Google Summer of Code 2018

Replicant intends to apply to GSoC 2018 as part of the FSF umbrella. This page lists suggested projects for GSoC students. **This list is a draft!**

Note: Working on most of these projects requires a sufficiently powerful computer with enough disk space to build Replicant.

U-Boot and Linux mainline related tasks and ports

Select and/or port a tablet with an Allwinner SOC to mainline Linux and U-boot, and Replicant

Tablets with Allwinner SOCs are an interesting targets because they do not use signed bootloaders and the SOCs and various devices using them have good Linux and u-boot mainline support. If not much work is required for that, once the code is merged, the candidate is also required to work on the generic abstraction layer project which is also documented in this page.

The chosen tablet should have:

- A SOC that has good mainline support, see the Linux mainlining effort page on linux-sunxi for more details.
- A Free software bootloader, or the ability to easily add support for the tablet to a free software bootloader.
- The ability to power and use an USB WiFi card or chip that is compatible with the ath9k_htc driver.

It would be better if the chosen tablet doesn't use an AllWinner SOC with a PowerVR GPU, as MALI GPU have more probability to be usable with free software in the future.

Difficulty: Medium

Requirements/Prerequisites: C programming language, driver development

Expected Outcomes/Deliverables: Replicant support for a tablet using an Allwinner SOC, with free software bootloader and mainline based Linux kernel.

Possible Mentors: Paul (confirmed), GNUtoo (confirmed, backup)

Advance the Optimus Black U-Boot and Linux mainline ports

The Optimus Black from LG is an interesting device from the perspective of freedom and privacy/security. It has the ability to run a free bootloader and uses an OMAP3 SoC that is well-documented and supported in upstream U-Boot (bootloader) and Linux (kernel). Its modem is well-isolated from the rest of the device, ensuring a sane base for privacy/security. Currently, the device-specific parts of the mainline U-Boot and Linux ports are still at an early stage, where they are functional with a very limited set of supported hardware.

Advancing the Optimus Black U-Boot and Linux mainline ports would allow using the device with free, up-to-date and maintainable software and would pave the way for support in GNU/Linux systems as well as Replicant. A list of priorities in hardware support will be defined, with the objective of tackling as many as possible.

Difficulty: Medium

Requirements/Prerequisites: C programming language, driver development

Expected Outcomes/Deliverables: Improved hardware support for the Optimus Black in U-Boot and Linux

Possible Mentors: Paul (confirmed), GNUtoo (confirmed, backup)

Port the Galaxy S3 and Galaxy Note 2 to Mainline Linux

The Galaxy S3 and Galaxy Note 2 currently use a vendor fork of Linux, which poses a maintainability and security issue. Forkbomb has done some initial work on porting these devices to use mainline Linux. You can help by continuing this work. This would also enable these devices to use generic hardware abstraction layers (HAL) when abstractions layers are ready, and to do some research on whether the TrustZone operating system can be removed from such devices.

Difficulty: Medium to Hard

Requirements/Prerequisites: Knowledge of C

Expected Outcomes/Deliverables: Audio working, modem working, and Replicant or LineageOS booting with mainline Linux.

Possible Mentors: Forkbomb (confirmed), GNUtoo (confirmed)

Advance the Kindle Fire (first generation) U-Boot and Linux mainline ports

The Kindle Fire (first generation) from Amazon is an interesting device from the perspective of freedom and privacy/security. It has the ability to run a free bootloader and uses an OMAP4 SoC that is well-documented and supported in upstream U-Boot (bootloader) and Linux (kernel). It does not embed a modem, ensuring a sane base for privacy/security. Currently, the device-specific parts of the mainline U-Boot and Linux ports are still at an early stage, where they are functional with a very limited set of supported hardware.

Advancing the Kindle Fire (first generation) U-Boot and Linux mainline ports would allow using the device with free, up-to-date and maintainable software and would pave the way for support in GNU/Linux systems as well as Replicant. A list of priorities in hardware support will be defined, with the objective of tackling as many as possible.

Difficulty: Medium

Requirements/Prerequisites: C programming language, driver development

Expected Outcomes/Deliverables: Improved hardware support for the Kindle Fire (first generation) in U-Boot and Linux

Possible Mentors: Paul (confirmed), GNUtoo (confirmed, backup)

Implementing generic Hardware Abstraction Layers (HALs)

Currently, Replicant uses a dedicated Hardware Abstraction Layer per device, because device manufacturers implemented non-standard kernel interfaces. However, Android works with mainline kernels and supports plug-n-play hardware nowadays, so it makes sense to have generic Hardware Abstraction Layers for the standard interfaces of the Linux kernel (ALSA, V4L2, etc). In particular, this would allow supporting external Wi-Fi dongles such as the ones supported by the ath9k_htc driver and free firmwares without the need for a specific application or configuration.

Difficulty: Medium

Requirements/Prerequisites: C programming language, kernel interfaces knowledge

Expected Outcomes/Deliverables: A collection of generic HALs for Android and Replicant

Possible Mentors: Forkbomb (confirmed), Paul (confirmed, backup), GNUtoo (confirmed)

RIL related tasks

Implement the missing features of Samsung-RIL

Samsung-RIL is the RIL (Radio Interface Layer) that many Replicant devices use to communicate with the modem. It is a free, reverse-engineered replacement for the proprietary RIL that the Samsung phones ship with by default (which has been found to have backdoors).

Right now, Samsung-RIL mostly implements only the protocol features that are absolutely necessary for the phone to be operable. As a result, many more rarely used protocol features are unimplemented, which decreases functionality compared to the proprietary RIL. You can help by implementing the missing features of Samsung-RIL.

It would also be nice to fix most the reported bugs involving samsung-ril and libsamsung-ipc that are impacting users very seriously. This includes the bugs about the SIM card not being detected, and the issue about having metallic sound quality when doing voice calls over 3G (bug #1773). It would also be nice to be able to recover from EFS (the modem filesystem) corruptions (Bug #1869).

Difficulty: Medium to Hard

Requirements/Prerequisites: Knowledge of C.

Expected Outcomes/Deliverables: Implement the missing features listed at Samsung-RIL.

Possible Mentors: Forkbomb (confirmed), Wolfgang?

Implement a fully-featured QMI-RIL

The LTE variants of the Samsung Galaxy S3 and Samsung Galaxy Note 2 use a different modem from the non-LTE variants that Replicant currently supports. You can help Replicant support those modems by implementing a QMI-RIL, which performs a similar role on the LTE variants as what Samsung-RIL performs on the currently-supported non-LTE variants. Wolfgang has done some preliminary work on this, so you'll probably be picking up where he left off.

Difficulty: Hard

Requirements/**Prerequisites**: Knowledge of C.

Expected Outcomes/**Deliverables**: A QMI-RIL that supports voice calls, SMS, and data, with as complete a protocol implementation as possible.

Possible Mentors: Forkbomb (confirmed), Wolfgang?

Access Point mode for RepWifi

RepWifi is Replicant's app for using an external USB WiFi adapter. RepWifi is useful for Replicant because there exist USB WiFi adapters with free firmware, while the built-in WiFi chipsets in mobile phones do not have free firmware. Right now, RepWifi doesn't support acting as an access point (e.g. for WiFi tethering purposes); you can help by adding this functionality to RepWifi.

Note that, if we receive high-quality student proposals for both this project and **Implementing generic Hardware Abstraction Layers (HALs)**, we will probably prioritize **Implementing generic Hardware Abstraction Layers (HALs)** over this project.

Difficulty: Easy to Medium

Requirements/**Prerequisites**: Knowledge of Java and basic shell scripting. Basic knowledge about wpa_supplicant and general network management in POSIX environments.
Required knowledge builds up very fast by trial and error, no need to be experts in networking, it's mostly about researching and learning.

Expected Outcomes/**Deliverables**: Make wpa_supplicant run in "Access Point mode", allowing another device to connect to the phone via WiFi, and use its mobile data connection to access the internet. Integrate the needed GUI functions into RepWifi.

Possible Mentors: Fil (confirmed)

Port Replicant to a newer LineageOS version and support in-system updates

Replicant is currently based on LineageOS 13. It would be desirable to upgrade Replicant to a newer release of LineageOS. While at it, it would be useful for a Replicant device to be able to update itself to a new version of Replicant without requiring being connected to a PC. LineageOS already supports this; we suspect that it should be possible to adapt this LineageOS functionality to Replicant. Whenever possible, it would be useful to complete and submit some of the code written for Replicant to LineageOS.

Difficulty: Medium

Requirements/**Prerequisites**: Knowledge of C, C++, and Java.

Expected Outcomes/**Deliverables**: Remove all proprietary components of LineageOS, port all the changes needed to successfully boot without any blobs, rebrand LineageOS as Replicant and support in-system updates

Possible Mentors: Forkbomb (confirmed), Wolfgang?

Improve support for the free software compatible external WiFi adapter

All devices currently supported by Replicant have WiFi chips that requires a non-free firmware to work. So to have WiFi working with free software, users need to use external WiFi adapters. They typically use tiny ath9k_htc compatible USB WiFi adapter along with a tiny USB OTG Host adapter.

Such external USB WiFi adapters used with Replicant are originally intended for laptops, not phones. As a result, they tend to consume a lot of power. According to lsusb some ath9k_htc compatible devices can consume up to 500mA.

This poses several issues:

- Some smartphones and tablets might not be compatible, at the hardware level, with such big power consumption.
- They can adversely impact battery life

Such USB WiFi adapters can also randomly stop working completely on some devices (e.g. needing to unplug and replug the adapter periodically to keep it operational).

You will need to investigate reliability issues such as the one mentioned above and look how power consumption can be improved in the adapter firmware and/or kernel driver.

You will also need to investigate how much miliampers USB devices can use, at the hardware level, on the smartphones and tablets Replicant supports.

Difficulty: Medium/Hard

Requirements/Prerequisites: Knowledge of C

Expected Outcomes/Deliverables: Reliable WiFi with external WiFi adapter

Possible Mentors: GNUtoo (confirmed)

Tackle security issues in Replicant

Replicant is plagued by various security issues, that are mostly due to using a downstream codebase. One of the most crucial issues is that Replicant uses an old version of the Android WebView (from circa 2015), which is also a functionality drawback.
An initial evaluation of the security issues in Replicant should be conducted, followed by the integration or update of the concerned components of the system.

It would also be nice to do the same for privacy issues. Since Replicant indirectly depends on the "Android Open Source Project" and directly depends on LineageOS, not all privacy issues might have been found fixed by Replicant. Once security issues have been fixed, it would be nice to try to identify as many privacy issues as possible, and in a second time to fix them.

Difficulty: Medium-Hard

Requirements/Prerequisites: Android build system, knowledge of system security, advanced git

Expected Outcomes/Deliverables: Integration or update of components of Replicant to tackle security issues

Possible Mentors: Wolfgang?

Fix the Free software distribution guidelines issues and improve the build system.

Replicant has some issues with FSDG compliance: F-droid repository is not FSDG compliant anymore (Bug #1629), and Replicant can't be built from an FSDG distribution (Bug #1861). This ought to be fixed. Replicant should also be fixed to build without issue.

It would also be nice to have the build system not depend on pre-built dependencies anymore, and to document which FSDG compliant F-droid applications crash because Replicant's incomplete EGL implementation (#705) and tag such applications as incompatible (so they are greyed out) until the EGL implemetation is fixed. Ideally Replicant builds should also be made reproducible if they are not already.

Difficulty: Easy

Requirements/Prerequisites: Knowledge of shell scripts and the ability to learn the Android build system

Expected Outcomes/Deliverables: The ability to compile Replicant from an FSDG distribution, F-droid only showing FSDG compliant software.

Possible Mentors: GNUtoo (confirmed)

Projects mentored by other organizations

llvmpipe ARM optimizations

Project description: Replicant's EGL is implementation is incomplete (This has many consequences: #705). The goal is to make llvmpipe usable under Replicant and replace the incomplete EGL implemetation with that. The advantage of this solution over other solutions is that it is supposed to work on all Replicant supported devices and is also used within GNU/Linux.
See also the wiki page about llvmpipe for more details about the issue. Other solutions fixing the problem might be acceptable too, if it makes more sense.

Difficulty: See with the MESA project

Requirements/**Prerequisites**: See with the MESA project

Expected Outcomes/**Deliverables**: Working EGL implementation, fast enough graphics, F-droid applications not crashing anymore because of EGL.

Possible Mentors: Mesa would probably be a good organization for mentoring this project. If interested in working on this project, please propose it to Mesa. (Replicant contributors would also be happy to help.)