

Waydroid: Android Apps am Linux Desktop

Sollte der Fall eintreten, dass man eine App vom Android Telefon oder Tablet unbedingt auch am Linux Desktop einsetzen will oder gar muss, so bietet sich mit Waydroid eine einfache und effektive Lösung dafür an. Waydroid mountet in einem vorgefertigten Container eine vorgefertigte Android Instanz und reicht deren Anforderungen weiter an die sie umgebende Desktop-Umgebung einer Linux Installation, die auf systemd basiert. Harald Jele

Anwendungen »fremder« Betriebssysteme in unterschiedlichen Konstellationen unter Linux zu betreiben, mag eine von vielen Vorzügen dieses Systems sein. Der Eindruck dazu kann allein dadurch entstehen, wenn man die Vielzahl an vorhandenen Emulatoren und Ähnlichem betrachtet, welche unterschiedlichste Software unter Linux ausführen können, die dafür eigentlich gar nicht gemacht wurde. So führt Wine Windows Anwendungen aus, indem Systembefehle abgefangen und genau so an die Linux Installation weitergereicht werden, wie diese erwartet werden. Die DosBox emuliert neben dem Betriebssystem auch gleich einen Großteil der Hardware, die ein DOS System zu seinem ordnungsgemäßen Betrieb erwartet. Und mit Snes steht ein vielbeachteter Spielekonsolen Emulator bereit, der zum Einsatz kommt, wenn (ältere) Nintendo Spiele auf neuerer Hardware eingesetzt werden sollen. Die Liste dafür kann im Grunde beliebig fortgesetzt werden und eine entsprechende Suche im Web liefert unzählige weitere Ergebnisse dazu.

Zudem existiert eine deutliche Nähe zwischen den beiden Systemen. Der jeweils aktuelle Android Kernel ist ein von einem Linux Kernel mit Long Time Support (LTS) abgeleiteter, mit Patches versehener, der im Grunde mehr Ähnlichkeiten zum Original als Unterschiede aufweist. Und trotz der augenscheinlichen Nähe und der vielen Übereinstimmungen ist es im Detail aufwändig und schwierig Android Anwendungen nativ unter Linux zu betreiben. [1]

Mit Waydroid [2] betritt seit geraumer Zeit eine Software die Open Source Welt, mit der es sich ihre Macher zum Ziel gesetzt haben, Android Anwendungen möglichst einfach und flexibel ins Linux Universum zu integrieren. Dabei wird auf einen Ansatz zurückgegriffen, der nicht völlig neu ist und auch für diesen Fall nicht von null auf neu entwickelt werden musste. Denn mit der Software Anbox wurde bereits vor Jahren (2017) ein sehr ähnlicher Weg bestritten, der jedoch nicht bis ins Heute weiterverfolgt wurde. Die Weiterentwicklung von Anbox wurde im heurigen Jahr schließlich abgebrochen.

Waydroid (wie weiland Anbox) basiert auf einer Containerlösung, innerhalb dessen ein Session Management ein Android Image mountet und anschließend startet. Zur Verfügung stehen davon zur Zeit zwei. Eines mit den zentralen Google Apps und eines ganz frei davon. Die Bezeichnungen da-

für sind sprechend gewählt: »GAPPS« und »VANILLA«. Beide sind Abkömmlinge von LineageOS (aktuell 18.1-20230422) und entsprechen einem Android 11. Ein Updatemechanismus ist für beide integriert, sodass diese aktuell gehalten werden können.

Installation

Um Waydroid in der aktuellen Version (1.4.1) in ein Ubuntu 22.04 LTS zu bringen, sind im Grunde nur einige wenige Schritte notwendig. Alle davon sind auf den Web-Seiten des Projekts ausführlich beschrieben und zudem einfach nachzuvollziehen. [3]

In einem ersten Schritt werden, für den Fall dass diese noch nicht Teil der vorhandenen Installation sind, zwei der im Weiteren benötigten Pakete installiert.

```
sudo apt install curl ca-certificates -y
```

Anschließend wird das offizielle Repositorium des Projekts zu den lokalen Quellen für apt hinzugefügt, um darüber auch in Zukunft am aktuellen Stand zu bleiben.

```
curl https://repo.waydro.id | sudo bash
```

Aus diesen Quellen erfolgt im Anschluss die Installation der Anwendung in der aktuellen Version.

```
sudo apt install waydroid wl-clipboard -y
```

Damit liegen auch schon die notwendigen Programmteile innerhalb der Linux Installation. Der nachfolgende Befehl weist den System- und Sitzungs-Manager von Ubuntu (Systemd) an, den Waydroid Container beim Hochfahren des Betriebssystems automatisch zu starten.

```
sudo systemctl enable --now waydroid-container
```

Der erste Start

Beim Start von Waydroid erkundet dieses seine Konfiguration und wird somit vor dem ersten Start feststellen, dass dem Programm bislang noch keine Android Instanz hinzugefügt worden ist. Dementsprechend meldet sich Waydroid mit einer gra-

fischen Aufforderung, sich für eine der beiden verfügbaren Instanzen zu entscheiden (siehe [Abb. 1](#)).

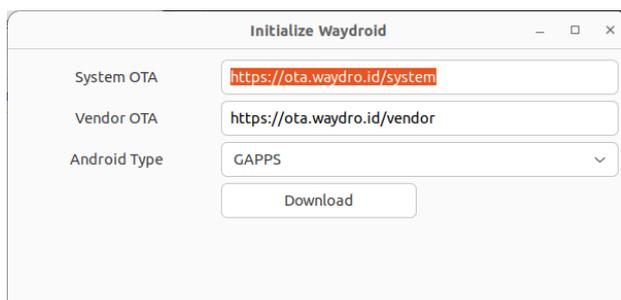


Abbildung 1: Die Wahl eines Android Images bei der Installation.

An dieser Stelle wählt man entweder die bereits erwähnte Google freie »VANILLA« oder aber »GAPPS«, die für die nachfolgende Integration ins Google Versum vorbereitet ist.

Sollte man seine Meinung im Lauf der nachfolgenden Arbeit ändern und doch das jeweils andere Android Image einsetzen wollen, so bemüht man dazu Waydroid auf der Kommandozeile, indem dieses mit »init« angewiesen wird, das andere zu laden und zum Mounten und Starten vorzubereiten.

```
sudo waydroid init -s SYSTEM_TYPE (VANILLA|GAPPS)
```

lautet dazu das passende Kommando.

Leider kann Waydroid bislang bloß eine Android Session innerhalb eines Containers ausführen, sodass man das Image, das man bereits geladen hat, am besten umbenennt, bevor man es durch das Downloaden des anderen überschreibt. Zudem kennt der Startbefehl einer Waydroid Session momentan keinen Parameter, mit dem man ein bestimmtes Image angeben könnte, sodass mehrere nebeneinander in der lokalen Installation vorhanden sein und ein bestimmtes davon gemountet und gestartet werden könnte.

Die Images einer Waydroid Session liegen im Verzeichnis

```
/var/lib/waydroid/images/
```

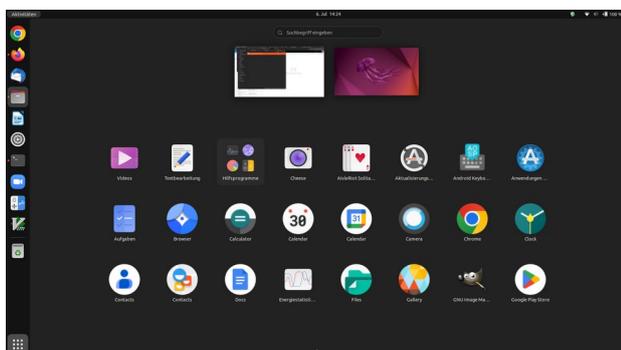


Abbildung 2: Android Anwendungen im Programmstarter von Ubuntu.

Die Dateien system.img und vendor.img sollten im Fall des Falles umbenannt werden, um sie vor dem ungewollten Überschreiben durch ein neues zu retten.

Fertig, um loszulegen

Mittlerweile ist die Installation soweit abgeschlossen, sodass erste Android Apps gestartet werden können. Diese sind im Startordner von Ubuntu unter die nativen Linux Anwendungen gemischt ([Abb. 2](#)).

Wer das Google Image startet, muss dieses bei Google noch registrieren (Google Play Certification), um in den Genuss von Google Play ohne weitere Einschränkung zu gelangen.

Dazu wird (1) eine Waydroid Shell gestartet, anschließend (2) mit dem nachfolgenden Befehl die Device ID ermittelt und diese (3) bei Google registriert.

- (1) `sudo waydroid shell`
- (2) `ANDROID_RUNTIME_ROOT=/apex/com.android.runtime
ANDROID_DATA=/data
ANDROID_TZDATA_ROOT=/apex/com.android.tzdata
ANDROID_I18N_ROOT=/apex/com.android.i18n
sqlite3 /data/data/com.google.android.gsf/databases/gservices.Db "select * from main where name = \"android_id\";"`
- (3) `https://www.google.com/android/uncertified`

Der umständlich und lang gehaltene Befehl aus (2) lässt sich aus der hier genannten Web-Seite [\[4\]](#) zum einfachen Kopieren entnehmen.

Die Registrierung dauert nach der Anmeldung bei seinem Google Konto in der Regel einige Sekunden. In den einschlägigen Foren wird jedoch darauf hingewiesen, dass dies auch ein paar Minuten dauern kann. Nach dem Registrieren ist ein Neustart der Waydroid Session notwendig.

```
waydroid session stop|start
```

erledigt diesen Schritt.

Im Android Universum angekommen

[Abb. 2](#) zeigt einen Ausschnitt der mit dem Google Image installierten Android Anwendungen. Da diese im Anwendungsstarter den Icons der nativen Installation untergemischt sind, müssen diese dort genauen Auges erkundet werden, sofern sie nicht ohnehin bekannt sind. Will man eine Übersicht darüber erlangen, welche Android Apps mit dem aus-

gelieferten Image installiert wurden, geht dieser Vorgang auf der Kommandozeile deutlich eleganter, denn Waydroid bietet einen Befehl, um die installierten Apps aufzulisten.

waydroid app list

liefert ein vollständige Aufzählung aller installierten Apps. Die Einträge dieser Liste können im Weiteren auch zum Starten einer Android Anwendung von der Kommandozeile aus genutzt werden.

```
Name: Docs
packageName: com.google.android.apps.docs.
  editors.docs
categories: android.intent.category.
  LAUNCHER
```

lautet zum Beispiel der Eintrag für ein installiertes Google Docs. Dieses kann im Terminal wie folgt gestartet werden:

```
waydroid app launch com.google.android.apps.
docs.editors.docs
```

Android Apps können somit wie am Handy oder Tablet über Google Play und die Google Einstellungen (Settings → Apps), sowie mit Waydroid zudem über den Ubuntu Anwendungsstarter oder eben auch aus einem Terminal heraus gestartet werden. Für jede dieser Möglichkeiten gibt es ihren Grund und jeder Android / Waydroid Benutzer ist sicher bereits in die Lage versetzt wurden, eine bestimmte davon (und in der spezifischen Situation eben nicht eine andere) zu nutzen.

Anwendungen lassen sich im Weiteren wie gewohnt über Google Play installieren. Wer F-Droid als eine weitere Quelle verwenden oder hinzufügen will, macht das in gewohnter Form. Zudem bietet Waydroid eine weitere Möglichkeit, Anwendungen direkt im apk Dateiformat zu nutzen.

waydroid app install appname.apk

ist eine Möglichkeit, eine Installation eines Android Pakets innerhalb eines Terminals auszulösen.

Wer sich im Weiteren auf reges Installieren einlässt, wird früher oder später feststellen, dass es an der einen oder anderen Stelle des Projekts noch ein wenig hakt.

Zu den lästigsten Problemen zählt, dass das Rotieren der Anzeige die eine oder andere Anwendung stolpern lässt. So sind nicht alle Anwendungen für die Anzeige im Querformat geeignet und verhalten sich entsprechend störrisch. Dies ist keine Eigenheit von Waydroid, denn im Grunde scheitern solche Apps auch nativ am Mobiltelefon oder Tablet. Hinzu kommt jedoch der störende Umstand, dass in einem solchen Fall eine Hälfte der Anzeige mit der Maus nicht mehr erreichbar ist, da

Android für die Verwendung dieser nur die Breite des Hochformats anbietet. In früheren Versionen von Waydroid blieb dieser Bereich einfach schwarz. In der aktuellen ist die Ansicht in Ordnung, jedoch weiterhin nicht benutzbar. In **Abb. 3** sieht man eine solche Anwendung sowie den (in der Abbildung sehr klein dargestellten) Mauszeiger am äußerst rechten Rand der mit ihr ansteuerbaren Fläche.

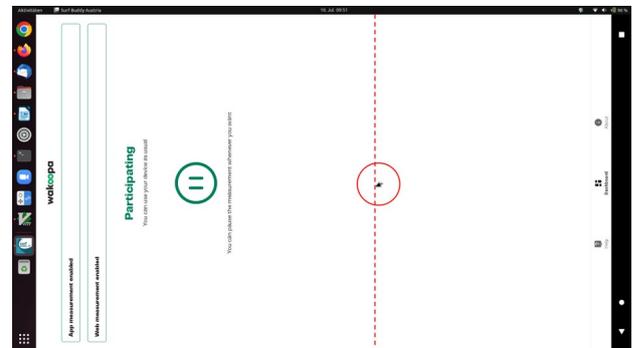


Abb. 3: Beispiel einer Anwendung, die im Querformat nicht benutzbar ist.

Vielfach hilft dann auch nicht der Umstand, die Anwendung durch ein manuelles Rotieren in die richtige Ausrichtung zu versetzen. Das manuelle Übermitteln der Anzeigegeometrie ist zwar möglich, nur stört ein solches dann zugleich alle anderen Anwendungen.

Die einfachste Lösung dieser Kalamitäten ist die Aktivierung des Mehrfensterbetriebs der Anzeige, bei dem alle Anwendungen standardmäßig im Hochformat angezeigt werden. Dies gelingt mit dem nach folgenden Kommando und führt zur Anzeige in **Abb. 4**.

```
waydroid prop set persist.waydroid.multi_windows
true
systemctl restart waydroid-container.service
```

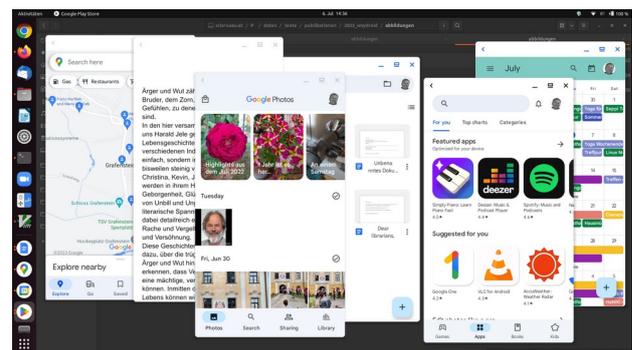


Abb. 4: Android Apps im Multi Windows Mode von Waydroid.

Weitere Probleme können momentan noch der Betrieb der Kamera, der Lautsprecher und des Mikrofons bereiten. Zwar ist man bei Waydroid sehr darauf bedacht, die Standards von Linux möglichst gleichförmig in den Container durchzuschleusen

bzw. in gleicher Weise zu nutzen, trotzdem bleibt es momentan noch ein durchaus spannendes Glücksspiel, ob dieser Teil der Hardware richtig angesprochen wird. Gerade dem Zugriff auf die Kamera wird durch die Entwickler schon seit geraumer Zeit viel Aufmerksamkeit eingeräumt, sodass wesentliche Fortschritte bereits in naher Zukunft erwartbar sind.

Kommandozeilig

Die Möglichkeiten, Waydroid über die Kommandozeile zu bedienen und zu konfigurieren sind mannigfaltig und da die Software in vielen Bereichen problemlos ihren Dienst verrichtet, bleiben viele dieser Optionen quasi unsichtbar im Verborgenen. Wer den einzelnen Diensten, die Waydroid auch nach außen hin anbietet, auf die Finger schauen will, findet unter [5] die umfangreichen Funktionen gut dokumentiert.

Wer von Anfang an einen Eindruck davon erhalten möchte, lässt sich via

```
waydroid status
```

den aktuellen Zustand berichten. Die Antwort darauf sieht in der Regel wie folgt aus:

```
Session: RUNNING
Container: RUNNING
Vendor type: MAINLINE
IP address: 192.168.240.112
Session user: admunix(1000)
Wayland display: wayland-0
```

Ist keine Session innerhalb des Containers aktiv, kann man diese mit

```
waydroid session start|stop
```

in den jeweils anderen Zustand bringen. Wenn keine Session läuft, man aber eine Android App startet, so wird eine solche (nämlich eine Session) mitgestartet.

Sollte gerade kein Container aktiv sein, so geht man mit

```
sudo waydroid container start|stop|restart|freeze|unfreeze
```

ans Werk. Für Neugierige und jene, die einem Problem auf die Spur kommen wollen, ist ein Blick in die Log-Datei vielleicht von Interesse. waydroid.log liegt im Verzeichnis

```
/var/lib/waydroid
```

Fazit

Mit Waydroid ist eine sehr flexibel gehaltene Software am Open Source Markt vorhanden, die Android Anwendungen gelungen in eine Linux Installation integriert. Apps lassen sich damit in gewohnter Weise wie am Handy als auch am Tablet installieren. Mit dem Wl-Clipboard existiert zudem eine Möglichkeit, Daten zwischen dem nativen Linux und den Android Anwendungen im Container auszutauschen.

Die Software integriert sich gut in den Desktop einer Ubuntu Installation, sodass sich deren Bedienung kaum von den nativen Linux Anwendungen unterscheidet.

Sollte in naher Zukunft der Zugriff auf die Kamera als auch auf Mikrofon und Lautsprecher zuverlässiger gelingen, bleiben nur noch wenige Hindernisse bestehen, Apps, die sonst nur am Handy und Tablet verfügbar sind, auch innerhalb einer Linux-Installation vollwertig einzusetzen.

Quellen

[1] Der gemeinsame Android-Kernel.

[<https://source.android.com/docs/core/architecture/kernel/android-common?hl=de>]

[2] Waydroid. Android in a Linux container.

[<https://waydro.id/>]

[3] Waydroid Instructions: Quick install reference for systemd distributions.

[<https://waydro.id/#install>]

[4] Google Play Certification.

[<https://docs.waydro.id/faq/google-play-certification>]

[5] Waydroid command line options.

[<https://docs.waydro.id/usage/waydroid-command-line-options>]

Der Autor



Dr. Harald Jele ist Mitarbeiter an der Universität Klagenfurt. 1993 stieß er durch einen glücklichen Zufall auf Linux. Seitdem kann er sich dieses gleichermaßen am Server wie am Desktop nicht mehr wegdenken.