I debugging environment construction

yao00037 Posted on 2022-12-19 15:35:46 Read 1.5k Collection 3 Likes

Category columns: UEFI edk2 Article Tags: linux Operations windows

UEFI Included in 2 columns at the same time 2 articles Subscri

our col
our colu

摘要 This article details how to use QEMU and OVMF images with the GDB tool in a Linux environment to debug the HelloWorld program of MdeModulePkg, incl ling debugging code, compiling the OVMF image, setting the debugging environment, and setting breakpoints for in-depth troubleshooting.

ummary is generated in C Know , supported by DeepSeek-R1 full version, go to experience>

EMU and OVMF image under Linux to debug UEFI program directly with GDB

elloWorld under MdeModulePkg as the target debugging program as an example.

Add debugging code in HelloWorld.c

inly used to assist debugging, such as determining whether symbols are loaded correctly, whether the program is executed correctly, etc.
ader files:

AI generated projects 登录复制

```
#include <Library/DebugLib.h>
```

og to obtain the program entry address in the main program UefiMain.

AI generated projects 登录复制

```
Print(L"HelloWorld:My Entry Point is 0x%08x\n", (CHAR16 *)UefiMain);
```

Compile the OVMF image

dd HelloWorl to OvmfPkg for easy compilation. Open OvmfPkgX64. dsc and add after
onents]:

AI generated projects 登录复制

```
  MdeModulePkg/Application/HelloWorld/HelloWorld.inf {
    <LibraryClasses>
    DebugLib|MdePkg/Library/UefiDebugLibConOut/UefiDebugLibConOut.inf
  }
```

ompile OVMF:

AI generated projects 登录

```
source edksetup.sh
build -p OvmfPkg/OvmfPkgX64.dsc -t GCC5 -a X64 -b DEBUG
```

: Prepare the debug folder

AI generated projects 登录

```
mkdir ovmf_dbg
cd ovmf_dbg
mkdir hda-contents
cp ../Build/OvmfX64/DEBUG_GCC5/FV/OVMF.fd ./
cp ../Build/OvmfX64/DEBUG_GCC5/X64/HelloWorld.* ./hda-contents/
```

: Start QEMU and run UEFI Shell

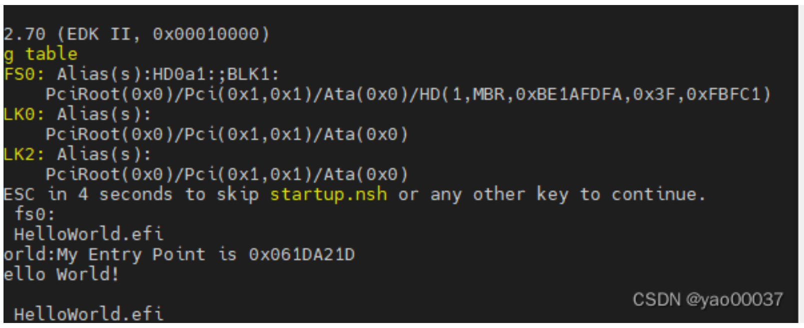
AI generated projects 登录

```
qemu-system-x86_64 -s -pflash OVMF.fd -hadfat:rw:hda-contents/ -net none -debugcon file:debug.log -global isa-debugcon.iobase=0
```

he UEFI shell and execute the HelloWorld.efi program. The content added in step 1 will be printed out. At the same time, check the location where
/orld.efi is loaded in debug.log , which will be used later.

AI generated projects 登录

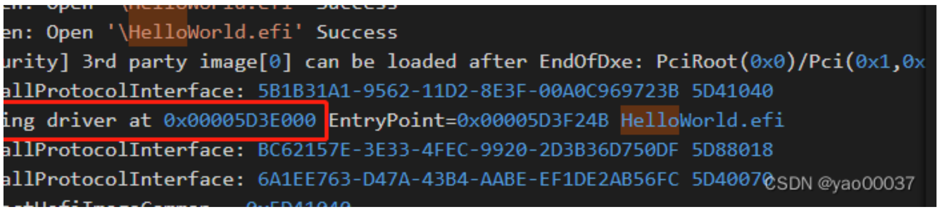
```
fs0:
HelloWorld.efi
```



```
2.70 (EDK II, 0x00010000)
g table
FS0: Alias(s):HD0a1:;BLK1:
    PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)/HD(1,MBR,0xBE1AFDFA,0x3F,0xFBFC1)
LK0: Alias(s):
    PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
LK2: Alias(s):
    PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
ESC in 4 seconds to skip startup.nsh or any other key to continue.
 fs0:
 HelloWorld.efi
orld:My Entry Point is 0x061DA21D
ello World!

 HelloWorld.efi
```

CSDN @yao00037



```
en: Open '\HelloWorld.efi' Success
urity] 3rd party image[0] can be loaded after EndOfDxe: PciRoot(0x0)/Pci(0x1,0x
allProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 5D41040
ing driver at 0x00005D3E000 EntryPoint=0x00005D3F24B HelloWorld.efi
allProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 5D88018
allProtocolInterface: 6A1EE763-D47A-43B4-AABE-EF1DE2AB56FC 5D40070
```

CSDN @yao00037

: Start GDB and mount the debugger

another terminal, enter the subdirectory hda-contents, and start gdb.

AI generated projects 登录

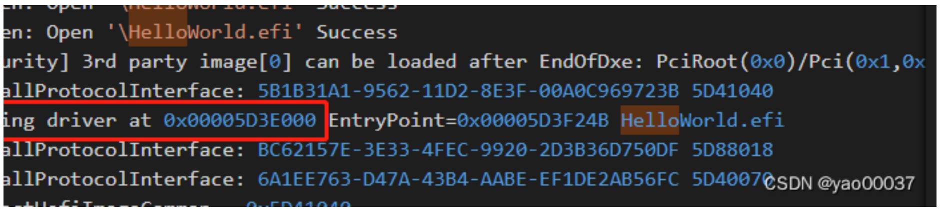
```
gdb
```

AI generated projects 登录

```
(gdb) file HelloWorld.efi
Reading symbols from HelloWorld.efi...
(No debugging symbols found in HelloWorld.efi)
(gdb) info file
Symbols from "/home/code/edk2/_UDK_gdb/hda-contents/HelloWorld.efi".
Local exec file:
        `/home/code/edk2/_UDK_gdb/hda-contents/HelloWorld.efi',
        file type pei-x86-64.
        Entry point: 0x124b
```

展开

ing to the address of the image loading obtained in the previous step:



```
en: Open '\HelloWorld.efi' Success
urity] 3rd party image[0] can be loaded after EndOfDxe: PciRoot(0x0)/Pci(0x1,0x
allProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 5D41040
ing driver at 0x00005D3E000 EntryPoint=0x00005D3F24B HelloWorld.efi
allProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 5D88018
allProtocolInterface: 6A1EE763-D47A-43B4-AABE-EF1DE2AB56FC 5D40070
```

CSDN @yao00037

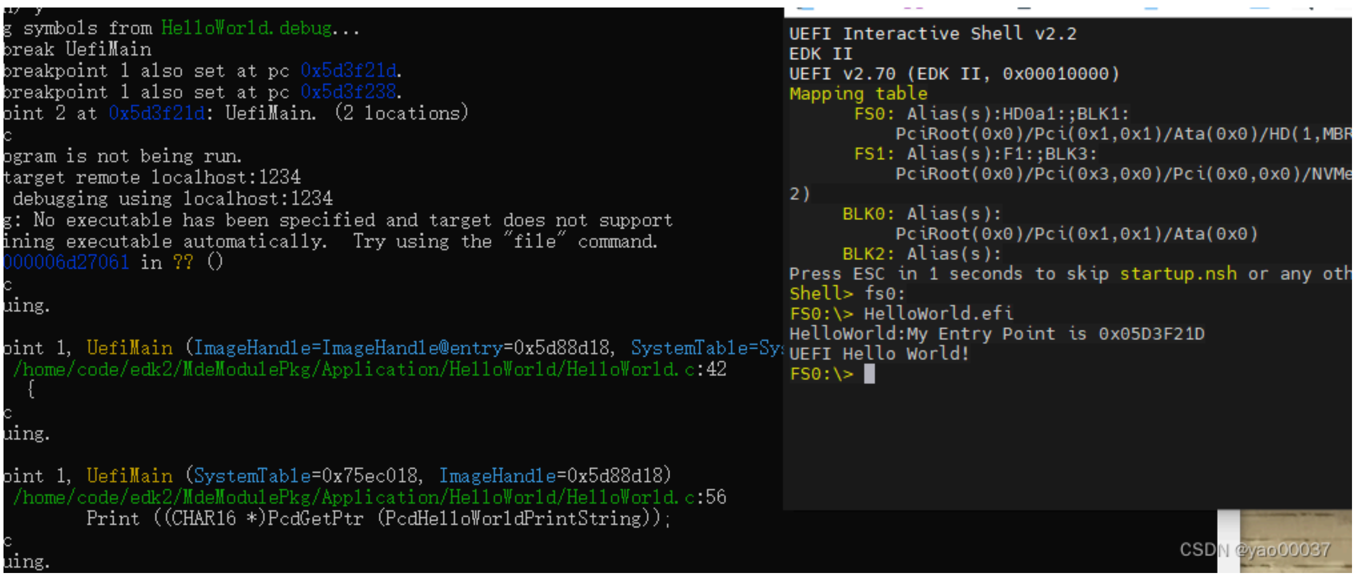
ate the offset address of the code segment and the data segment:
)x5D3E000 + 0x240 = 0x5D3E240
0x5D3E000 + 0x240 + 0x1F00 = 0x5D40140After
culation is completed, you can unload the efi file loaded by GDB.
ou can load symbols, set breakpoints , and mount GDB on QEMU to prepare for debugging.

AI generated projects 登录

```
(gdb) add-symbol-file HelloWorld.debug 0x5d3e240 -s .data 0x5d40140
add symbol table from file "HelloWorld.debug" at
```

```
        .text_addr = 0x5d3e240
        .data_addr = 0x5d40140
(y or n) y
Reading symbols from HelloWorld.debug...
(gdb) break UefiMain
Note: breakpoint 1 also set at pc 0x5d3f21d.
```

展开



```
g symbols from HelloWorld.debug...
break UefiMain
breakpoint 1 also set at pc 0x5d3f21d.
breakpoint 1 also set at pc 0x5d3f238.
oint 2 at 0x5d3f21d: UefiMain. (2 locations)
c
ogram is not being run.
target remote localhost:1234
 debugging using localhost:1234
g: No executable has been specified and target does not support
ining executable automatically.  Try using the "file" command.
000006d27061 in ?? ()
c
uing.

oint 1, UefiMain (ImageHandle=ImageHandle@entry=0x5d88d18, SystemTable=Sys
 /home/code/edk2/MdeModulePkg/Application/HelloWorld/HelloWorld.c:42
  {
c
uing.

oint 1, UefiMain (SystemTable=0x75ec018, ImageHandle=0x5d88d18)
 /home/code/edk2/MdeModulePkg/Application/HelloWorld/HelloWorld.c:56
        Print ((CHAR16 *)PcdGetPtr (PcdHelloWorldPrintString));
c
uing.
```

```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
      FS0: Alias(s):HD0a1:;BLK1:
          PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)/HD(1,MBR
      FS1: Alias(s):F1:;BLK3:
          PciRoot(0x0)/Pci(0x3,0x0)/Pci(0x0,0x0)/NVMe
2)
     BLK0: Alias(s):
          PciRoot(0x0)/Pci(0x1,0x1)/Ata(0x0)
     BLK2: Alias(s):
Press ESC in 1 seconds to skip startup.nsh or any oth
Shell> fs0:
FS0:\> HelloWorld.efi
HelloWorld:My Entry Point is 0x05D3F21D
UEFI Hello World!
FS0:\>
```

: Debug the code

ıe environment is ready, you can use gdb to perform various debugging.

ιove is an example of debugging a UEFI application. Drivers and applications can be debugged using the same method.

子挑战赛：码力全开，算赢未来

>>