






UEFI Basic Tutorial (Part 2) - Running the first APP HelloWorld

 xiaopangzi313

 Posted on 2019-04-27 01:17:58

 Read 1w


 Collection 40


 Likes 8

Category columns: 15_Firmware Development

Article Tags: UEFI firmware bios FirmWare

Copyright CC 4.0 BY-SA

 2048 AI Community The article has been collected by the community

 15_Firmware Development This column includes this content

Join the community

27 articles

Subscribe to our column



This article details how to write and run a simple UEFI application from scratch in Windows and Linux environments, including the entire process of writing source code, compiling to generate EFI files, and running EFI applications in QEMU using a virtual USB flash drive.

The summary is generated in [C Know](#) , supported by DeepSeek-R1 full version, [go to experience](#)>

1. Windows platform

1. Write source code

1. Write C:\edkii\OvmfPkg\HelloWorld\HelloWorld.c

AI generated projects 登录复制

```
1 #include <Library/UefiLib.h>
2 #include <Library/BaseLib.h>
3 #include <Library/DebugLib.h>
4 #include <Library/BaseMemoryLib.h>
5 #include <Library/UefiBootServicesTableLib.h>
6
7 //ShellCEntryLib call user interface ShellAppMain
8 EFI_STATUS
9 EFIAPI
10 HelloWorldEntry(
11     IN EFI_HANDLE      ImageHandle,
12     IN EFI_SYSTEM_TABLE *SystemTable
13 )
14 {
15     EFI_STATUS Status = EFI_SUCCESS;
16     Print (L"[Console] HelloWorldEntry Start...\n");
17
18     Print (L"[Console] HelloWorldEntry End ... \n");
```

```
19 | return Status;
20 | }
twen |
◀ ● ▶
```

收起 ^

2. Write C:\edkii\OvmfPkg\HelloWorld\HelloWorld.inf

```
1 | [Defines]
2 |   INF_VERSION = 0x00010007
3 |   BASE_NAME = HelloWorld
4 |   FILE_GUID = 69A6DE6D-FA9F-485E-9A4E-EA70FDCFD82F
5 |   MODULE_TYPE = UEFI_APPLICATION
6 |   VERSION_STRING = 1.0
7 |   ENTRY_POINT = HelloWorldEntry
8 |
9 | [Sources]
10 |   HelloWorld.c
11 |
12 | [Packages]
13 |   MdePkg/MdePkg.dec
14 |   ShellPkg/ShellPkg.dec
15 |   MdeModulePkg/MdeModulePkg.dec
16 |
17 | [LibraryClasses]
18 |   UefiShellCEntryLib
19 |   BaseLib
20 |   BaseMemoryLib
twen |   DebugLib
twen |   PrintLib
twen |   UefiBootServicesTableLib
twen |   MemoryAllocationLib
25 |   UefiLib
◀ ● ▶
```

收起 ^

3. Revise C:\edkii\OvmfPkg\OvmfPkgX64.dsc

```
1 | #
2 | [Components]
3 |   OvmfPkg/HelloWorld/HelloWorld.inf
```

2. Compile and generate EFI files

1. Compile UFI file

Run and `edksetup.bat` compile the entire OvmfPkg Package

```
build -a X64 -p OvmfPkg\OvmfPkgX64.dsc -D DEBUG_ON_SERIAL_PORT
```

2. Check the efi file

```
c:\edkii>dir Build\OvmfX64\DEBUG_VS2013x86\X64\OvmfPkg\HelloWorld\HelloWorld\OUTPUT\*.efi
Volume in drive C is OSDisk
Volume Serial Number is 2A88-3D76

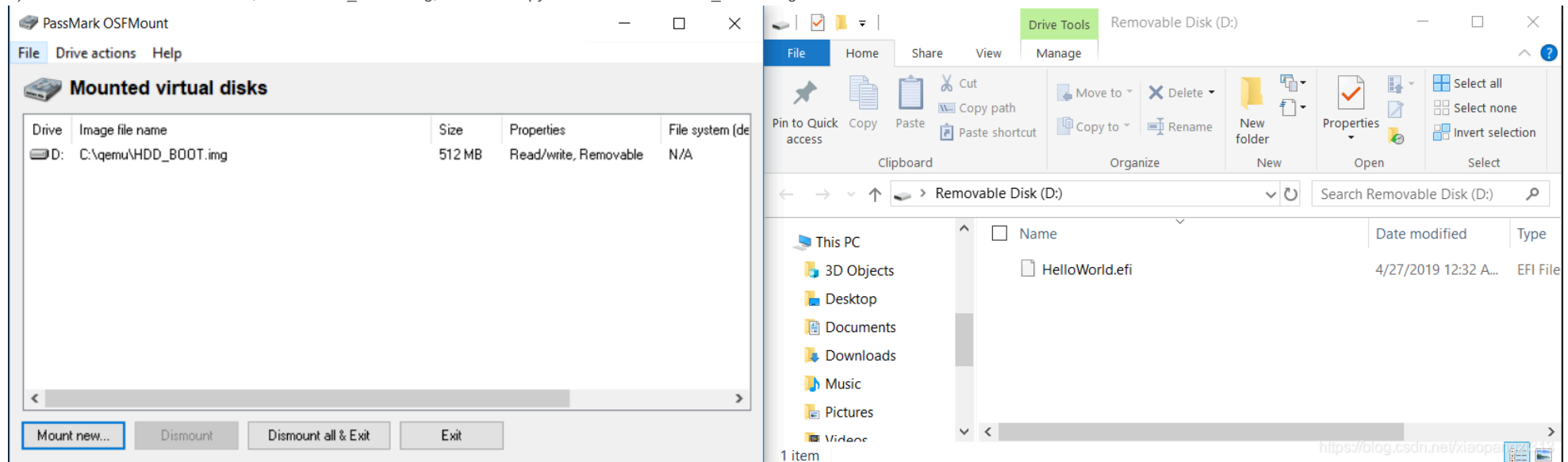
Directory of c:\edkii\Build\OvmfX64\DEBUG_VS2013x86\X64\OvmfPkg\HelloWorld\HelloWorld\OUTPUT

04/27/2019  12:32 AM                10,464 HelloWorld.efi
             1 File(s)                10,464 bytes
             0 Dir(s)  12,076,789,760 bytes free
```

3. Run HelloWorld.efi

1. Create a virtual USB disk

- 1) Download and install [UltraISO.exe](#) to create the image file HDD_BOOT.img,
- 2) Download and install osfmount, mount HDD_BOOT.img, and then copy HelloWorld.efi to HDD_BOOT.img



2. Mount the virtual USB drive to start QEMU

- 1) Rewrite the startup script `setup-qemu-x64.bat`,

```
C:\qemu>echo "C:\Program Files\qemu\qemu-system-x86_64.exe" -bios "OVMF.fd" -M "pc" -m 256 -cpu "qemu64" -boot order=dc -usbdevice disk:HDD_BOOT.img -serial stdio > setup-qemu-x64.bat
```

3. The results of running HelloWorld.efi

after startup are as follows:

```
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
  FS0: Alias(s) :HD2a0::BLK3:
      PciRoot (0x0) /Pci (0x1,0x2) /USB (0x0,0x0)
  BLK2: Alias(s) :
      PciRoot (0x0) /Pci (0x1,0x1) /Ata (0x0)
  BLK0: Alias(s) :
      PciRoot (0x0) /Pci (0x1,0x0) /Floppy (0x0)
  BLK1: Alias(s) :
      PciRoot (0x0) /Pci (0x1,0x0) /Floppy (0x1)
Press ESC in 1 seconds to skip startup.nsh or any other key to continue.
Shell> fs0:
FS0:\> ls
Directory of: FS0:\
04/27/2019  00:32           10,464  HelloWorld.efi
               1 File(s)          10,464 bytes
               0 Dir(s)
FS0:\> HelloWorld.efi
[Console]  HelloWorldEntry Start..
[Console]  HelloWorldEntry End ...
FS0:\> _
```

<https://blog.csdn.net/xiaopangzi313>

2. Linux platform

1. Write source code (enter the following command in shell)

AI generated projects

登录复制

```
1 # 新建 HelloWorld 目录
2 mkdir -p OvmfPkg/HelloWorld/
```

AI generated projects

登录复制

```
1 # 新建HelloWorld.c
2 echo '
3 #include <Library/UefiLib.h>
4 #include <Library/BaseLib.h>
5 #include <Library/DebugLib.h>
6
```

7~
8
9

#include <Library/BaseMemoryLib.h>
#include <Library/UefiBootServicesTableLib.h>

展开

AI generated projects

登录复制

1
2
3
4
5
6
7
8
9

新建HelloWorld.inf
echo '
[Defines]
INF_VERSION = 0x00010007
BASE_NAME = HelloWorld
FILE_GUID = 69A6DE6D-FA9F-485E-9A4E-EA70FDCFD82F
MODULE_TYPE = UEFI_APPLICATION
VERSION_STRING = 1.0
ENTRY_POINT = HelloWorldEntry

展开

AI generated projects

登录复制

1
2

添加 HelloWorld.inf 到OvmfPkg/OvmfPkgX64.dsc
sed "s/[Components\]/\[Components\]\nOvmfPkg\HelloWorld\HelloWorld.inf/" -i OvmfPkg/OvmfPkgX64.dsc

展开

AI generated projects

登录复制

1
2

添加 UefiShellEntryLib.inf到OvmfPkg/OvmfPkgX64.dsc
sed "s/[LibraryClasses\]/\[LibraryClasses\]\nUefiShellEntryLib\ShellPkg\Library\UefiShellEntryLib\UefiShellEntryLib.inf/" -i OvmfPkg/OvmfPkgX64.dsc

展开

2. Compile EFI APP

AI generated projects

登录复制

1
2

make -C /root/edk2/BaseTools/Source/C
./OvmfPkg/build.sh -D DEBUG_ON_SERIAL_PORT

展开

3. In QEMU operation EFI

1. Create a virtual USB device file HDD_BOOT.img

AI generated projects

登录复制

1
2
3
4
5
6
7
8
9

新建HDD_BOOT.img的空文件
qemu-img create -f raw HDD_BOOT.img 64M
格式化img文件
mkfs.vfat HDD_BOOT.img
将文件加载到设备文件
losetup /dev/loop8 HDD_BOOT.img
#新建挂载目录
mkdir -p /mnt/hello

展开

2. Copy HelloWorld.efi to HDD_BOOT.img

AI generated projects

登录复制

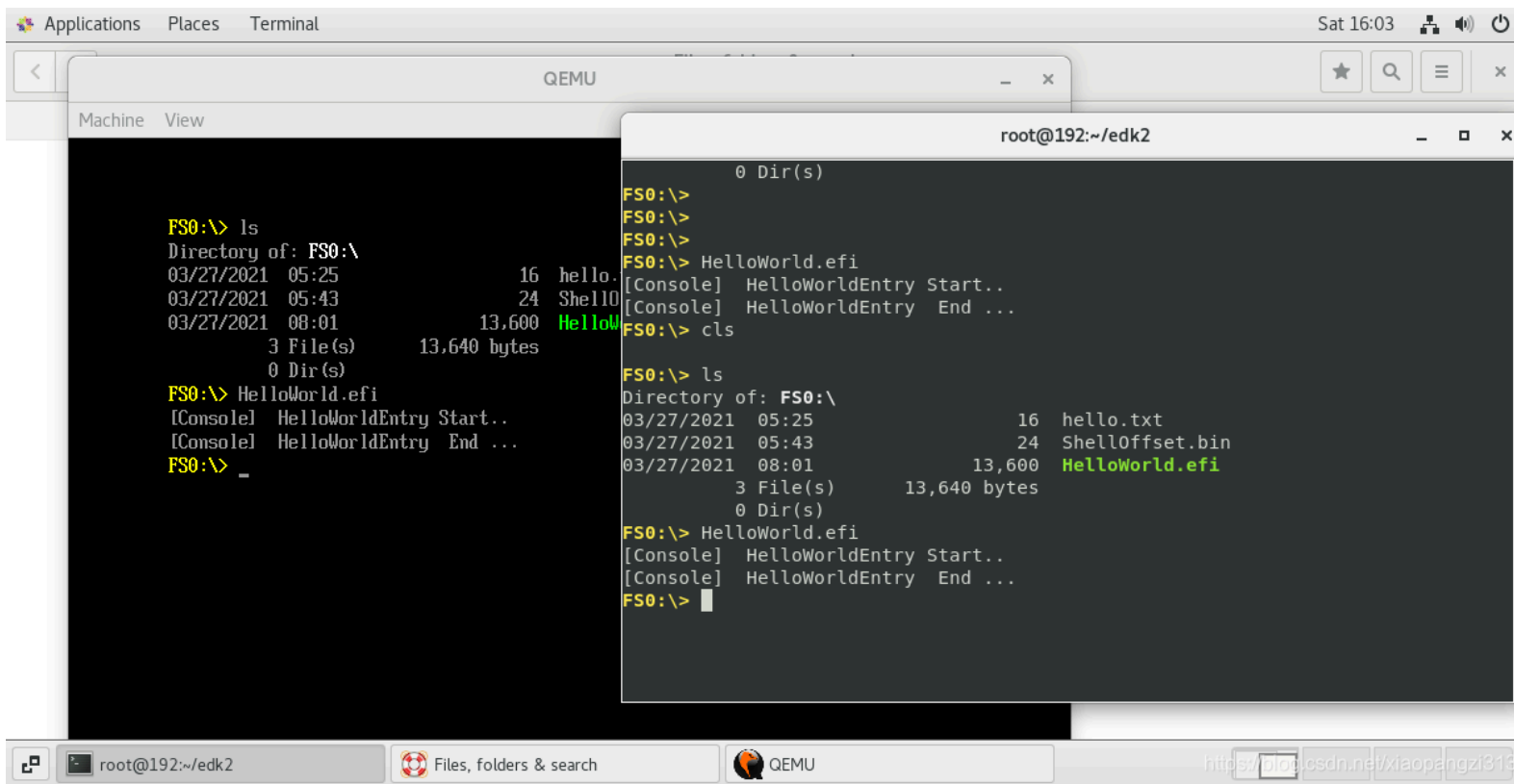
```
1 cp ./Build/OvmfX64/DEBUG_GCC48/X64/OvmfPkg/HelloWorld/HelloWorld/DEBUG/HelloWorld.efi /mnt/hello
2 # 将efi文件回写到 HDD_BOOT.img
3 umount /dev/loop8
4 losetup -d /dev/loop8
```

3. UEFI loads HDD_BOOT.img and runs

AI generated projects

登录复制

```
1 qemu-system-x86_64 -bios ./Build/OvmfX64/DEBUG_GCC48/FV/OVMF.fd \
2 -serial stdio -usb -drive if=none,format=raw,id=disk1,file=HDD_BOOT.img \
3 -device usb-storage,drive=disk1
```



AI副业到底有多牛？想转行却无从下手？

从LLM技术原理到多模态应用开发！三天掌握AI获客/数字人开发/接单渠道，开启第二职业收入管道

广告