How Replicant uses Guix?

Denis 'GNUtoo' Carikli

September 17, 2022

Introduction

In this presentation we will talk about:

- Why it's important to support smartphones with free software
- The architecture of Android and smartphones (this introduce the next presentation)
- Where Replicant uses or doesn't use Guix and why

Presentation structure:

- Introduction (this section)
- Why caring about smartphones?
- Smartphone hardware and status
- What is Replicant?
- Building Replicant and Guix.
- Running Guix on top of Replicant?
- Automatic Testing
- Replicant infrastructure.

Why caring about smartphones?

Why people have smartphones?

- Pressure to be reacheable at all times?
- Mobile computing?
- Mobile banking?
- Cheap computer?
- Add your ideas here.

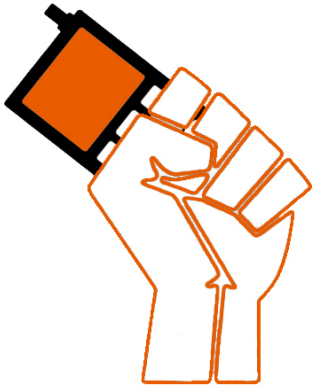
Main issues with smartphones

- Making smartphones destroys people (bad work conditions) and the planet (manufacturing).
- The network knows the smartphone location.
- Runs nonfree software.
- Unclear if empowers people or not (depends users freedom, use cases, etc).
- Add your issues here.

Solution:
Not use smartphones at all
$\rightarrow$ No need to fix the issue

Doesn't work?

- At least 2.7 billion people still using it, including people at risk like:[**?**]
  - Political activists
  - Journalists
  - etc
- Or indigenous people in Oxaca for their security



And yet:

- Making smartphones destroys people (bad work conditions) and the planet (manufacturing)
- The network knows the smartphone location
- Runs nonfree software

How to fix that?

- Destroy civilization, capitalism, etc? → Does that require smartphones that run free software in the first place?
- Destroy all factories? → Not my area of expertize. People also do depend on smartphones and that cannot work without big support from people.

Free software to advance in the right direction

- $\rightarrow$ We need free software OS on smartphones too.

Other ways to help getting there:

Political pressure:

- Practical right to repair, devices that last longer → Easier to support with free software.
- Against DRM: Corelation between DRM and the unability for users to completely control their device (can't change the nonfree bootloader, TrustZone OS, etc).
- Practical right to install your own OS.

Other ways to help:

- Funding work (Nlnet, etc) and helping projects that do that in sustainable ways (Work to upstream support for devices).
- Connecting with other struggles (anti-planned obsolescence, environmental movement, people that destroy factories if that exists, etc) to go in the right direction and getting stronger together instead of fighting each other.

Have clear demands and attainable goal: First steps

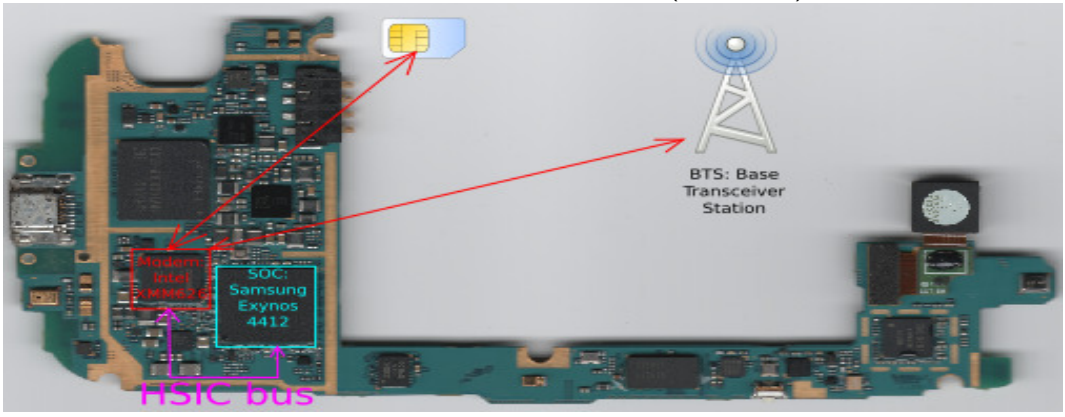
- 100% free sofware on the main CPU
- 100% free sofware distribution(s)
- Usable by people:
  - Easy to use if possible
  - Hardware easy to find
  - Hardware that can last and that is not too expensive
- Limits the damage:
  - Add some limits to the invasion of intimacy.
  - Hardware lasts longer.

How Replicant achieve that?

- → Lot of work by many people over many years.
- → Not shipping nonfree firmwares or any nonfree software.
- → Collaborating with other distributions if possible.

Smartphone hardware and status

Example smartphone: Galaxy SIII (GT-I9300):



- System on a chip
- Smartphones and the (isolated) modem

Additional particularities:

- Very small display with very high DPI/PPI
- Big fingers (cannot click on small close window buttons)
- No hardware keyboard (cryptsetup, games)

Easily available smartphones today:

Smartphone	WiFi	Boot	Modem	Battery
Second hand GT-I9300	nonfree firmware	nonfree, signed	Isolated, free drivers	Removable
Librem 5	firmware on flash chip	nonfree DDR4 firmware	can be isolated, free drivers	Removable
Pinephone	nonfree firmware	free	can be isolated, free drivers	Removable
Exynos / Qualcomm SOC	nonfree firmware	nonfree, signed	shared memory, free drivers possible	non replaceable

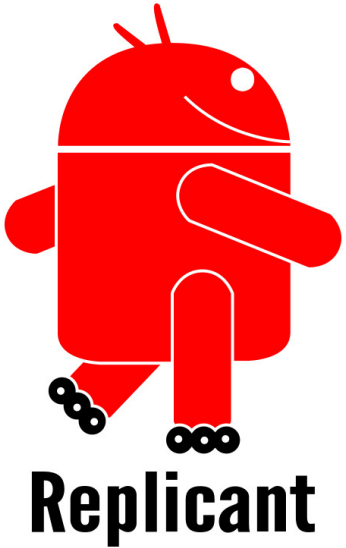
What we have now: FSDG Distributions

Distribution	OS	Smartphones support	Shortcommings
Guix	GNU/Linux	Missing packages	/gnu size?, requires to know lisp (Fix WIP?)
Parabola	GNU/Linux	Missing packages	Installation, rolling release
PureOS	GNU/Linux	Supports the Librem5 well	rolling release, 1 device only?
Replicant 6	Android	Support 10 devices	Security issues, old, not sustainable
Replicant 11	Android	Work in progress	Not ready
Trisquel 10	GNU/Linux	Missing packages	Installation, Applications to adapt

What is Replicant?

Replicant:

- Fully free Android distribution approved by the FSF
- But the hardware it runs on is not...



Quick Facts

- Website: replicant.us
- Exists since September 2009
- Two main versions:
  - Replicant 6:
    - Android 6, last security update: October 2017 [**?**]
    - based on LineageOS
    - Supports $\sim$ 10 devices (smartphones and tablets)
  - Replicant 11:
    - Android 11, work in progress[**?**]
    - based on official Android source code
- $\sim$ About 1 full time equivalent contributors (sometimes more) and a community of users and contributors.


SAMSUNG
11:56


12:14


SAMSUNG
2:34


SAMSUNG
19:30


SAMSUNG
11:24


SAMSUNG

Best (and lot of) effort:

- Display working and graphics fast enough
- Sound working
- Be able to make calls
- etc
- GPS, Camera, and other non crucial hardware may not work, or work in later releases

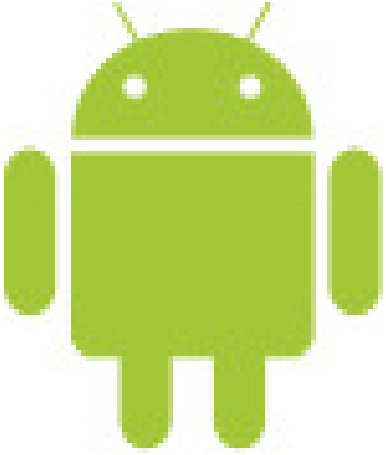
Also:

- Freedom privacy security page
- Extensive documentation on the wiki, not always easy to find

Why it is based on Android?

Android

- GUI and applications adapted to big fingers
- → run on devices that:
  - Lack keyboard
  - Have capacitve touchscreen and no stylus
  - Have very small displays with very high number of pixel



Issues with Android?

- Part of the GNU/Linux architecture is light years away: package management and build system, graphics, audio, etc
- Huge unknown code from Google
- Meant to run proprietary software, not to empower users: root, application data access

Issues with F-Droid

- F-Droid not FSDG compliant (example: Yalp Store)
- Dependency rot with Maven central vs https://f-droid.org/en/docs/Inclusion_Policy/ → Can guix help?

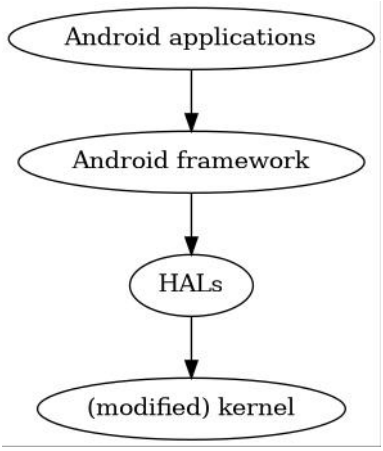
Android architecture and building Replicant

Android architecture

- Goals: time to market
  - Write the code that work as fast as possible
  - Support as many hardware features as possible including new hardware
  - → Varying kernel drivers code quality
  - → Example: One driver rewritten 3 times

How it's done?

- Breaking Kernel API and ABI
  - It can take time (years) to bring in a new framework in Linux
  - Example of API breakage: HTC dream audio driver
  - Solution: Apps ↔ Android framework ↔ HAL ↔ Kernel
  - Getting better in Android: Trebble and Generic Kernel images



Android: not made for 100% free software

- Malware often found (and removed) in App stores
- Applications are Sandboxed + permission system
- root discouraged or absent
- User access to their data extremely difficult

Most GNU/Linux distributions

- Users can easily run applications
- Users have root and access to their data
- Compiler obtimizations (-fsanitize)
- Secomp and privilege drop
- Sometimes, daemons are sandboxed

How Replicant makes a 100% free Android

- Avoid or replace non-free software
  - Replace nonfree software: Write Hardware abstraction libraries
  - Avoid nonfree software:
    - Tweak the source code not to depend on nonfree software (like 3D acceleration)
    - Remove nonfree software

	Android + vendor kernel	Android + upstream Linux	Guix
Libc	Bionic	Bionic	glibc
Kernel	Vendor kernel	Upstream kernel	Upstream kernel
Build system	No packages	No packages	Packages

Android Build system

- No packages, one big build tree:
  - Requires specific GNU/Linux distribution versions to build
  - Unclear licenses
  - No abstraction of lower level build systems like autotools, cmake, etc
  - Only Android.mk and Android.bp available, not very flexible (no ./configure –with-options)
  - Prebuilt toolchain and sometimes even linux kernel

Building requirements

Replicant version	Distributions	arch	RAM	Space
Replicant 6.0	Debian 9 (stretch)	x86_64	8G	220GiB for 10 devices
Replicant 11	Debian 10 (buster), Trisquel 9, Trisquel 10	x86_64	8G	250GiB for 2 devices

Use Guix to replace Android build system completely?

- Advantages:
  - Reproducible and bootstrapable builds
  - Works on any distribution (no need to install Trisquel)
  - Packages (better licensing)
  - Substitutes: faster builds
  - Can mix and match Android and GNU/Linux components
- Issues:
  - Guix not ready for that (Fragile Android ndk build system)
  - Cannot go back easily
  - If there is no maintenance: Replicant is dead

Use Guix inside the Android build system

- Advantages:
  - Like previous slide
  - Can do it step by step
  - Can roll back to pure Android build system
- Issues:
  - Need to add support for generating Android.mk / Android.bp in Guix
  - How to deploy (guix pack uses gnu/*)
  - Need to be careful not to depend too much on it

As host distribution to build Replicant

- Advantages:
  - Reproducible builds
  - Works in all distros
- Issues:
  - Too complicated?:
    - Broken script
    - Broken executables
- Though:
  - Bitcoin uses something like that
  - Android build is isolated from host by default
    - path interposer
    - can use namespaces isolation

Use of Guix for building Replicant

- Replicant uses repo-tool and repo-tool updates itself
- Replicant uses stable distributions that don't update python
- $\rightarrow$ We use guix pack to release a recent repo-tool that works on older distributions

Running Guix on top of Replicant?

Replicant 6

- Replicant 6 use vendor kernels based on Linux 3.0:
  - Would require to use Linux 3.0 kenrel header and to rebuild everything
  - Quick try didn't work

Replicant 11

- Recent kernel based on upstream Linux
- Guix pack works, no integration with Android at all
- Replicant doesn't want to ship Guix (too big) but ship guix-install.sh instead.
  - Can try to reuse information from https://lepiller.eu/en/guix-on-android.html
  - Work has started to modify Replicant 11 to support guix-install.sh:
    - Bash and wget implementation (wrapper over curl) now shipped in Replicant 11
    - Commands still missing: gpg (how to build it?), getent groupadd xz
    - Can ship /etc/protocols, /etc/services, create users and groups in Replicant, etc
    - Might need to send patches to install.sh to support Replicant
    - At some point, might need root, selinux integration, etc

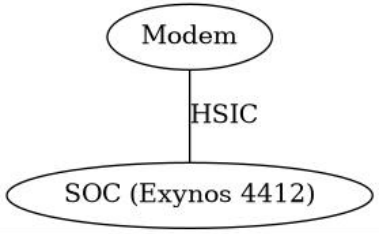
guix pack -f apk PACKAGE ?

- Requires Android NDK to be portable
- Bionic libc instead of glibc
- → New target like x86_64-w64-mingw32

Automatic Testing

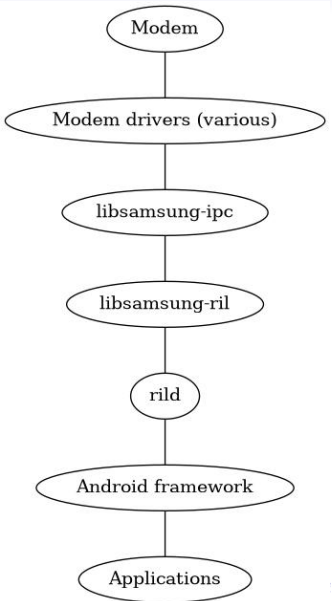
Telephony stack: Hardware

- The modem uses a custom protocol
- → Needs software to talk to it
- → The protocol was reverse engineered and reimplemented by various people



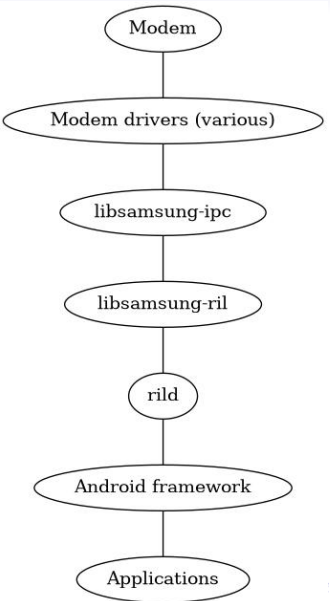
Telephony stack: Software

- libsamsung-ipc also works on GNU/Linux



Telephony stack: Software

- Use of guix.scm for both libsamsung-ipc and libsamsung-ril:
  - Can compile for various configuration (Android, GNU/Linux) and run
  - Supports static compilation and cross compilation (without transformation)
  - Supports PowerPC (big endian)



Replicant infrastructure and Guix.

Infrastructure:

- OSUOSL: Mailing list, Wordpress, website, Redmine, FTP
- La Quadrature du net: Mastodon account on mamot.fr
- 1 VM @ FSF: git, domain name, contact address, IRC bridge
- Slow move to Mediawiki on FSF VM

Guix?

- OSUOSL: Mailing list, Wordpress, website, Redmine, FTP → Not possible
- La Quadrature du net: Mastodon account on mamot.fr → Not possible
- 1 VM @ FSF: git, domain name, contact address, IRC bridge

Can we use Guix system in an VM @ FSF?

- FSF Requirement: Encrypted rootfs
  - Before: Required grub-crypt patches (In Ubuntu) + the FSF used custom script with debootstrap.
  - Now (19 July 2022): (some of the) grub-crypt patches upstreamed, encryption done outside of the VM $\rightarrow$ we can use guix system image.
  - Remaining issue: How to install Guix?
    - Convert Trisquel to Guix?
    - Provide a trusted image to the FSF?
  - Maintenance: automatic updates could work? (Rolling release but maintained config system)

- Guix advantages:
  - Anybody can contribute (everything in git)
  - Clean and standard solution (no custom made deploy scripts, etc)
  - Can fork and re-deploy, system backups inside git
  - Question: Which license to use? GPLv3 or AGPLv3?
    - Use case: Enable copyleft forks: Redirect scripts part of Replicant
- Guix disadvantages:
  - Require people to know Guix too
  - More work to package missing things
  - More work to make it work out of the box (letsencrypt, other services to add)

VM Status:

- Use Trisquel with automatic updates
- Part of the configuration in git:
  - Mail setup
  - Apache configuration
    - The Apache configuration is a key component of Replicant source code as it contains redirects necessary to build and maintain several Replicant versions.
  - Matterbridge configuration
- Uses Guix pack + systemd + config in git for matterbridge

Licenses

- Moblie phones electronic waste https://commons.wikimedia.org/wiki/File:Day_6_Warehouse_(25890985098).jpg CC-BY-SA 2.0 Generic
- anti-smartphone https://commons.wikimedia.org/wiki/File:Bia%C5%82a_Podlaska_~21ujcqtd.jpg CC-BY-SA 4.0 International
- Phone and phist https://wiki.rhizomatica.org/index.php/File:Phist.jpg CC-BY 3.0
- The leader of the Luddites https://en.wikipedia.org/wiki/File:Luddite.jpg Public domain
- 4 freedoms https://www.gnu.org/graphics/amihud-4-freedoms.html CC-BY-SA 4.0 International License or GPLv3 or later
- anti-drm protest https://www.flickr.com/photos/bcatch/164393994/in/album-72157594161735152/ CC-BY-SA 2.0
- Raised fist: https://commons.wikimedia.org/wiki/File:Raised_fist.jpg Creative Commons Attribution 3.0 Unported
- We Can Do It: https://commons.wikimedia.org/wiki/File:We_Can_Do_It!_NARA_535413_-_Restoration_2.jpg Public domain

- GT-I9300 internals: Same author and license as this presentation, probably also available on archive.org
- Replicant Logo https://git.replicant.us/replicant/vendor_replicant_artwork/plain/replicant_logo_alpha.svg?h=main&id=fc213d2ca94cef9047d1e4a71c21c4c4c87f349d CC-BY-SA 3.0 Unported
- Replicant devices https://git.replicant.us/infrastructure/www.replicant.us/tree/images/supported-devices?id=2a331698cead2677fa953c3e1ab5d78528e39ef9 CC-BY-SA 3.0 Unported
- Worker https://commons.wikimedia.org/wiki/File:African_American_worker_Richmond_Shipyards.jpg Public domain
- Android logo https://commons.wikimedia.org/wiki/File:Android_Robot_Cleaner_(2014-2019).svg CC BY-SA 4.0 International
- Googlag https://wiki.fuckoffgoogle.de/index.php?title=File:IMG_20180428_200243.jpg CC BY-SA
- F-Droid logo https://commons.wikimedia.org/wiki/File:F-Droid_Logo_4.svg CC

- Phone shop https://commons.wikimedia.org/wiki/File:HK_WC_%E7%81%A3%E4%BB%94_Wan_Chai_%E8%8E%8A%E5%A3%AB%E6%95%A6%E9%81%93_Johnston_Road_shop_window_display_smart_phones_August_2021_SS2.jpg CC-BY-SA 4.0 International