



Open source UEFI and TianoCore

Leif Lindholm, NUVIA (and TianoCore)

Agenda

- UEFI “Code First” process
- Open-source UEFI SCT
- Random bits

UEFI “code first” process

Code First?

My guess is everyone in this room knows “code first”, just not that it needed its own name.

What is meant by “code first” in this context is perhaps most easily explained by explaining how the UEFI Forum has worked on specification changes/additions up to this point.

Code Last? (1)

The UEFI Forum was set up as a collaborative environment between on the one side a bunch of competing hardware manufacturers (AMD/Intel/ARM as well as system and card vendors), and on the other side a bunch of competing firmware houses (“BIOS vendors”).

Peace was maintained in this wild frontier town by the UEFI Forum bylaws. Which as in many industry standards organisations (for reasons well known) focus strongly on protections against submarine patents.

Code Last? (2)

The ultimate guarantee of this is the process through which specification changes happen - the “Engineering Change Request” or ECR.

ECRs are bandied about in NDA-covered meetings. As part of the specification publication process members are given a deadline of speaking up by or the bylaws say they have given up their right to pursue infringement claims (against implementations of the specification). Before publication, code implementing new features must not be published.

Only after spec release has happened will anyone’s code be protected by the bylaws.

So, Code First?

So what is this “Code First” thing all about then?

Well, it is a proposal for how to organise prototyping new features in public without violating the bylaws.

Proposal sent to edk2-rfc and edk2-devel mailing lists

<https://edk2.groups.io/g/devel/message/53420>

Bottom line is - this development will be tracked in Tianocore Bugzilla (<https://bugzilla.tianocore.org/>) with code (and documentation) held under the Tianocore github area (<https://github.com/tianocore/>)

Open Source UEFI SCT

UEFI SCT Open Source

There are many test suites available to help verify the correctness of your system firmware, like:

- FWTS
- ARM Server-ACS
- CHIPSEC

The UEFI Self-Certification Testsuite is a very basic one, but it verifies fundamental API conformance for UEFI implementations

Available from <https://github.com/tianocore/edk2-test/tree/master/uefi-sct> since 2018-10.

SCT is for all UEFI flavours

With the addition of UEFI interfaces to U-Boot, it is really useful to be able to prove interface consistency between implementations - and it's really valuable to be able to test the test suite against different implementations.

As part of his work with U-Boot, Heinrich has already resolved issues in the UEFI Shell and in SCT itself.

Random Bits

Support for 64-bit RISC-V going upstream!

Proof-of-concept EDK2 port initially submitted in 2016? is currently undergoing rework by Abner Chang at HPE and being upstreamed.

- Track current status on RISC-V-V2 port in <https://github.com/tianocore/edk2-staging/>.
- Goal to include in EDK2 Q2 2020 stable tag.

EDK2 License

EDK2 has been using the TianoCore Contribution Agreement, operating in conjunction with a 2-clause BSD license.

As of 9th of April 2019, EDK2 is now licensed using

```
SPDX-License-Identifier: BSD-2-Clause-Patent
```

without any separate contribution agreement.

Explicit license text has been removed from all source files.

edk2-platforms followed suit on 17th May 2019.

Misc

BaseTools (TianoCore build system) now supports Python 3 and Python 2.

Maintainers.txt format updated to something very similar to QEMU/Linux with paths and wildcard support.

New developer/maintainer helper scripts have been added to edk2/Basetools/Scripts/:

- GetMaintainer.py
- SetupGit.py