



# UEFI & EDK II Training

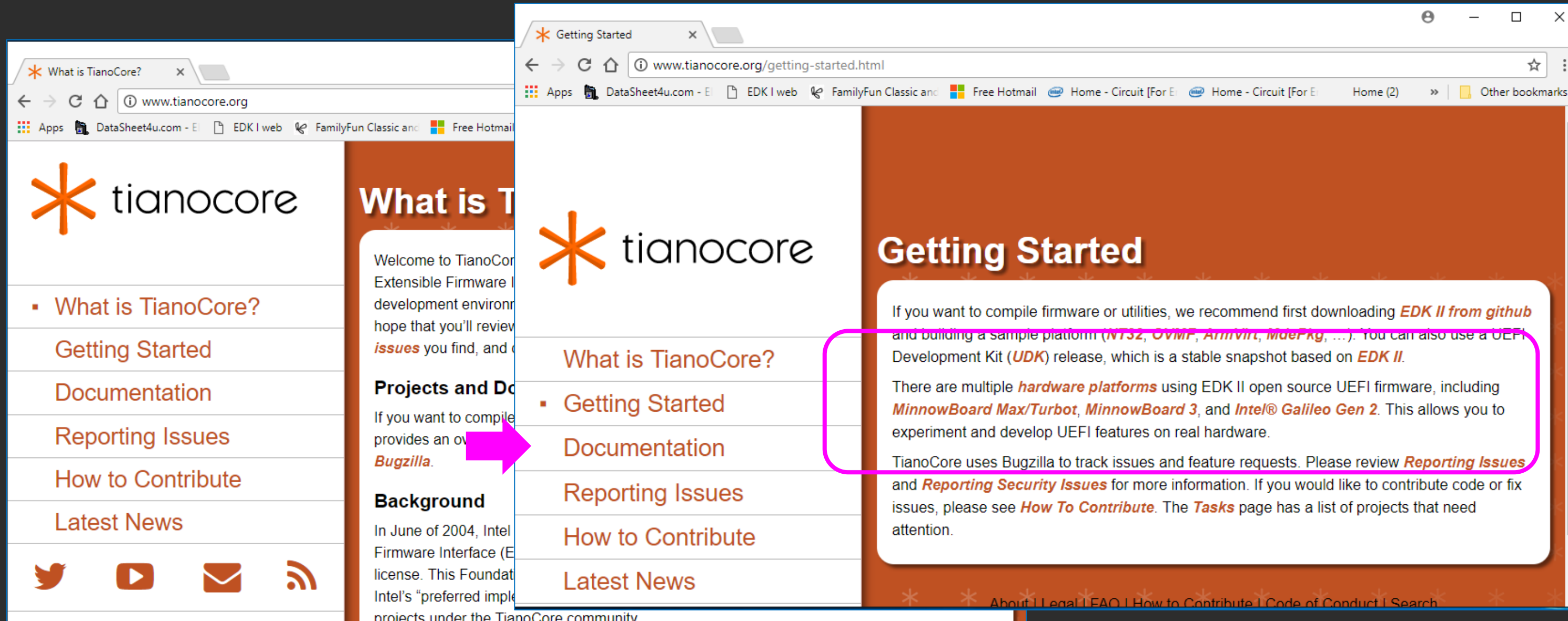
Open Source UEFI Platforms

[tianocore.org](https://tianocore.org)



# LESSON OBJECTIVE

-  Chart the organization of the Tianocore.org repositories
-  Recognize the various Open Source UEFI Platforms



**What is TianoCore?**

Welcome to TianoCore Extensible Firmware Interface (EFI) development environment. We hope that you'll review [issues](#) you find, and contribute.

**Projects and Documentation**

If you want to compile firmware or utilities, we recommend first downloading [EDK II from github](#) and building a sample platform ([NTS2](#), [OVMF](#), [ArmVirt](#), [MdePkg](#), ...). You can also use a UEFI Development Kit ([UDK](#)) release, which is a stable snapshot based on [EDK II](#).

There are multiple [hardware platforms](#) using EDK II open source UEFI firmware, including [MinnowBoard Max/Turbot](#), [MinnowBoard 3](#), and [Intel® Galileo Gen 2](#). This allows you to experiment and develop UEFI features on real hardware.

TianoCore uses Bugzilla to track issues and feature requests. Please review [Reporting Issues](#) and [Reporting Security Issues](#) for more information. If you would like to contribute code or fix issues, please see [How To Contribute](#). The [Tasks](#) page has a list of projects that need attention.

Platforms [Emulator](#), [OVMF](#), [ArmVirt](#), [MdePkgHardware platforms](#): [MinnowBoard Max/Turbot](#), [Up Squared](#), and [Intel® Galileo Gen 2](#).

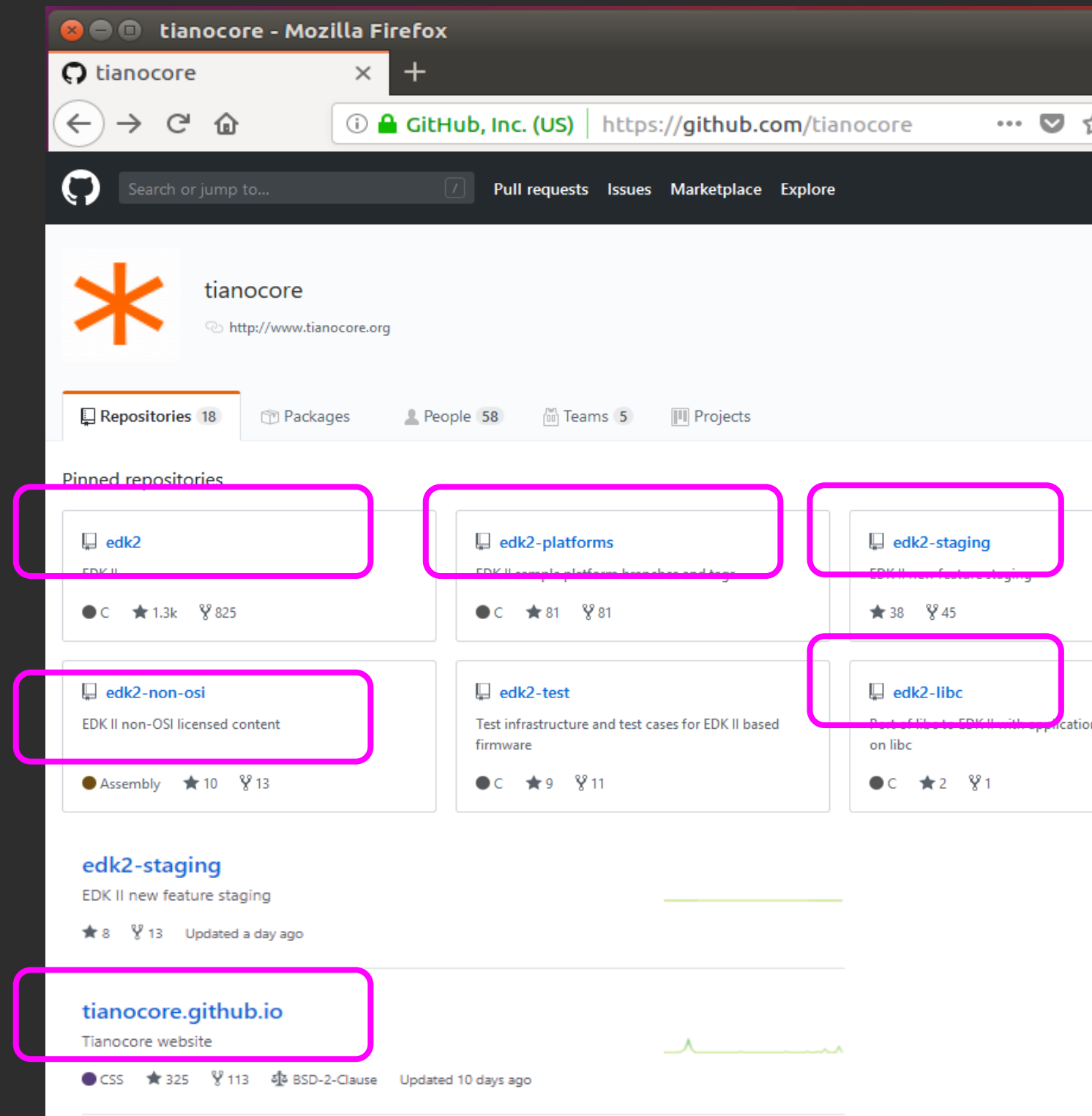


# GitHub

[Github/tianocore](https://github.com/tianocore)

## Concept of Repositories

- Main development - **edk2**
  - Other platforms - **edk2-platforms**
  - Not compatible w/ edk2 & edk2-platforms licensing - **edk2-non-osi**
  - C Library- Python **edk2-libc**
  - Work in Progress - **edk2-staging**
  - Online Info & Help (Wiki pages) **[tianocore.github.io](https://tianocore.github.io)**
- To download use “**git clone**” then “**git checkout**”

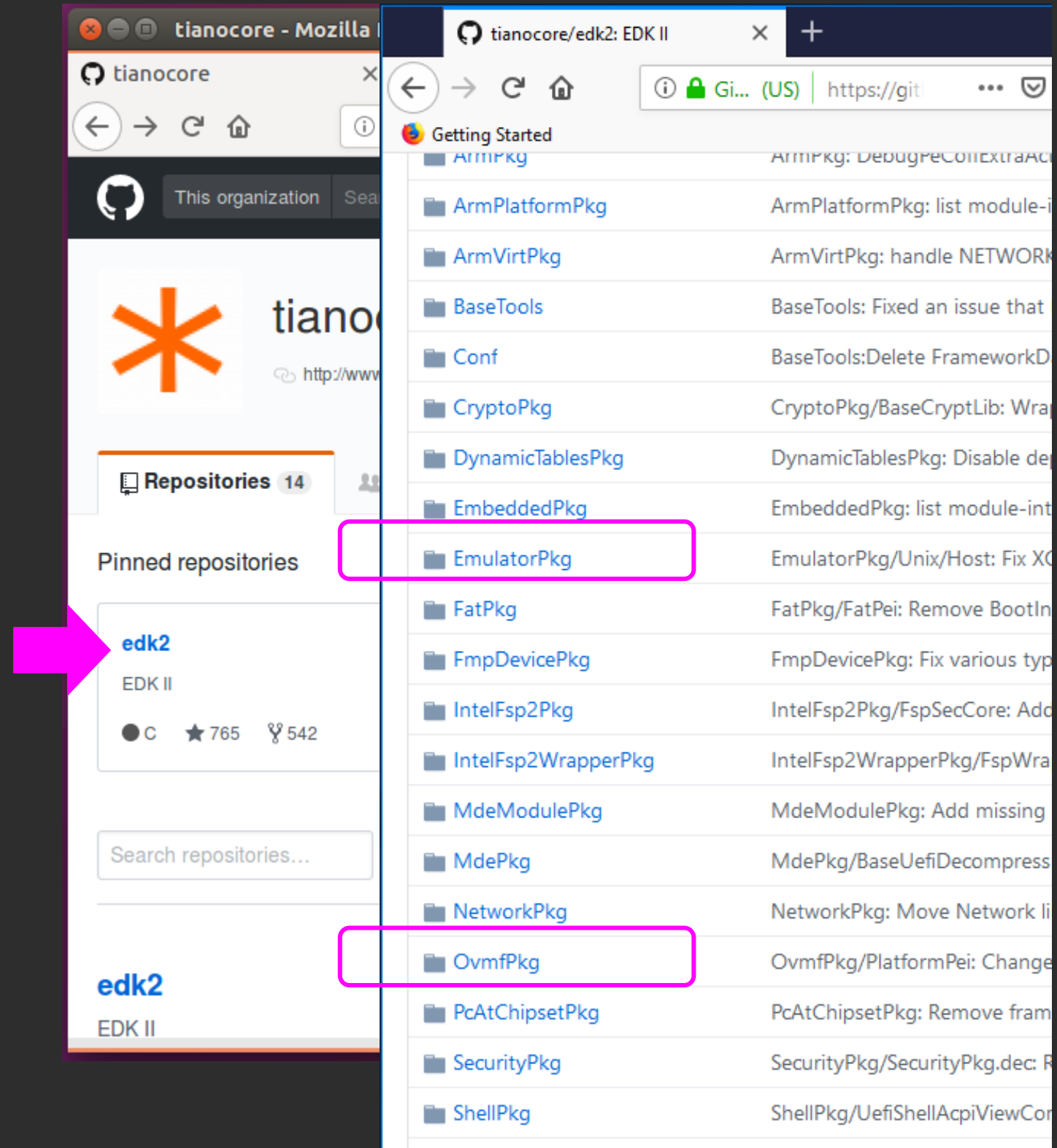


edk2 – Platforms on edk2- “CORE”

EmulatorPkg

OvmfPkg

See *Readme.md* files

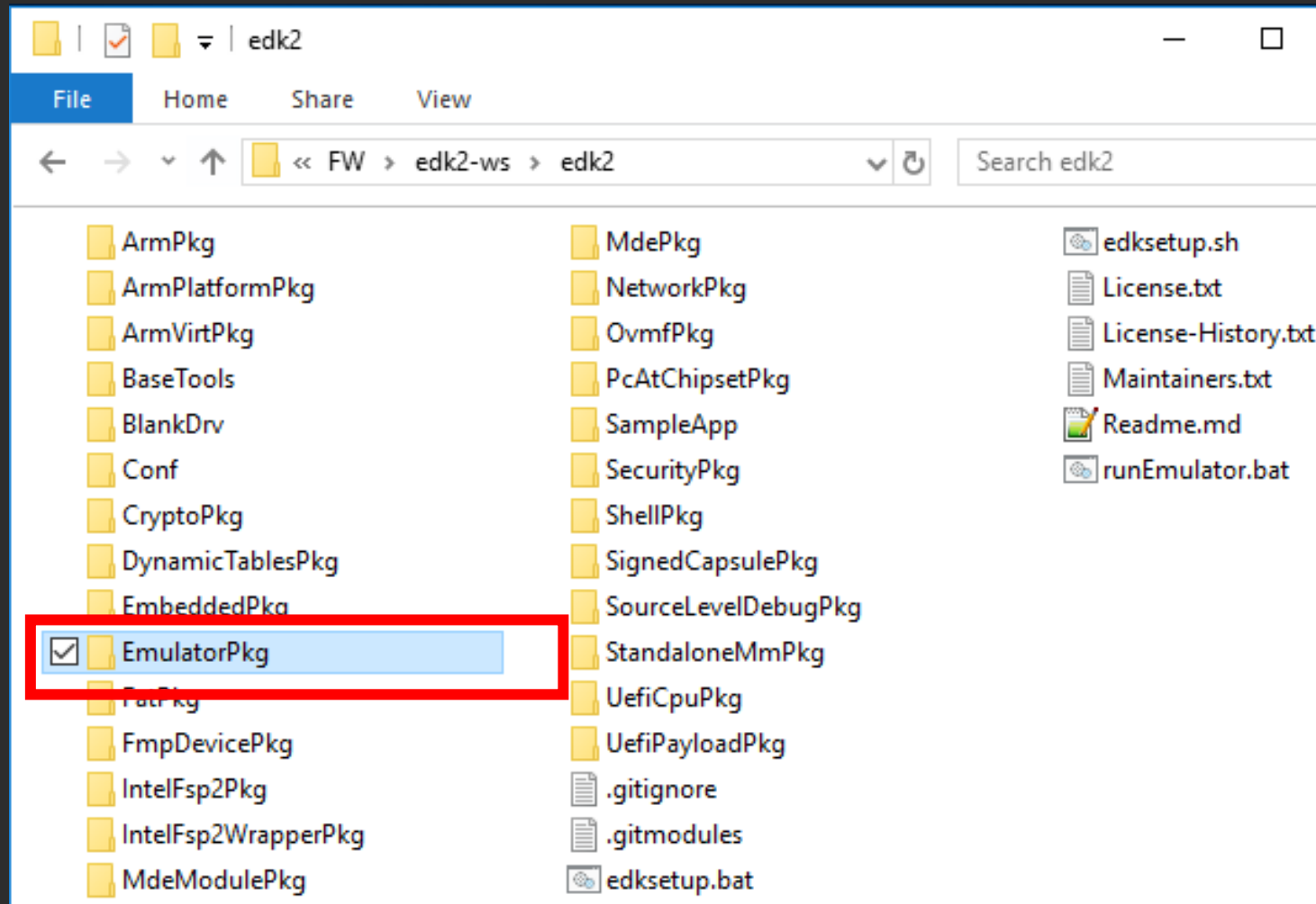




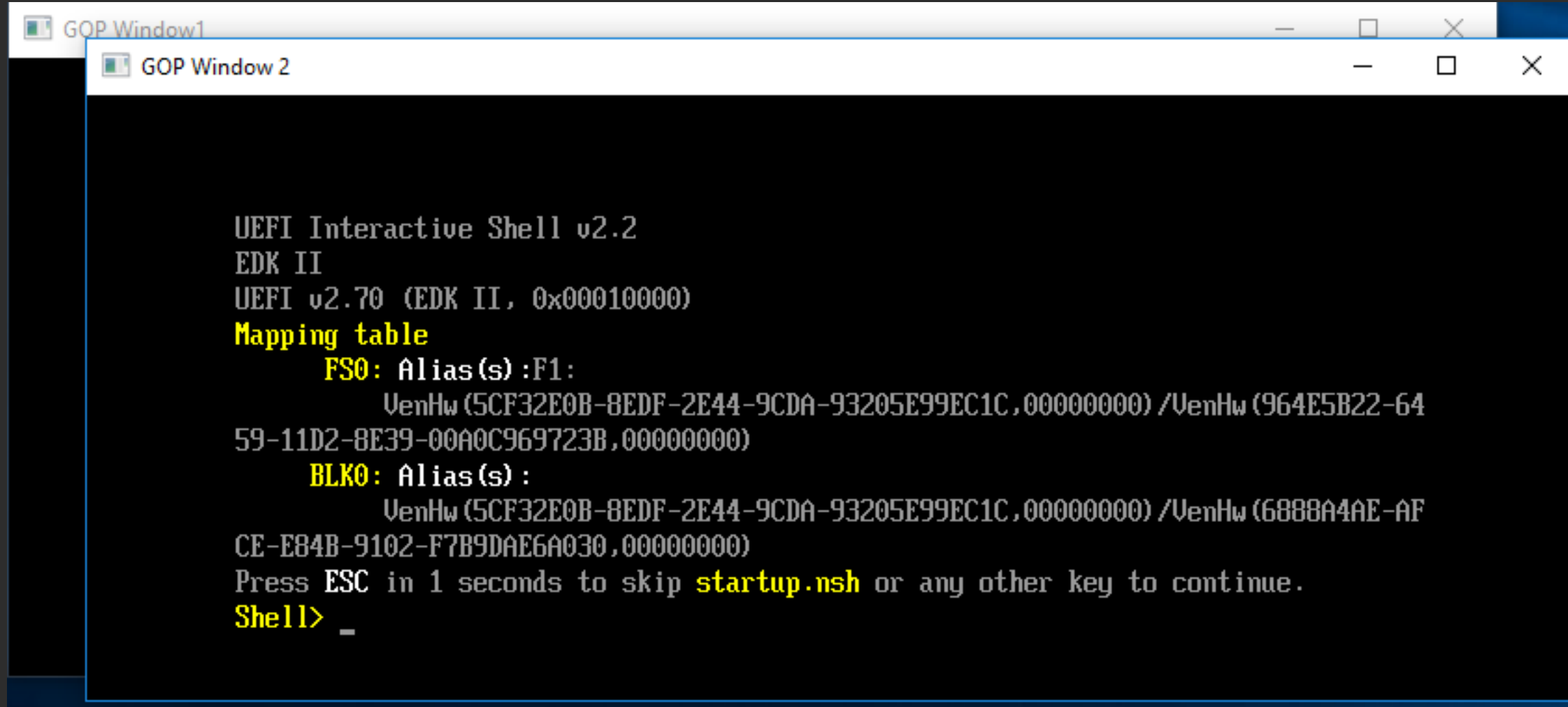
# Emulation Directory Structure

## EmulatorPkg files

- ✓ EmulatorPkg.dsc
- ✓ EmulatorPkg.dec
- ✓ EmulatorPkg.fdf



# Running Emulator with Windows



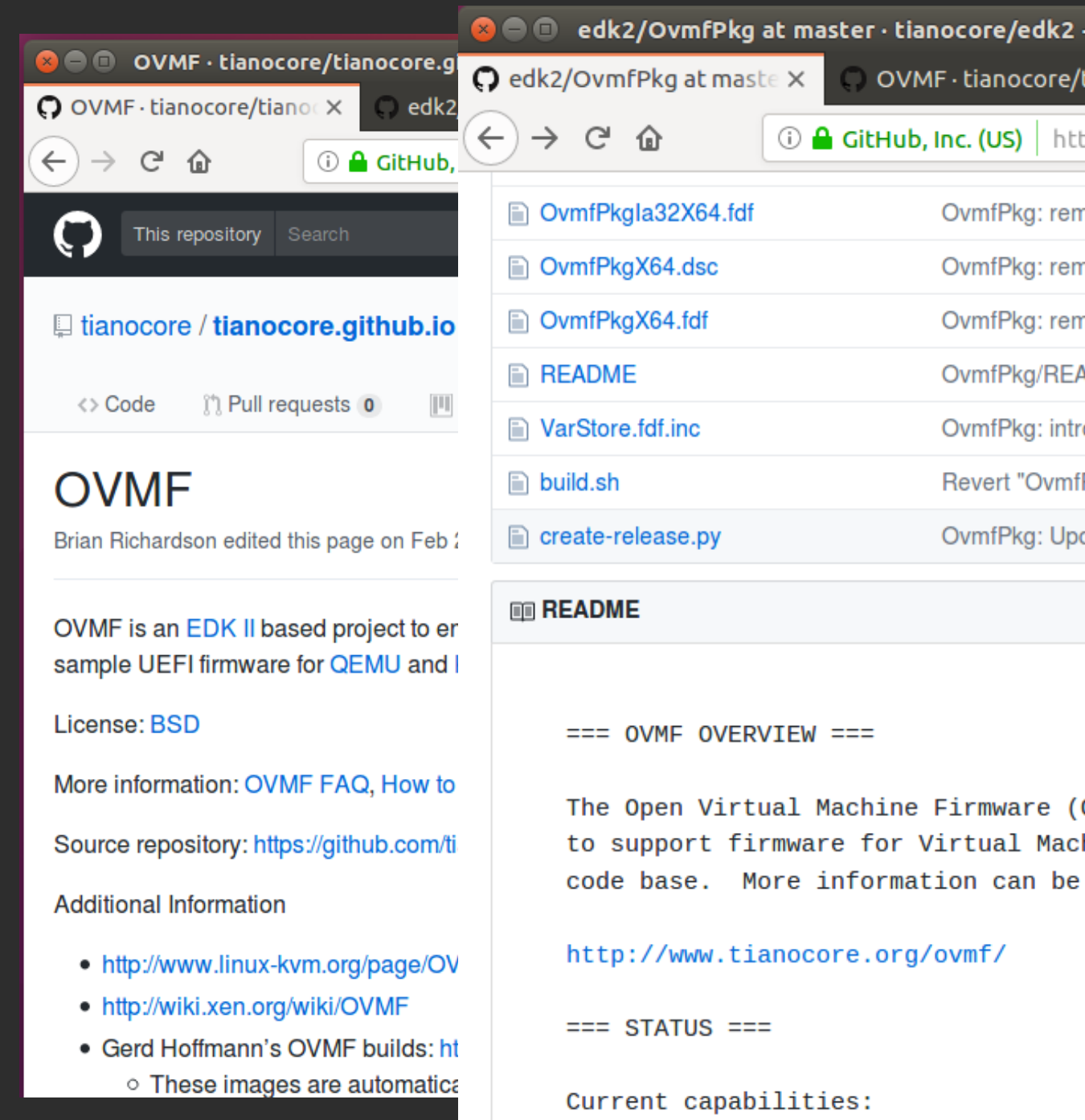
```

GOP Window1
GOP Window 2

UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s):F1:
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000) /VenHw (964E5B22-64
59-11D2-8E39-00A0C969723B,00000000)
  BLK0: Alias(s):
        VenHw (5CF32E0B-8EDF-2E44-9CDA-93205E99EC1C,00000000) /VenHw (6888A4AE-AF
CE-E84B-9102-F7B9DAE6A030,00000000)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> _
```

# Open Virtual Machine Firmware (OVMF)

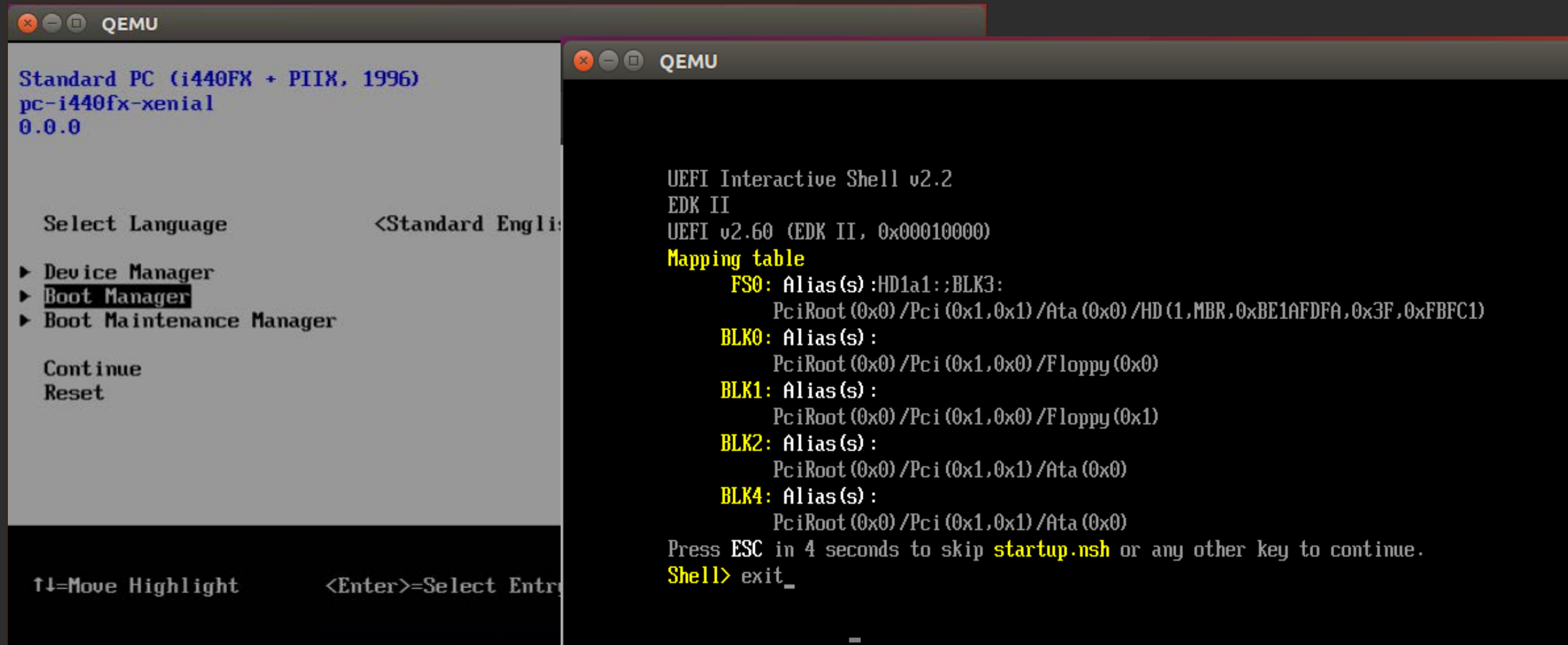
- Uses EDK II to support firmware in the OvmfPkg platform package
- Supports UEFI: Helps develop/debug drivers & applications
- QEMU VM; emulates IA32 (x86)/X64 (x86-64) based system
- Exit condition → UEFI Shell
- Tool Chain/OS Support
- Information [Ovmf wiki](https://www.tianocore.org/ovmf/wiki/), Tianocore.org





# OVMF BIOS w/ QEMU

## Boots to UEFI Shell



The image shows two overlapping QEMU window screenshots. The left window displays the BIOS boot menu for a 'Standard PC (i440FX + PIIX, 1996)' with 'pc-i440fx-xenial 0.0.0' as the version. The menu includes 'Select Language' (set to '<Standard English>'), 'Device Manager', 'Boot Manager' (highlighted), and 'Boot Maintenance Manager'. At the bottom, it shows 'Continue' and 'Reset' options, and navigation instructions: '↑↓=Move Highlight' and '<Enter>=Select Entry'. The right window shows the 'UEFI Interactive Shell v2.2' with 'EDK II' and 'UEFI v2.60 (EDK II, 0x00010000)'. It displays a 'Mapping table' with entries for 'FS0' (HD1a1::BLK3), 'BLK0' (Floppy 0x0), 'BLK1' (Floppy 0x1), 'BLK2' (Ata 0x0), and 'BLK4' (Ata 0x0). It prompts the user to 'Press ESC in 4 seconds to skip startup.nsh or any other key to continue.' and shows the 'Shell> exit\_' command.

```
Standard PC (i440FX + PIIX, 1996)
pc-i440fx-xenial
0.0.0

Select Language          <Standard English>

▶ Device Manager
▶ Boot Manager
▶ Boot Maintenance Manager

Continue
Reset

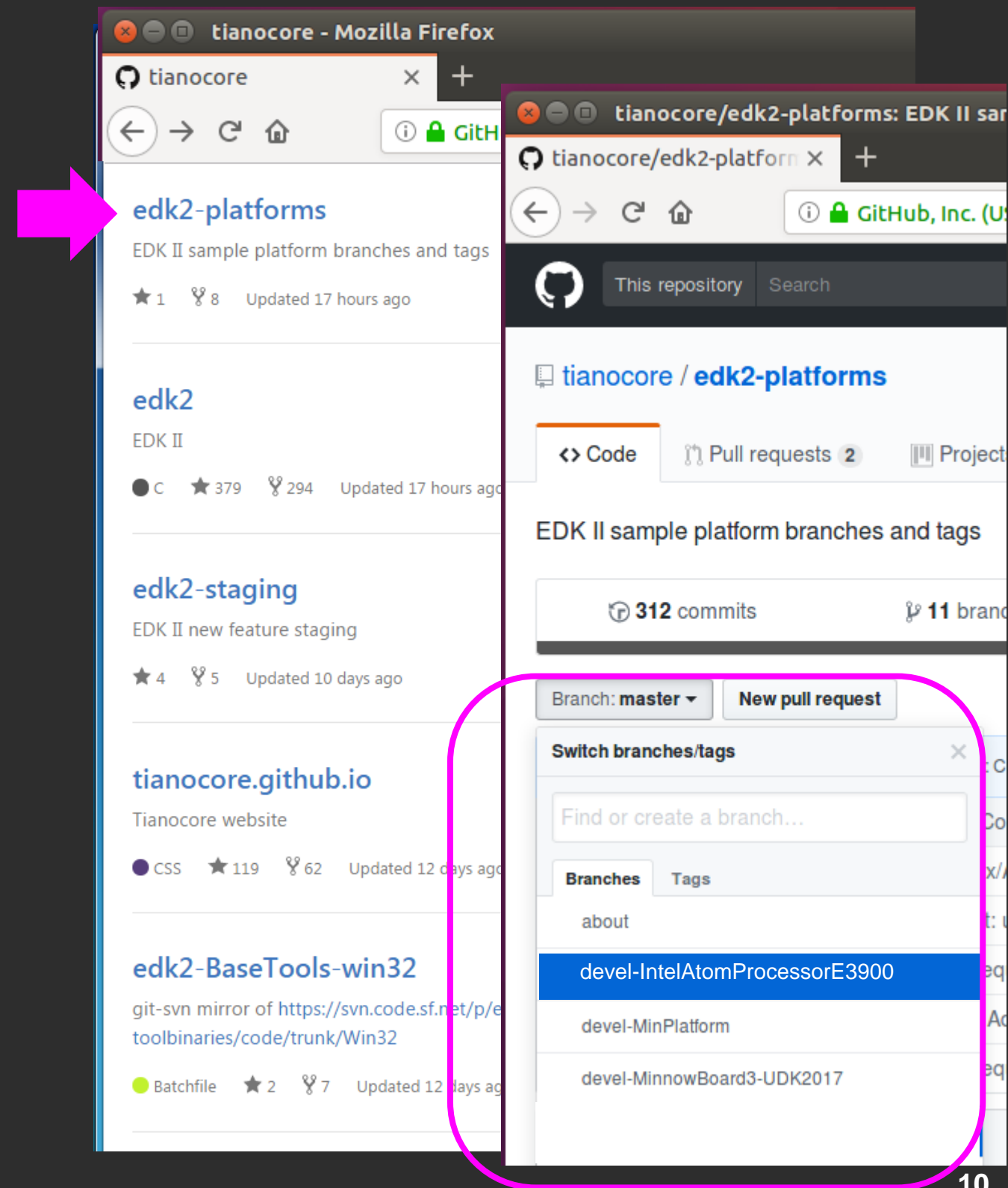
↑↓=Move Highlight      <Enter>=Select Entry

UEFI Interactive Shell v2.2
EDK II
UEFI v2.60 (EDK II, 0x00010000)
Mapping table
  FS0: Alias(s) :HD1a1::BLK3:
          PciRoot (0x0) /Pci (0x1,0x1) /Ata (0x0) /HD (1,MBR,0xBE1AFDFA,0x3F,0xFBFC1)
  BLK0: Alias(s) :
          PciRoot (0x0) /Pci (0x1,0x0) /Floppy (0x0)
  BLK1: Alias(s) :
          PciRoot (0x0) /Pci (0x1,0x0) /Floppy (0x1)
  BLK2: Alias(s) :
          PciRoot (0x0) /Pci (0x1,0x1) /Ata (0x0)
  BLK4: Alias(s) :
          PciRoot (0x0) /Pci (0x1,0x1) /Ata (0x0)
Press ESC in 4 seconds to skip startup.nsh or any other key to continue.
Shell> exit_
```

# Platforms Tianocore.org

## edk2-platforms – Platforms

- devel-IntelAtomProcessorE3900  
– Leaf Hill, Up Squared (Apollo Lake)
- Vlv2TbltDevicePkg  
– BayTrail-I
- MinPlatformPkg – (w/ FSP )
  - KabylakeOpenBoardPkg
  - TigerlakeOpenBoardPkg
  - WhiskeyLakeOpenBoardPkg
  - WhitleyOpenBoardPkg
- How to build  
See *Readme.md* files



# Slim BootLoader (SBL) Project



Fast & Secure Open source boot solution  
for IoT Use Cases

Github: <https://github.com/slimbootloader>

Supported Hardware:

QEMU

UP2 Board

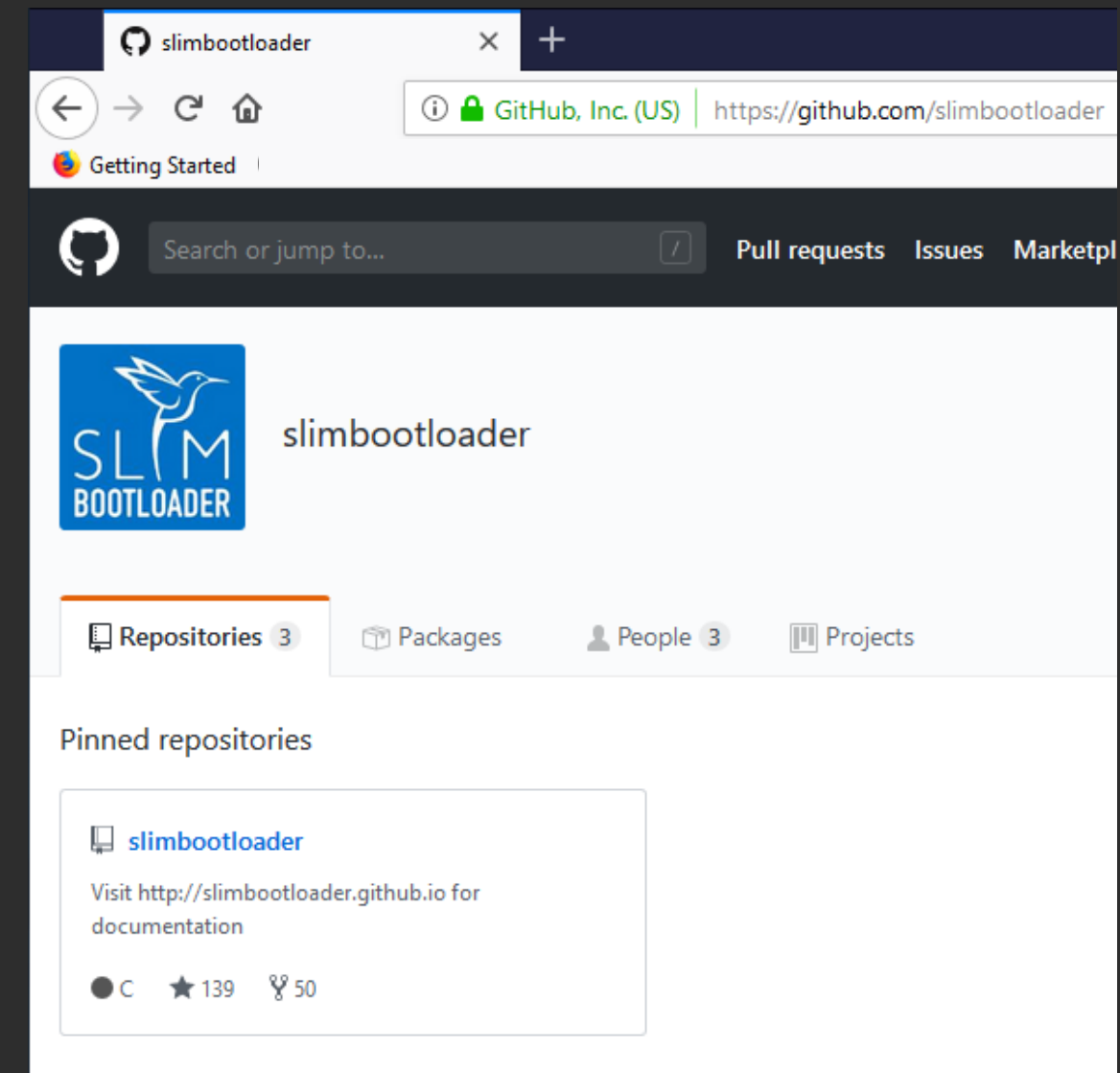
Apollo Lake CRB

Whisky Lake CRB

Coffee Lake Refresh CRB

UP Xtreme Board

Documentation: [Slim Bootloader Project](#)



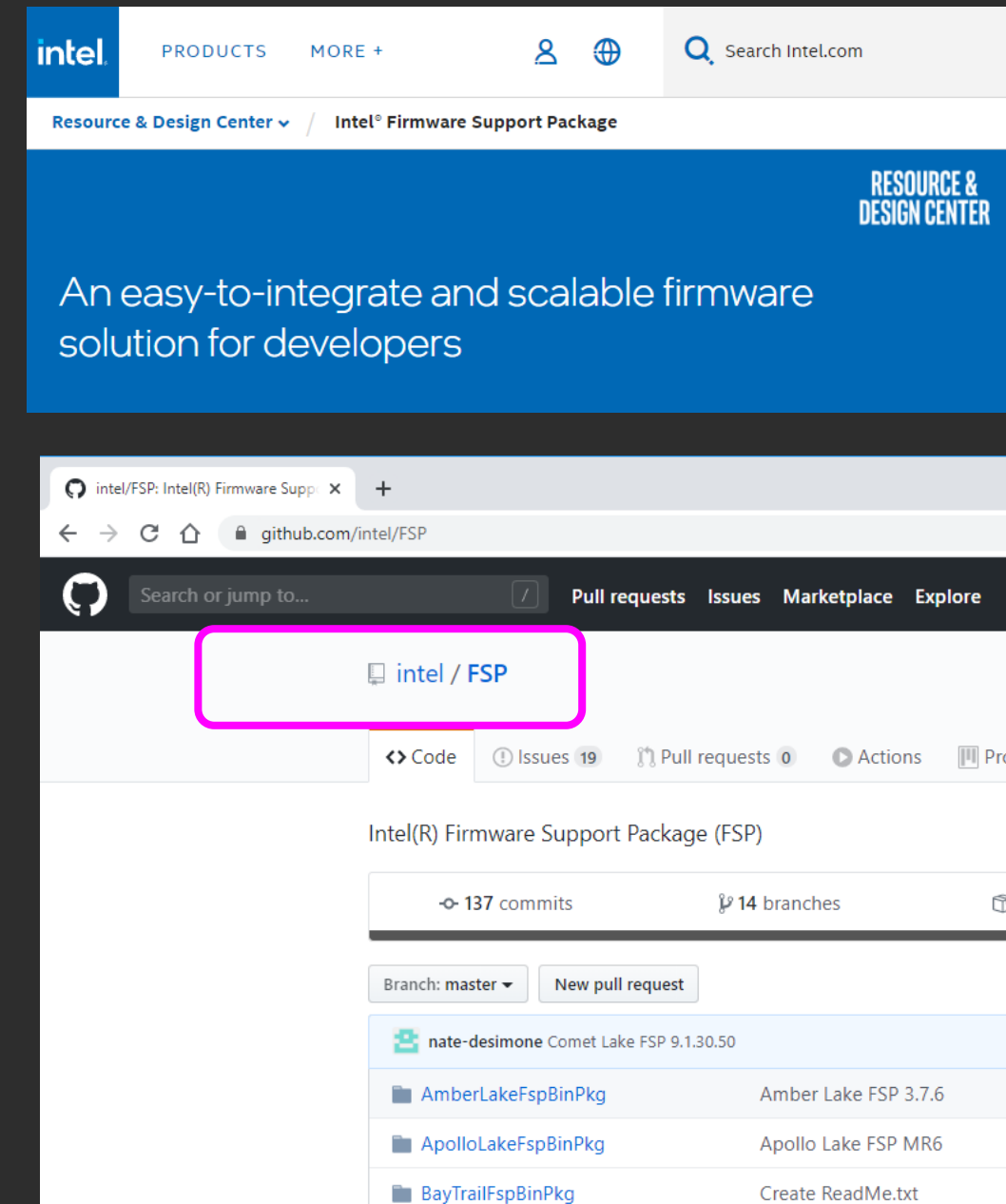
## intel Intel Developer Zone Overview

Repository of Intel FSP binaries posted by Intel on github:

Includes documentation on how to integrate with various platforms: <https://github.com/intel/FSP>

Wiki: <https://github.com/intel/FSP/wiki>

- current specifications

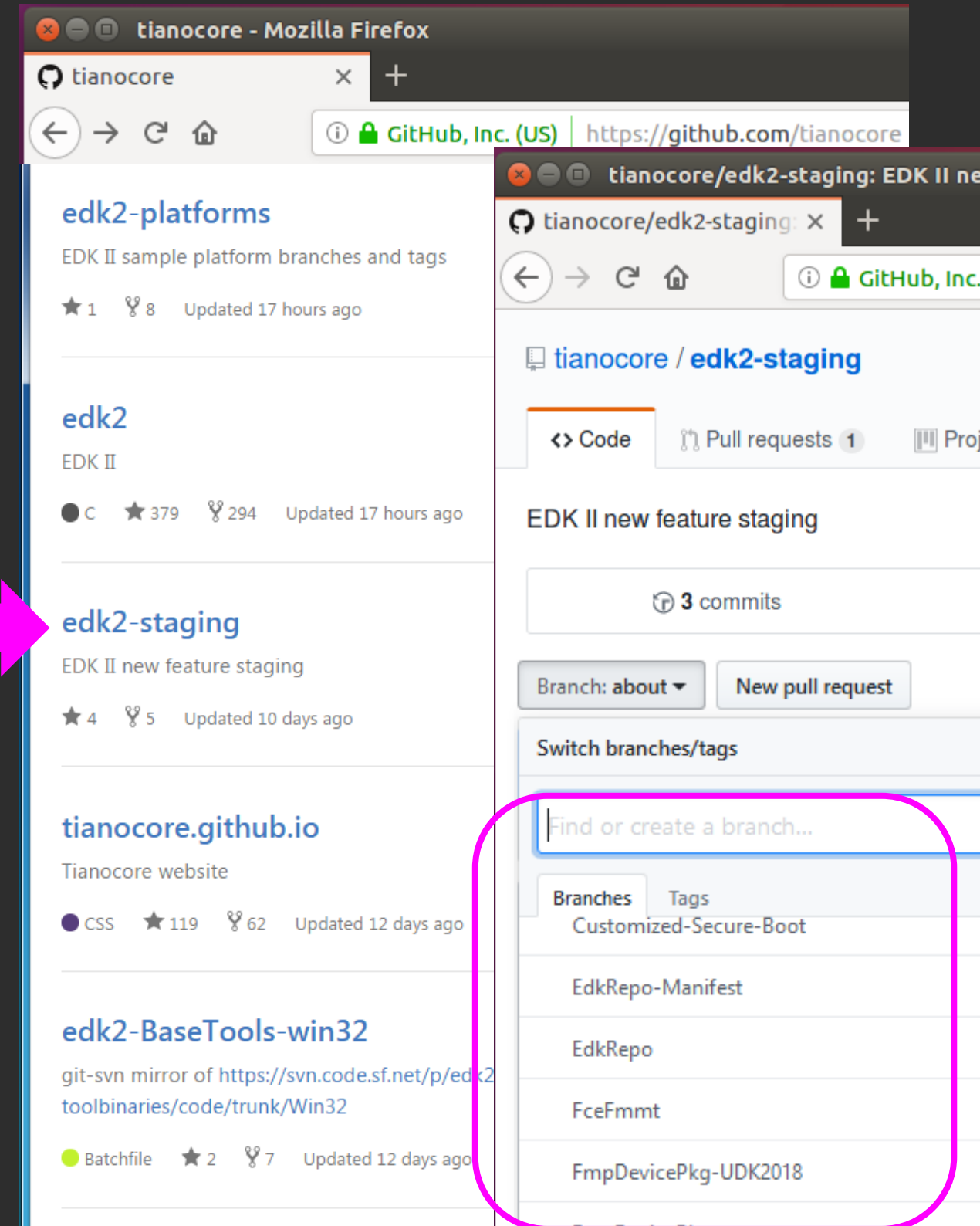


# Staging TIANOCORE.ORG

Implementations not yet Ready for EDK II Main  
[edk2-staging](#)

## Projects on branches



- Host-based FW analysis (HBFA)
- edk2-host-test
- FceFmmt (FW Utils)
- UEFI\_PCI\_ENHANCE-2
- EdkRepo
- Cpu/6-level
- HTTPS-TLS
- RICS-V
- ...
- See *Readme.md* files



The image shows two browser windows. The left window displays the GitHub repository list for 'tianocore', with 'edk2-staging' highlighted. The right window shows the 'tianocore/edk2-staging' repository page, with the 'Switch branches/tags' section circled in pink. This section includes a search bar 'Find or create a branch...' and a list of branches and tags.

Branches	Tags
Customized-Secure-Boot	
EdkRepo-Manifest	
EdkRepo	
FceFmmt	
FmpDevicePkg-UDK2018	

# Summary

-  Chart the organization of the Tianocore.org repositories
-  Recognize the various Open Source UEFI Platforms



# Questions?



# Return to Main Training Page



Return to Training Table of contents for next presentation [link](#)





# ACKNOWLEDGEMENTS

Redistribution and use in source (original document form) and 'compiled' forms (converted to PDF, epub, HTML and other formats) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (original document form) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, epub, HTML and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

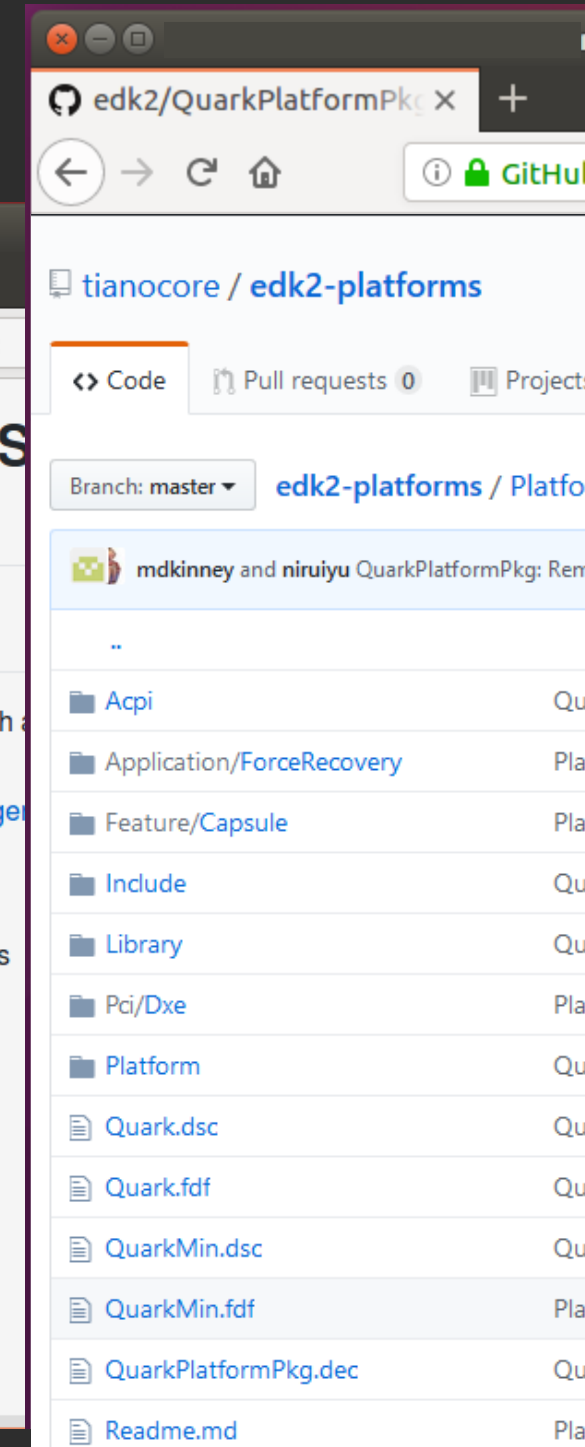
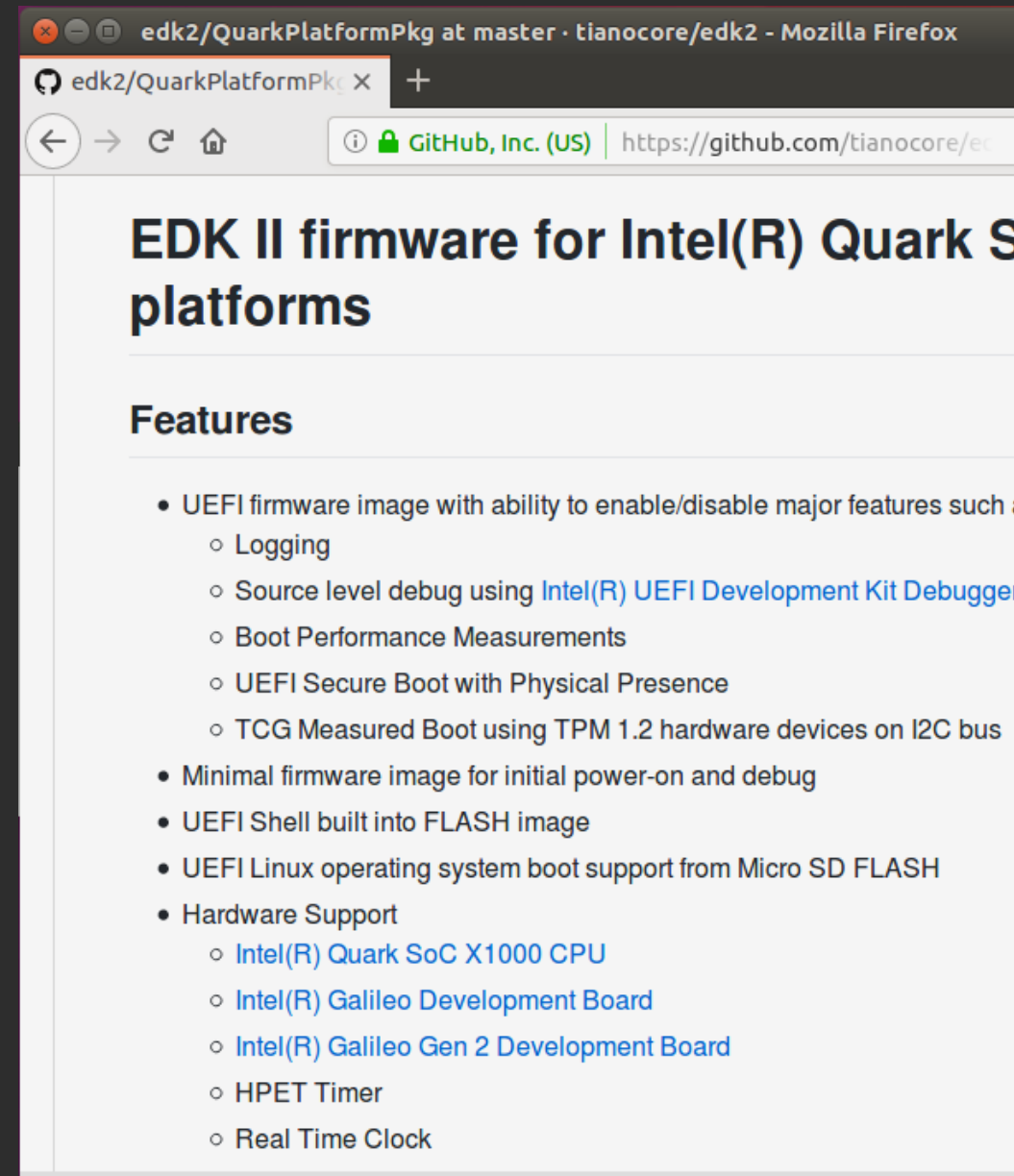
THIS DOCUMENTATION IS PROVIDED BY TIANOCORE PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TIANOCORE PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2021, Intel Corporation. All rights reserved.

**BACK UP**

# Intel® Quark SoC X1000 Platform Project EDK II

- Uses EDK II to support firmware
- QuarkPlatformPkg  
-Intel® Galileo Gen2
- How to Build: [Quark Readme.md](#)



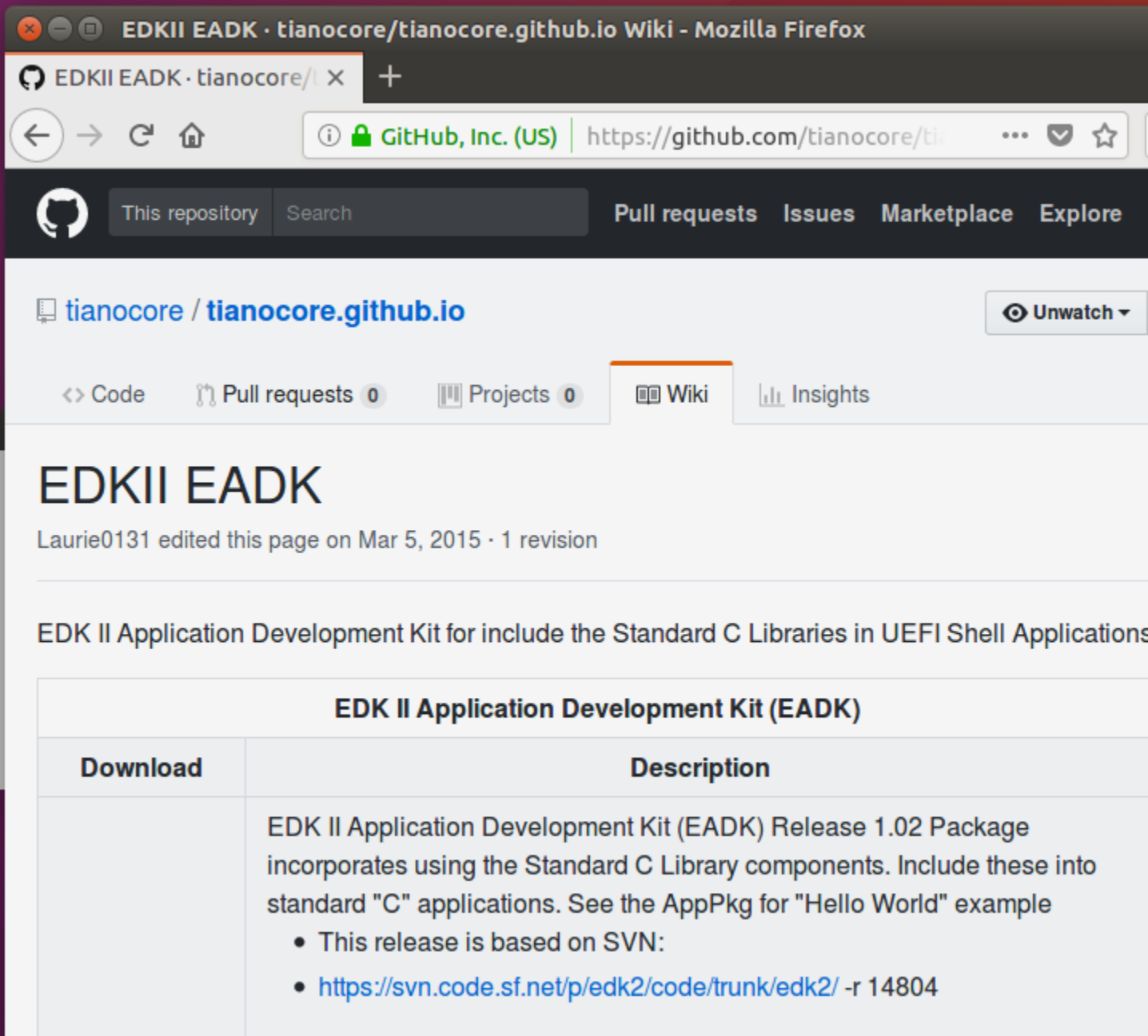


# EDK II EADK

EDK II Application Development Kit includes the Standard “C” Libraries in UEFI Shell Applications

Link: [wiki EADK](#)

Github: [edk2-libc](#)



The screenshot shows a web browser window displaying the GitHub Wiki page for the EDKII EADK repository. The browser's address bar shows the URL <https://github.com/tianocore/tianocore.github.io/wiki/EDKII-EADK>. The page title is "EDKII EADK" and it indicates that Laurie0131 edited this page on Mar 5, 2015, with 1 revision. The main content of the page is a table titled "EDK II Application Development Kit (EADK)".

Download	Description
	<p>EDK II Application Development Kit (EADK) Release 1.02 Package incorporates using the Standard C Library components. Include these into standard "C" applications. See the AppPkg for "Hello World" example</p> <ul style="list-style-type: none"><li>• This release is based on SVN:</li><li>• <a href="https://svn.code.sf.net/p/edk2/code/trunk/edk2/">https://svn.code.sf.net/p/edk2/code/trunk/edk2/</a> -r 14804</li></ul>

# EDK II EADK COMPONENTS

EDK II Application Development Kit includes the Standard C Libraries in UEFI Shell Applications

## Components

- Utilities (Python 2.7.2, & 2.7.10 etc.)
- C Library
- BSD Socket Library
- Network Socket Library – Ipv4 / Ipv6

## Packages /AppPkg /StdLib

# EDK II EADK – STANDARD ANSI C LIBRARY

## FreeBSD Port

## ANSI/POSIX compliant

System I/O	- open(), read(), write(), close(), stat()
Standard I/O	- fopen(), printf(), gets(), getchar(), . . .
String/Char	- strcmp(), isascii(), atoi(), . . .
Memory	- malloc(), free(), realloc(), . . .
Time/Date	- time(), asctime(), ctime(), . . .
Math	- sqrt(), pow(), sin(), log(), . . .