



# Introducing Embedded Firmware Development Using TianoCore's UEFI Implementation on the EDK2 Platform

Course: Embedded Firmware Development with UEFI

Student Teaching Assistant: [saigovardhanmc@pesu.pes.edu](mailto:saigovardhanmc@pesu.pes.edu)



## Table of Contents

- 1. Introduction*
- 2. TianoCore Platform*
- 3. Installing Dependencies*
- 4. Building the OVMF Firmware Package*
- 5. QEMU setup*
- 6. QEMU interfacing*
- 7. UEFI Shell*
- 8. Reference Manual*
- 9. Conclusion*



## 1. Introduction



### **UEFI:**

**UEFI (Unified Extensible Firmware Interface)** is a modern firmware standard replacing BIOS, providing an interface between a computer's hardware and operating system during the boot process. It supports advanced features like secure boot, faster startup, network booting, and large disk support.

UEFI enables better hardware compatibility, manageability, and extensibility in computing systems.



## 1. Introduction

**TianoCore's EDKII Platform:**



TianoCore's EDK2 (UEFI Development Kit 2) is an open-source development framework for building UEFI firmware. It provides a modular architecture, allowing developers to create firmware for various platforms and hardware configurations.

EDK2 streamlines UEFI development, promotes code reuse, and fosters collaboration, enabling the creation of advanced and customizable firmware solutions to enhance hardware and operating system interactions.



## 1. Introduction

**QEMU**



QEMU: Quick Emulator is a free open-source emulator used to run Hardware Platforms and Operating systems on a virtual machine of the host computer.

Here, we use QEMU for the TianoCore's EDKII Platform to test firmware and boot processes.



## 2. Tianocore Platform:



<https://github.com/tianocore>

The screenshot shows the GitHub profile for the **tianocore** organization. The profile includes a header with the organization's name, a "Follow" button, and statistics: 221 followers and the website <http://www.tianocore.org>. Below the header is a navigation bar with tabs for Overview, Repositories (26), Projects, Packages, and People (29). The main content area is titled "Pinned" and displays six repositories in a grid:

- edk2** (Public): EDK II, 3.5k stars, 2k forks.
- edk2-platforms** (Public): EDK II sample platform branches and tags, 426 stars, 351 forks.
- edk2-staging** (Public): EDK II new feature staging, 124 stars, 127 forks.
- edk2-non-osi** (Public): EDK II non-OSI licensed content, 45 stars, 62 forks.
- edk2-test** (Public): Test infrastructure and test cases for EDK II based firmware, 50 stars, 63 forks.
- edk2-libc** (Public): Port of libc to EDK II with applications that depend on libc, 51 stars, 46 forks.

On the right side of the profile, there is a "People" section showing a grid of 12 user avatars and a "View all" link. Below this is a "Top languages" section with a horizontal bar chart showing the distribution of languages: C, Python, Assembly, Rust, and HTML. The "Most used topics" section is partially visible at the bottom.



## 2. TianoCore Platform:



Git clone the repository to your host PC: <https://github.com/tianocore/edk2.git>

tianocore / edk2 Public

Watch 263 Fork 2k Star 3.5k

Code Issues 5 Pull requests 68 Discussions Actions Projects Security Insights

master 13 branches 26 tags

makubacki and mergify[bot] .github/codeql/edk2.qls: Enable CWE 12

.azurepipelines .azurepipelines/templates: Incr

.devcontainer .devcontainer/devcontainer.js

.github .github/codeql/edk2.qls: Enabl

.mergify .mergify/config.yml: Remove re

.pytool .pytool/Plugin/EccCheck: Add f

ArmPkg ArmPkg, BaseTools AARCH64: A

ArmPlatformPkg ArmPlatformPkg/PrePeiCore: M

ArmVirtPkg ArmVirtPkg/ArmPlatformLibQemu: Make IdMap.S BTI compatible 5 days ago

BaseTools BaseTools/VfrCompile: Fix potential buffer overwrites 12 hours ago

Conf BaseTools>Delete FrameworkDatabase from BaseTools/Conf 4 years ago

Local Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/tianocore/edk2.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

EDK II

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

python c firmware uefi

Readme

Unknown, Unknown licenses found

3.5k stars

263 watching

2k forks

Releases 21

edk2-stable202302 Latest on Mar 4

+ 20 releases



## 2. TianoCore Platform:



Git clone the repository to your host PC: <https://github.com/tianocore/edk2.git>

```
govardhan@DESKTOP-NLI0UAI MINGW64 ~/Desktop/tianocore
$ pwd
/c/Users/saigo/Desktop/tianocore

govardhan@DESKTOP-NLI0UAI MINGW64 ~/Desktop/tianocore
$ git clone https://github.com/tianocore/edk2.git
Cloning into 'edk2'...
remote: Enumerating objects: 378682, done.
remote: Counting objects: 100% (306/306), done.
remote: Compressing objects: 100% (182/182), done.
Receiving objects: 22% (83869/378682), 46.65 MiB | 3.34 MiB/s
```





## 2. TianoCore Platform:



`git submodule update --init`

```
MINGW64/c/Users/saigo/Desktop/tianocore/edk2
Conf/      License-History.txt  ReadMe.rst  UnitTestFrameworkPkg/
CryptoPkg/ License.txt                RedfishPkg/ edksetup.bat
DynamicTablesPkg/ Maintainers.txt  SecurityPkg/ edksetup.sh
EmbeddedPkg/ MdeModulePkg/  ShellPkg/   pip-requirements.txt

govardhan@DESKTOP-NLI0UAI MINGW64 ~/Desktop/tianocore/edk2 (master)
$ git submodule update --init
Submodule 'SoftFloat' (https://github.com/ucb-bar/berkeley-softfloat-3.git) regi
sterd for path 'ArmPkg/Library/ArmSoftFloatLib/berkeley-softfloat-3'
Submodule 'BaseTools/Source/C/BrotliCompress/brotli' (https://github.com/google/
brotli) registered for path 'BaseTools/Source/C/BrotliCompress/brotli'
Submodule 'CryptoPkg/Library/OpensslLib/openssl' (https://github.com/openssl/ope
nssl) registered for path 'CryptoPkg/Library/OpensslLib/openssl'
Submodule 'MdeModulePkg/Library/BrotliCustomDecompressLib/brotli' (https://githu
b.com/google/brotli) registered for path 'MdeModulePkg/Library/BrotliCustomDecom
pressLib/brotli'
Submodule 'MdeModulePkg/Universal/RegularExpressionDxe/oniguruma' (https://githu
b.com/kkos/oniguruma) registered for path 'MdeModulePkg/Universal/RegularExpress
ionDxe/oniguruma'
Submodule 'RedfishPkg/Library/JsonLib/jansson' (https://github.com/akheron/janss
on) registered for path 'RedfishPkg/Library/JsonLib/jansson'
Submodule 'UnitTestFrameworkPkg/Library/CmockaLib/cmocka' (https://github.com/ti
anocore/edk2-cmocka.git) registered for path 'UnitTestFrameworkPkg/Library/Cmock
aLib/cmocka'
```



## 2. TianoCore Platform:



Editing the config file

C:\Users\saigo\Desktop\tianocore\edk2\Conf

**Change the  
Active Platform  
to OVMF and  
the Target  
Architecture to  
your host PC's  
system  
architecture.**

```
# files.
# ACTIVE_PLATFORM = EmulatorPkg/EmulatorPkg.dsc
ACTIVE_PLATFORM = OvmfPkg/OvmfPkgX64.dsc
# TARGET List Optional Zero or more of the fol
# UserDefined; separated
# If the line is missing
```

```
# If not sp
# in the pl
# built.
# TARGET_ARCH = IA32
TARGET_ARCH = X64
# TOOL_DEFINITION_FILE Filename Optional Specify the
# the tools t
# WORKSPACE/C
TOOL_CHAIN_CONF = Conf/tools_def.txt
```



### 3. Installing Dependencies



Downloading other dependencies:

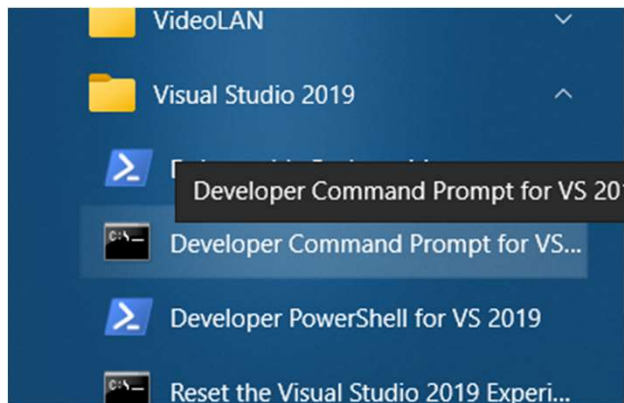
ASL - <https://github.com/tianocore/tianocore.github.io/wiki/Asl-Setup>

NASM - <https://github.com/tianocore/tianocore.github.io/wiki/Nasm-Setup>

Python 27

Visual Studio 2019

Open VS Shell





## 4. Building the OVMF Firmware Package (OVMF: Open Virtual Machine Firmware Package)



Enter the following commands into the VS Shell

```
cd C:\Users\saigo\Desktop\tianocore\toolchain\edk2
```

```
set PYTHON_HOME=C:\Python27\
```

```
set NASM_PREFIX=C:\NASM\
```

```
edksetup.bat
```

```
build -t VS2019
```

```
Developer Command Prompt for VS 2019
C:\Users\saigo\Desktop\tianocore\edk2>
C:\Users\saigo\Desktop\tianocore\edk2>cd C:\Users\saigo\Desktop\tianocore\edk2
C:\Users\saigo\Desktop\tianocore\edk2>set PYTHON_HOME=C:\Python27\
C:\Users\saigo\Desktop\tianocore\edk2>set NASM_PREFIX=C:\NASM\
C:\Users\saigo\Desktop\tianocore\edk2>edksetup.bat
PATH = C:\Users\saigo\Desktop\tianocore\edk2\BaseTools\BinWrappers\WindowsLike;C:\Users\saigo\Desktop\tianocore\edk2\BaseTools\Bin\Win32;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\Extensions\Microsoft\IntelliCode\CLI;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\bin\Win32\x86;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\VC\VC Packages;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\CommonExtensions\Microsoft\TestWindow;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team Explorer;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\bin\Roslyn;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Team Tools\Performance Tools;C:\Program Files (x86)\Microsoft Visual Studio\Shared\Common\VSPerfCollectionTools;C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.8 Tools;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\CommonExtensions\Microsoft\FSharp\Tools;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Tools\devinit;C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x86;C:\Program Files (x86)\Windows Kits\10\bin\x86;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin;C:\Windows\Microsoft.NET\Framework\v4.0.30319;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE;C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\Tools;C:\Program Files (x86)\Eclipse Foundation\jdk-8.0.302\bin;C:\Program Files\Python38\Scripts;C:\Program Files\Python38;C:\Program Files (x86)\Intel\oneAPI\tbb\latest\redist\intel64\vc_mt;C:\Program Files (x86)\Intel\oneAPI\tbb\latest\redist\ia32\vc_mt;C:\Program Files (x86)\Intel\oneAPI\compiler\latest\windows\redist\intel64_win\compiler;C:\Program Files (x86)\Intel\oneAPI\compiler\latest\windows\redist\ia32_win\compiler;C:\Program Files\Common Files\Oracle\Java\javapath;C:\WINDOWS\system32;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0;C:\WINDOWS\System32\OpenSSH;C:\Program Files\MATLAB\R2021b\bin;C:\Program Files (x86)\Calibre2;C:\Program Files (x86)\IVI Foundation\VISA\WinNT\Bin;C:\Program Files\IVI Foundation\Win64\Bin;C:\Program Files\Git\cmd;C:\Program Files\PuTTY;C:\Program Files\Microsoft SQL Server\130\Tools\Binn;C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Binn;C:\Users\saigo\AppData\Local\Programs\Python\Python38\Scripts;C:\Users\saigo\AppData\Local\Programs\Python\Python38
```



## 4. Building the OVMF Firmware Package



Enter the following commands into the VS Shell  
Start the Build:

**build -t VS2019**

```
C:\Users\saigo\Desktop\tianocore\edk2>build -t VS2019
Build environment: Windows-10-10.0.19044-SP0
Build start time: 10:11:44, Apr.04 2023

WORKSPACE      = c:\users\saigo\desktop\tianocore\edk2
EDK_TOOLS_PATH  = c:\users\saigo\desktop\tianocore\edk2\basetools
EDK_TOOLS_BIN   = c:\users\saigo\desktop\tianocore\edk2\basetools\bin\win32
CONF_PATH      = c:\users\saigo\desktop\tianocore\edk2\conf
PYTHON_COMMAND = py -3

Processing meta-data .
Architecture(s) = X64
Build target    = DEBUG
Toolchain       = VS2019

Active Platform      = c:\users\saigo\desktop\tianocore\edk2\OvmfPkg\OvmfPkgX64.dsc
..
```

*Note: This step will take time to complete (~8 minutes)*





## 4. Building the OVMF Firmware Package



After successfully completing the build of the EDKII for OvmfPkg,

```
tianocore
Name
  edk2
  tecture

Developer Command Prompt for VS 2019
Region Name = None
Generate Region at Offset 0xE000
  Region Size = 0x1000
  Region Name = None
Padding region starting from offset 0xF000, with size 0x1000

Generate Region at Offset 0xF000
  Region Size = 0x1000
  Region Name = None

Generate Region at Offset 0x10000
  Region Size = 0x10000
  Region Name = None

Generate Region at Offset 0x20000
  Region Size = 0xE0000
  Region Name = FV

Generate Region at Offset 0x100000
  Region Size = 0xD00000
  Region Name = FV

GUID cross reference file can be found at c:\users\saigo\desktop\tianocore\edk2\Build\OvmfX64\DEBUG_VS2019\FV\Guid.xref

FV Space Information
SECFV [31%Full] 212992 (0x34000) total, 66960 (0x10590) used, 146032 (0x23a70) free
PEIFV [36%Full] 917504 (0xe0000) total, 336680 (0x52328) used, 580824 (0x8dcd8) free
DXEFV [36%Full] 13631488 (0xd00000) total, 4997384 (0x4c4108) used, 8634104 (0x83bef8) free
FVMAIN_COMPACT [36%Full] 3440640 (0x348000) total, 1257944 (0x1331d8) used, 2182696 (0x214e28) free

- Done -
Build end time: 10:20:18, Apr.04 2023
Build total time: 00:08:34

C:\Users\saigo\Desktop\tianocore\edk2>
```



## 5. QEMU setup



Install QEMU for your host PC: <https://www.qemu.org/download/#windows>

### Download QEMU

Source code	Linux	macOS	Windows
-------------	-------	-------	---------

Stefan Weil provides binaries and installers for both **32-bit** and **64-bit** Windows.

MSYS2:

QEMU can be installed using **MSYS2** also. MSYS2 uses **pacman** to manage packages. First, follow the **MSYS2** installation procedure. Then update the packages with `pacman -Syu` command. Now choose the proper command for your system as following:

- For 32 bit Windows 7 or above (in MINGW32):

```
pacman -S mingw-w64-i686-qemu
```

- For 64 bit Windows 7 or above (in MINGW64):

```
pacman -S mingw-w64-x86_64-qemu
```

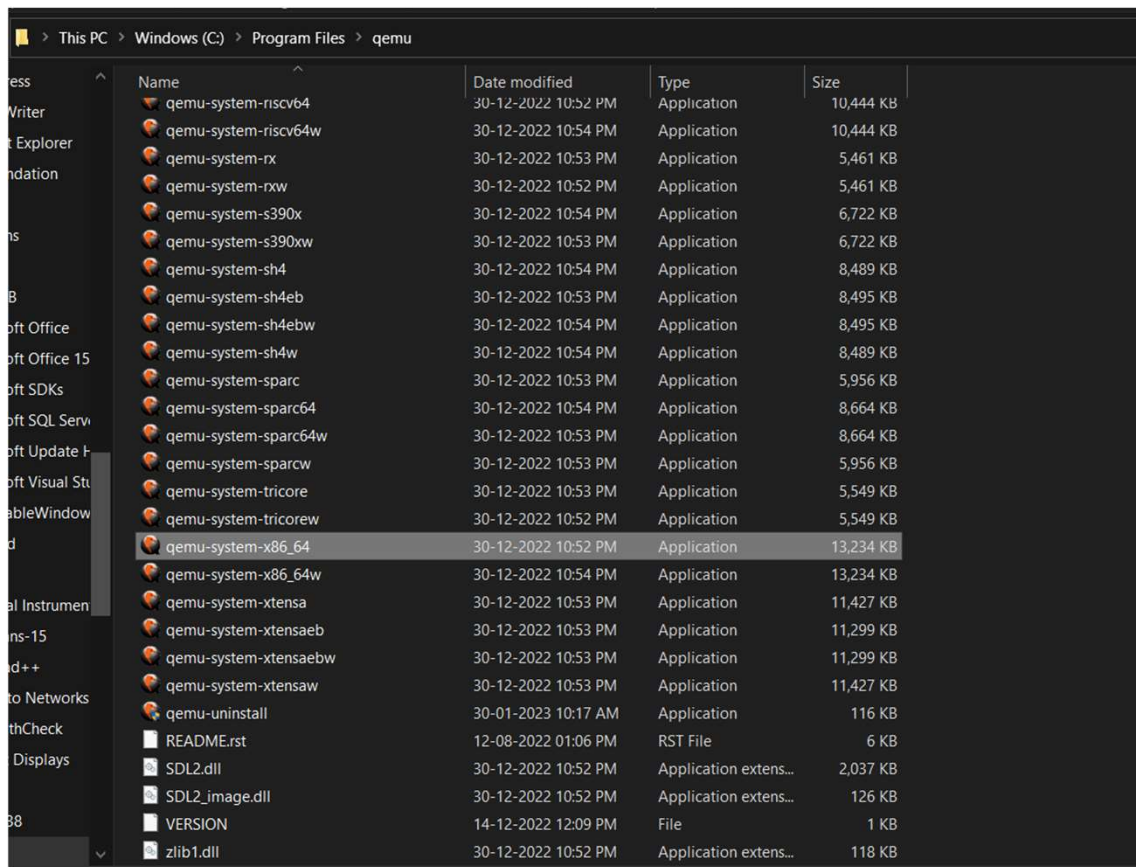
**Setup QEMU and provide the install location**



## 5. QEMU setup



The *qemu-system-x86\_64* application can be located in the install path (Program Files)



The screenshot shows a Windows File Explorer window with the address bar set to 'This PC > Windows (C:) > Program Files > qemu'. The left sidebar shows the 'qemu' folder selected. The main pane displays a list of files and folders with columns for Name, Date modified, Type, and Size. The file 'qemu-system-x86\_64' is highlighted.

Name	Date modified	Type	Size
qemu-system-nv64	30-12-2022 10:52 PM	Application	10,444 KB
qemu-system-riscv64w	30-12-2022 10:54 PM	Application	10,444 KB
qemu-system-rx	30-12-2022 10:53 PM	Application	5,461 KB
qemu-system-rxw	30-12-2022 10:52 PM	Application	5,461 KB
qemu-system-s390x	30-12-2022 10:54 PM	Application	6,722 KB
qemu-system-s390xw	30-12-2022 10:53 PM	Application	6,722 KB
qemu-system-sh4	30-12-2022 10:54 PM	Application	8,489 KB
qemu-system-sh4eb	30-12-2022 10:53 PM	Application	8,495 KB
qemu-system-sh4ebw	30-12-2022 10:54 PM	Application	8,495 KB
qemu-system-sh4w	30-12-2022 10:54 PM	Application	8,489 KB
qemu-system-sparc	30-12-2022 10:53 PM	Application	5,956 KB
qemu-system-sparc64	30-12-2022 10:54 PM	Application	8,664 KB
qemu-system-sparc64w	30-12-2022 10:53 PM	Application	8,664 KB
qemu-system-sparcw	30-12-2022 10:53 PM	Application	5,956 KB
qemu-system-tricore	30-12-2022 10:53 PM	Application	5,549 KB
qemu-system-tricorew	30-12-2022 10:52 PM	Application	5,549 KB
qemu-system-x86_64	30-12-2022 10:52 PM	Application	13,234 KB
qemu-system-x86_64w	30-12-2022 10:54 PM	Application	13,234 KB
qemu-system-xtensa	30-12-2022 10:53 PM	Application	11,427 KB
qemu-system-xtensaeb	30-12-2022 10:53 PM	Application	11,299 KB
qemu-system-xtensaebw	30-12-2022 10:53 PM	Application	11,299 KB
qemu-system-xtensaw	30-12-2022 10:53 PM	Application	11,427 KB
qemu-uninstall	30-01-2023 10:17 AM	Application	116 KB
README.rst	12-08-2022 01:06 PM	RST File	6 KB
SDL2.dll	30-12-2022 10:52 PM	Application extens...	2,037 KB
SDL2_image.dll	30-12-2022 10:52 PM	Application extens...	126 KB
VERSION	14-12-2022 12:09 PM	File	1 KB
zlib1.dll	30-12-2022 10:52 PM	Application extens...	118 KB





## 6. QEMU interfacing



Open the cmd prompt at the qemu directory and load the build to be emulated on QEMU using the following command

```
qemu-system-x86_64 -cpu qemu64 -drive  
if=pflash,format=raw,unit=0,file=C:\Users\saigo\Desktop\tianocore\toolch  
ain\edk2\Build\OvmfX64\DEBUG_VS2019\FV\OVMF_CODE.fd,readonly=on -drive  
if=pflash,format=raw,unit=1,file=C:\Users\saigo\Desktop\tianocore\toolch  
ain\edk2\Build\OvmfX64\DEBUG_VS2019\FV\OVMF_VARS.fd -net none
```



## 6. QEMU interfacing



The UEFI interactive shell will be displayed when the build is successfully virtualised. Note that this is a command line interface (like bash)

```
QEMU - Press Ctrl+Alt+G to release grab
Machine View

UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
BLK0: Alias(s):
      PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> _

C:\Windows\System32\cmd.exe - qemu-system-x86_64 -cpu qemu64 -drive if=pflash,format=raw,unit=0,file=C:\U...
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\qemu>qemu-system-x86_64 -cpu qemu64 -drive if=pflash,format=raw,unit=0,file=C:\Users\sai
go\Desktop\tianocore\toolchain\edk2\Build\OvmfX64\DEBUG_VS2019\FV\OVMF_CODE.fd,readonly=on -drive if=pflash
,format=raw,unit=1,file=C:\Users\saigo\Desktop\tianocore\toolchain\edk2\Build\OvmfX64\DEBUG_VS2019\FV\OVMF_
VARS.fd -net none

(qemu:22384): Gtk-WARNING **: 10:38:22.525: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
```



## 7. UEFI Shell



The UEFI interactive shell will be displayed when the build is successfully virtualised. Note that this is a command line interface (like bash)

```
Shell> ver
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Shell> _
```

```
BLK0: Alias(s) :
      PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> echo "This is a UEFI Demo for PESU elective Embedded Firmware Dev with UEFI"
This is a UEFI Demo for PESU elective Embedded Firmware Dev with UEFI
Shell> _
```



## 7. UEFI Shell



The UEFI interactive shell will be displayed when the build is successfully virtualised. Note that this is a command line interface (like bash)  
Below, I have displayed the memory map (\$map)  
and the boot options (\$bcfg boot dump)

```
Optional- N
Shell> map
Mapping table
  BLK0: Alias(s) :
    PciRoot (0x0) / Pci (0x1,0x1) / Ata (0x0)
Shell> bcfg boot dump
Option: 00. Variable: Boot0000
  Desc - UiApp
  DevPath - Fv (7CB8BDC9-F8EB-4F34-AAEA-3EE4AF6516A1) / FvFile (462CAA21-7614-4503-836E-8AB6F4662331)
  Optional- N
Option: 01. Variable: Boot0001
  Desc - UEFI QEMU DVD-ROM QM00003
  DevPath - PciRoot (0x0) / Pci (0x1,0x1) / Ata (0x0)
  Optional- Y
Option: 02. Variable: Boot0002
  Desc - EFI Internal Shell
  DevPath - Fv (7CB8BDC9-F8EB-4F34-AAEA-3EE4AF6516A1) / FvFile (7C04A583-9E3E-4F1C-AD65-E05268D0B4D1)
  Optional- N
Shell> _
```



## 7. UEFI Shell



Other examples:

Type: too few arguments

Shell> mem

Memory Address 00000000079EE018 78 Bytes

```
079EE018: 49 42 49 20 53 59 53 54-46 00 02 00 78 00 00 00 *IBI SYSTF...x...*
079EE028: 8B 99 07 57 00 00 00 00-18 FF 95 07 00 00 00 00 *...W.....*
079EE038: 00 00 01 00 00 00 00 00-98 6E DB 06 00 00 00 00 *.....n.....*
079EE048: 40 0E D3 06 00 00 00 00-98 E6 0D 07 00 00 00 00 *@.....*
079EE058: 20 AE B4 06 00 00 00 00-18 59 DB 06 00 00 00 00 * .....Y.....*
079EE068: A0 11 D3 06 00 00 00 00-98 EB 9E 07 00 00 00 00 *.....*
079EE078: F0 5D EB 07 00 00 00 00-09 00 00 00 00 00 00 *.I.....*
079EE088: 98 EC 9E 07 00 00 00 00- *.....*
```

Valid EFI Header at Address 00000000079EE018

System: Table Structure size 00000078 revision 00020046

ConIn (0000000006D30E40) ConOut (0000000006B4AE20) StdErr (0000000006D311A0)

Runtime Services 00000000079EEB98

Boot Services 0000000007EB5DF0

SAL System Table 0000000000000000

ACPI Table 0000000007B7E000

ACPI 2.0 Table 0000000007B7E014

MPS Table 0000000000000000

SMBIOS Table 000000000795E000

Shell>

Shell> devices

```
T D
Y C I
P F A
CTRL E G G #P #D #C Device Name
=====
33 R - - 0 1 5 PciRoot(0x0)
66 D - - 2 0 0 Primary Console Input Device
67 D - - 2 0 0 Primary Console Output Device
68 D - - 1 0 0 Primary Standard Error Device
95 D - - 1 0 0 PciRoot(0x0)/Pci(0x0,0x0)
96 B - - 1 1 3 PciRoot(0x0)/Pci(0x1,0x0)
97 B - - 1 4 1 Sata Controller
98 D - - 1 0 0 PciRoot(0x0)/Pci(0x1,0x3)
99 B - - 1 1 1 QEMU Video PCI Adapter
9B B - - 1 3 1 PciRoot(0x0)/Pci(0x2,0x0)/AcpiAdr(0x80010100)
9E B - - 1 1 1 PciRoot(0x0)/Pci(0x1,0x0)/Serial(0x0)
9F D - - 1 0 0 PciRoot(0x0)/Pci(0x1,0x0)/Serial(0x1)
A0 B - - 1 3 1 PS/2 Keyboard Device
A1 B - - 1 1 1 SIO Serial Port #0
A2 B - - 1 5 3 PC-ANSI Serial Console
A3 D - - 1 2 0 SCSI Disk Device
```



## 7. UEFI Shell



You can additionally use (\$help) to display a list of all the commands, Reset (\$reset), Shut Down (\$shutdown)

```
Optional- N
Shell> map
Mapping table
  BLK0: Alias(s):
        PciRoot (0x0) / Pci (0x1,0x1) / Ata (0x0)
Shell> bcfg boot dump
Option: 00. Variable: Boot0000
  Desc    - UiApp
  DevPath - Fv (7CB8BDC9-F8EB-4F34-AAEA-3EE4AF6516A1) / FvFile (462CAA21-7614-4503-836E-8AB6F4662331)
Optional- N
Option: 01. Variable: Boot0001
  Desc    - UEFI QEMU DVD-ROM QM00003
  DevPath - PciRoot (0x0) / Pci (0x1,0x1) / Ata (0x0)
Optional- Y
Option: 02. Variable: Boot0002
  Desc    - EFI Internal Shell
  DevPath - Fv (7CB8BDC9-F8EB-4F34-AAEA-3EE4AF6516A1) / FvFile (7C04A583-9E3E-4F1C-AD65-E05268D0B4D1)
Optional- N
Shell> _
```



## 8. Reference Manual



All the reference commands can be found listed in the UEFI Shell Specification:  
[https://uefi.org/sites/default/files/resources/UEFI Shell 2 2.pdf](https://uefi.org/sites/default/files/resources/UEFI%20Shell%202.2.pdf)

*Note: Chapter 5 is a good collation of all the commands*

### 5

## Shell Commands



### UEFI Shell Specification

#### 5.1

#### Overview

This section describes the standard UEFI Shell commands.

The table below lists all standard UEFI Shell commands.

**Table 11 Commands from Default Build Shell**

Command	Description	Required at Shell Level or Profile
<b>alias</b>	Displays, creates, or deletes aliases in the UEFI Shell environment	3
<b>attrib</b>	Displays or changes the attributes of files or directories.	2
<b>bcfg</b>	Manipulate boot order and driver order	Debug1,



## 9. Conclusion



In this presentation, we have covered the UEFI toolchain installation, build procedure for OVMF Firmware and interfaced it with QEMU with basic shell commands.

This can be used for building custom .efi applications on a firmware package and can be emulated virtually on QEMU.





*Thank you*

Course: Embedded Firmware Development with UEFI

Student Teaching Assistant: [saigovardhanmc@pesu.pes.edu](mailto:saigovardhanmc@pesu.pes.edu)