper packages that the user is not expected to explicitly install).

Upstream development is not actively tested on Debian. This is tracked as <https://bugs.launchpad.net/snapd/+bug/1657101>

Debian doesn't support confinement at this stage. This is caused by non-upstreamed patches to the Linux kernel that expand what apparmor is capable of. Those patches are being continuously merged and it is expected that eventually Debian will be fully confined when the user chooses to enable Apparmor on their system, but this will not be ready for Stretch.

Distributionsunabhängigkeit

state of snapd support across distros

🕒 11 minute read ✍ Published: 13 Feb, 2017

> The story about getting snappy into other distributions

A while ago I compiled a [wiki page](#) that documents where are we with support of snapd on various popular distributions.

Unfortunately the table there may look a bit bleak. Apart from Ubuntu and Debian most of the other distributions don't ship snapd today. Since a few people were asking me about this (after all I was supposed to get this to work). I think some explanations are in order.

Preface

First of all, snapd development moves **fast**. There's a lot of things landing all the time. many of those non-trivial and introducing big changes. A few of those include

<https://new.zygoon.pl/post/state-of-snapd-support-across-distros/>

Distributionsunabhängigkeit

Letztlich: Ubuntu
einzigste Distro, wo
Snap richtig tut

Distributionsunabhängigkeit

Ubuntu Community Ask! Developer Design Hardware **Insights** Juju MAAS Partners Shop More ~

ubuntu[®] insights Cloud and server IoT Desktop Phone and tablet Press centre Search

Universal “snap” packages launch on multiple Linux distros

By Canonical on 14 June 2016

[f](#) [t](#) [g+](#) [✉](#) [in](#) [I](#) [♥](#)

Developers from multiple Linux distributions and companies today announced collaboration on the [“snap” universal Linux package](#) format, enabling a single binary package to work perfectly and securely on any Linux desktop, server, cloud or device. This community is working at [snapcraft.io](#) to provide a single publication mechanism for any software in any Linux environment. This release quotes Dell, Samsung, the Linux Foundation, The Document Foundation, Krita, Mycroft, Horizon Computing, contributors to Arch, Debian, OpenWrt, Ubuntu, and several of their

Sign up for email updates

Choose the topics you're interested in

- Cloud and server
- Desktop
- Phone and tablet
- Internet of Things



Your email

I would like to receive occasional updates from Canonical by email.

[Subscribe now](#)

<https://insights.ubuntu.com/2016/06/14/universal-snap-packages-launch-on-multiple-linux-distros/>

Distributionsunabhängigkeit

Flatpak hingegen tut
bei gängigen Distros

Distributionsunabhängigkeit

keine Abhängigkeit
von SELinux!

Paketangebot

Paketangebot

LibreOffice für beides
Verfügbar

Paketangebot

mehrere hundert
Snaps im Store

darunter auch Nextcloud

Paketangebot

Flatpak: verstreut,
schwer zu fassen

Paketangebot

beide: noch einige
Kinderkrankheiten &
Funktionslücken

Paketangebot

beide Ökosystemen
müssen noch reifen

Distributionsunabhängigkeit

Konverter denkbar,
aber mir noch nicht
untergekommen

Sicherheit

Sicherheit

Installation in eigene
Bereiche

Sicherheit

Abschirmung von
Host

Sicherheit

Abschirmung der
Apps gegeneinander

Sicherheit

volle Abschirmung
gibt es noch nicht

Sicherheit

Programme können
nach wie vor PINs und
TANs mitlesen

Sicherheit

das werden erst neue
Display-Techniken
lösen

Wayland/Mir

Sicherheit

außerdem noch neue
Audio- und Video-
Server angedacht

Sicherheit

Kommunikation mit
Host und anderer
Software?

Sicherheit

Snap: D-Bus

Sicherheit

Flatpak: Portale

*Berechtigungen ganz nebenbei
erteilen*

Sicherheit

Flatpak ruft externen
Datei-Öffnen-Dialog
auf

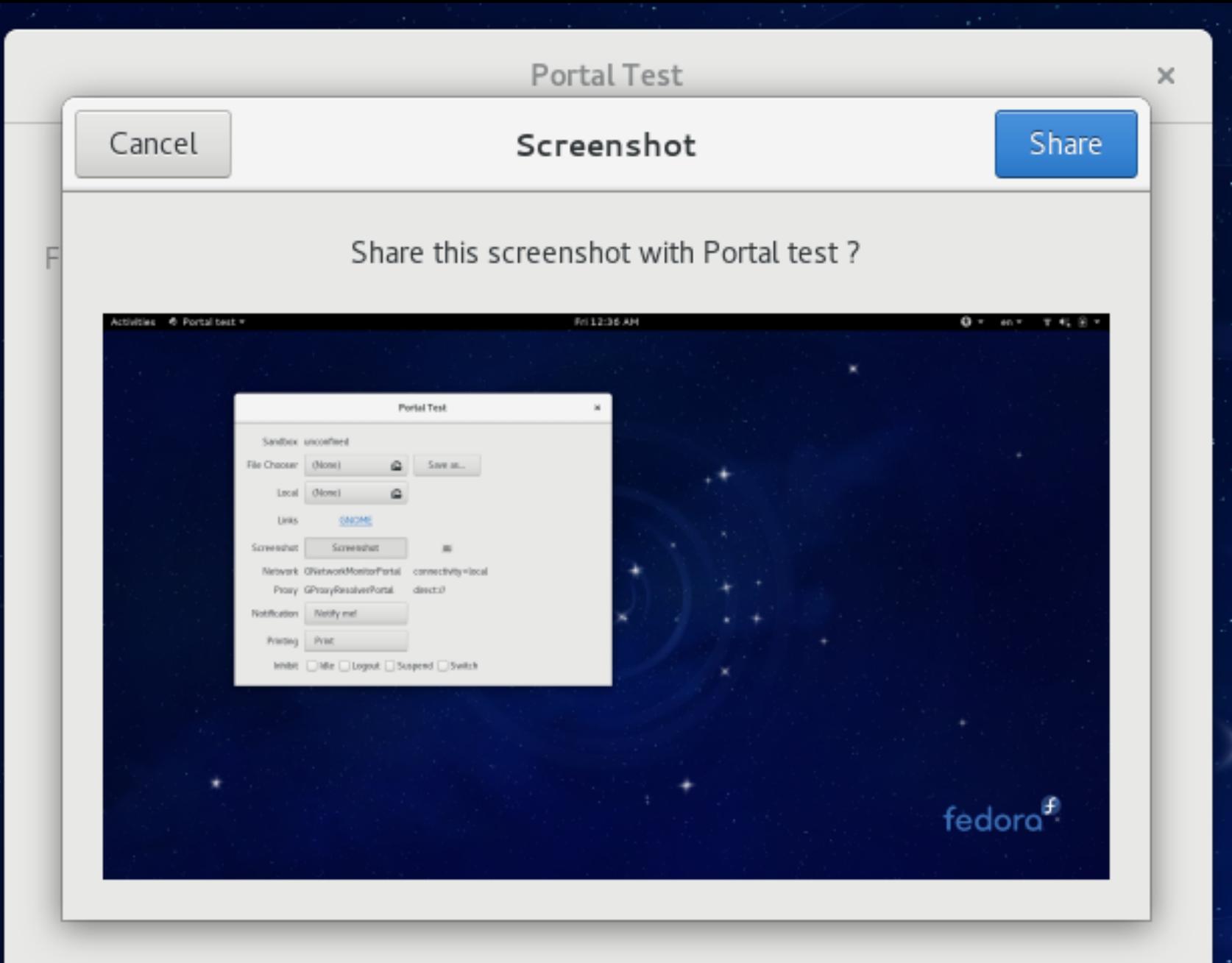
Sicherheit

Durch dessen
Verwendung wird
Zugriff auf Datei
erlaubt

Sicherheit

Teilweise aber doch
explizites Abnicken
nötig

Sicherheit



Konkurrenz mit RPM/DEB

Konkurrenz mit RPM/DEB

DEB- und RPM-Pakete
bleiben uns erhalten

Bau von OS-Images, Containern

*Vielleicht auch zum Bau von
Flatpaks & Snaps*

Konkurrenz mit RPM/DEB

Für Admins, die
Kontrolle über System
brauchen

Konkurrenz mit RPM/DEB

Fedora und Ubuntu
denken offen (aber
leise) darüber nach,
weniger gängiger
Apps rauszuwerfen

Konkurrenz mit RPM/DEB

Snap zentrale Rolle
bei Unity8

*"by 18.04 everything will be
Snaps and Unity 8 all the way
down"*

Konkurrenz mit RPM/DEB

Ubuntu Core:
Nur Snaps!

Konkurrenz mit RPM/DEB

Flatpak vielleicht
ähnlich wichtig bei
Fedora Atomic
Workstation

Konkurrenz mit RPM/DEB

Snap und Flatpak:

Noch nicht viel

Support in typischen

Admin-Tools

*kommt vermutlich mit
Marktdurchdringung*

Konkurrenz mit RPM/DEB

einige Details noch
unklar

Plugins packen?

Konkurrenz mit RPM/DEB

FOSDEM'17



About

News

Schedule

Practical



Brussels / 4 & 5 February 2017

[schedule](#)

[News](#)

[Sponsors](#)

[Contact](#)

[FOSDEM 2017](#) / [Schedule](#) / [Events](#) / [Main tracks](#) / [Cloud](#) / Resurrecting dinosaurs, what can possibly go wrong?

Resurrecting dinosaurs, what can possibly go wrong? How Containerised Applications could eat our users.

A Track: [Cloud](#)
📍 Room: [Janson](#)
📅 Day: [Saturday](#)
▶ Start: [13:00](#)
■ End: [13:50](#)

Containerised Application technologies like Appliance, Snappy and Flatpak promise a brave new world for Linux applications, free from the worries of shared libraries and dependency issues. Just one problem, this is a road long travelled before, such as in the application dark ages of Win32 applications and DLLs. And it worked out so wonderfully there... Do we risk a future where, like the resurrected dinosaurs of Jurassic Park, this family of applications will break their containment and start eating our users? This session will try to present a balanced argument about the situation, frankly discussing the benefits promised by these technologies, but highlighting the very real issues and risks their widespread adoption could, and in some cases are, already bringing to the table.

The talk will cover the promised benefits of application containers, such as Appliance, Snappy and Flatpak. It will detail the empowerment of developers who use the technologies, the ability for upstream projects to have a much closer role in delivering their

<https://fosdem.org/2017/schedule/event/dinosaurs/>

Andere Player

Andere Player

ganze Reihe ähnlicher
Projekte

Andere Player

ApplImage, (Limba),
ORB/Orbital Apps,
Subuser, Zero Install,

...

Andere Player

Applimage der
vielversprechendste
Konkurrent

Andere Player

Fun fact: enthält u.a.
Code eines Vorläufers
von Xdg-App/Flatpak

Andere Player

Linus Torvalds und
Dirk Hohndel nutzen
AppImage

Andere Player

Vorteil: Keine Client-
Tool nötig

```
chmod +x ./AppImagePaket;  
./AppImagePaket
```

Andere Player

Appliance-Pakete also
eher "Distributions-
Unabhängig"

Andere Player

Client-Tool verfügbar
(u.a. für Updates)

Andere Player

AppImages teilweise
kleiner als Flatpaks
und Snaps

Andere Player

keine Runtime/OS
Snaps

Andere Player

kein zentraler Store

Andere Player

Isolation kein Design-
Ziel von Applimage

Andere Player

mit Firejail und
Bubblewrap aber
möglich

*gute Integration wird es damit
nicht geben*

Andere Player

kein Rückenwind
durch große Player

Andere Player

ebensowenig wie die
anderen hier nicht
näher erläuterten
Lösungen

Andere Player

die sind durchaus
alltagstauglich,
konnten sich aber
nicht breit etablieren

Andere Player

Warum nicht Docker
nehmen?

Andere Player

andere Zielmarkt,
andere Ansprüche

Andere Player

Grafiktreiber
reinreichen?

Andere Player

Interaktion Ein- und
Ausgabegeräte?

Andere Player

IOW: da fehlt
sehr viel...

Andere Player

viele der genutzten
Basistechniken
ohnehin die gleichen

Summing things up

Fazit

DEB & RPM bleiben
uns erhalten

*(rücken aber vermutlich
mehr in den Hintergrund)*

Fazit

Die neuen Formate
bringen wichtige
Verbesserung

Fazit

"Linux-Apps"

Fazit

Sicherheit

Fazit

Applimage: eher
schlechte
Erfolgsaussichten

Fazit

Snap: ein Ubuntu-
Ding

Fazit

Flatpak: offen, wie es
sein sollte

Fazit

Snap: zerrt in ein von
Canonical dominiertes
Ökosystem

Fazit

Flatpak: offen, aber
dadurch auch
komplizierter

Fazit

Snap: Distributions-
Unabhängigkeit eher
Augenwischerei

Fazit

Flatpak: weitgehend
Distribution-
Unabhängig

Fazit

Beide: Sicherheit
noch WIP

Fazit

Ruhe bewahren?

Fazit

Fragen?

(TWIMC: dies ist Slide #227)

Thorsten Leemhuis

mail: linux@leemhuis.info,
thl@ct.de

GPG Key: 0x72B6E6EF4C583D2D

IRC@freenode.net: knurd

plus.google.com/ThorstenLeemhuis

#EOF