UEFI Basics Tutorial (V) - A Preliminary Study on PPI

原创 xiaopangzi313 Posted on 2019-04-27 16:00:53 Read 6.7k Collection 31 Likes 11

Category columns: 15_Firmware Development Article Tags: BIOS UEFI FIRMWARE

摘要 This article introduces the PPI (PEIM-to-PEIM Interface) communication mechanism in UEFI environment in detail, and shows how to use PeiServicesInstallPpi and PeiServicesLocatePpi to register and obtain services between PEIM modules and access resources through specific examples.

1. Write source code

1. Write C:\edkii\OvmfPkg\MyHelloWorldInstallPPI\MyHelloWorldInstallPPI.c,

C AI generated projects 登录复制 run

```c
#include <uefi.h>
#include <Library/UefiLib.h>
#include <Library/BaseLib.h>
#include <Library/DebugLib.h>
#include <Library/BaseMemoryLib.h>
#include <Library/UefiDriverEntryPoint.h>
#include <Library/PeimEntryPoint.h>
#include <Library/PeiServicesLib.h>
#include <Library/PeiServicesTablePointerLib.h>
#include <Pi/PiHob.h>
#include <Pi/PiPeiCis.h>

EFI_GUID gMyHelloWorldPEIGUID = { 0xbdb38129, 0x4d65, 0x39f4, { 0x72, 0x12, 0x68, 0xcf, 0x5a, 0x19, 0xa, 0xf8 }};

```

```
 15
 16 EFI_STATUS
 17 EFIAPI
 18 PrintHelloMsg (
 19   IN CHAR16 * Msg
 20   )
twen {
twen   DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] PrintHelloMsg : %s \n",Msg));
twen   return EFI_SUCCESS;
twen }
 25
 26 typedef EFI_STATUS (EFIAPI * PRINTMSG)(CHAR16 *Msg);
 27 typedef struct _EFI_HELLOWORLD_PROTOCOL  {
 28     PRINTMSG PrintMsg;
 29 }EFI_HELLOWORLD_PROTOCOL;
 30
 31 EFI_HELLOWORLD_PROTOCOL mHelloWorldPpi = {
 32     PrintHelloMsg
 33 };
 34
 35 EFI_PEI_PPI_DESCRIPTOR     mPpiListCodePpi = {
 36   (EFI_PEI_PPI_DESCRIPTOR_PPI | EFI_PEI_PPI_DESCRIPTOR_TERMINATE_LIST),
 37   &gMyHelloWorldPEIGUID,
 38   &mHelloWorldPpi
 39 };
 40
 41 EFI_STATUS
 42 EFIAPI
 43 MyHelloWorldInstallPPIEntry(
 44   IN       EFI_PEI_FILE_HANDLE  FileHandle,
 45   IN CONST EFI_PEI_SERVICES     **PeiServices
 46 )
 47 {
 48   EFI_STATUS                             Status = EFI_SUCCESS;
 49   DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] MyHelloWorldInstallPPIEntry Start..\n"));
 50   PeiServicesInstallPpi  (&mPpiListCodePpi);
 51   DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] MyHelloWorldInstallPPIEntry End..\n"));
 52
 53   return Status;
 }
```

收起 ^

2. Write C:\edkii\OvmfPkg\MyHelloWorldInstallPPI\MyHelloWorldInstallPPI.inf,

C AI generated projects 登录复制 run

```
[Defines]
  INF_VERSION = 0x00010006
  BASE_NAME = MyHelloWorldInstallPPI
  FILE_GUID = 69E6DE5D-F09E-485E-9936-EB70FDCFC82A
  MODULE_TYPE = PEIM
  VERSION_STRING = 1.0
  ENTRY_POINT = MyHelloWorldInstallPPIEntry

[Sources]
  MyHelloWorldInstallPPI.c

[Packages]
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec
  MdeModulePkg/MdeModulePkg.dec

[LibraryClasses]
  BaseLib
  PeimEntryPoint
  BaseMemoryLib
  DebugLib
  PeiServicesLib
  PrintLib
  #PeiLib
  #EfiCommonLib

[depex]
  TRUE
```

收起 ^

3. Write C:\edkii\OvmfPkg\MyHelloWorldLocatePPI\MyHelloWorldLocatePPI.c

```c
#include <uefi.h>
#include <Library/UefiLib.h>
#include <Library/BaseLib.h>
#include <Library/DebugLib.h>
#include <Library/BaseMemoryLib.h>
#include <Library/UefiDriverEntryPoint.h>
#include <Library/PeimEntryPoint.h>
#include <Library/PeiServicesLib.h>
#include <Library/PeiServicesTablePointerLib.h>
#include <Pi/PiHob.h>

EFI_GUID gMyHelloWorldPEIGUID = { 0xbdb38129, 0x4d65, 0x39f4, { 0x72, 0x12, 0x68, 0xcf, 0x5a, 0x19, 0xa, 0xf8 }};


typedef EFI_STATUS (EFIAPI * PRINTMSG)(CHAR16 *Msg);
typedef struct _EFI_HELLOWORLD_PROTOCOL  {
    PRINTMSG PrintMsg;
}EFI_HELLOWORLD_PROTOCOL;

//ShellCEntryLib call user interface ShellAppMain
EFI_STATUS
EFIAPI
MyHelloWorldLocatePPIEntry(
  IN       EFI_PEI_FILE_HANDLE  FileHandle,
  IN CONST EFI_PEI_SERVICES     **PeiServices
)
{
  EFI_STATUS  Status = EFI_SUCCESS;
  EFI_HELLOWORLD_PROTOCOL *mHelloWorldPpi = NULL;
  DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] MyHelloWorldLocatePPIEntry Locate PPI Start..\n"));
  Status = PeiServicesLocatePpi (
           &gMyHelloWorldPEIGUID,
           0,
           NULL,
           (VOID **)&mHelloWorldPpi
           );
  if (EFI_ERROR(Status)){
      DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] MyHelloWorldLocatePPIEntry Locate PPI Fail..%r\n",Status));
      return Status;
```

```
  }
  mHelloWorldPpi->PrintMsg(L"2019 CSDN Locate PPI Hello World ...\n");
  DEBUG ((EFI_D_ERROR, "[MyHelloWorldPPI] MyHelloWorldLocatePPIEntry Locate PPI End..\n"));
  return Status;
}
```

收起 ^

4. Write C:\edkii\OvmfPkg\MyHelloWorldLocatePPI\MyHelloWorldLocatePPI.inf

C AI generated projects 登录复制 run

```
[Defines]
  INF_VERSION = 0x00010006
  BASE_NAME = MyHelloWorldLocatePPI
  FILE_GUID = 69E6DE6D-F39F-485f-9037-EB70FDCFC82B
  MODULE_TYPE = PEIM
  VERSION_STRING = 1.0
  ENTRY_POINT = MyHelloWorldLocatePPIEntry

[Sources]
  MyHelloWorldLocatePPI.c

[Packages]
  MdePkg/MdePkg.dec
  ShellPkg/ShellPkg.dec
  MdeModulePkg/MdeModulePkg.dec

[LibraryClasses]
  BaseLib
  PeimEntryPoint
  BaseMemoryLib
  DebugLib
  PeiServicesLib
  PrintLib
  #PeiLib
  #EfiCommonLib

[depex]
  TRUE
```

收起 ^

5. Modify C:\edkii\OvmfPkg\OvmfPkgX64.dsc

```c
[Components]
...
#
# PEI Phase modules
#
INF  OvmfPkg/MyHelloWorldInstallPPI/MyHelloWorldInstallPPI.inf
INF  OvmfPkg/MyHelloWorldLocatePPI/MyHelloWorldLocatePPI.inf
...
```

6. Modify C:\edkii\OvmfPkg\OvmfPkgX64.fdf

```c
[FV.PEIFV]
...
#
# PEI Phase modules
#
INF  OvmfPkg/MyHelloWorldInstallPPI/MyHelloWorldInstallPPI.inf
INF  OvmfPkg/MyHelloWorldLocatePPI/MyHelloWorldLocatePPI.inf
...
```

2. Compile and generate EFI files

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

3. Run HelloWorld PPI Driver

1. Copy C:\edkii\Build\OvmfX64\DEBUG_VS2013x86\FV\OVMF.fd to C:\qemu

2. The output of executing is as follows, `setup-qemu-x64.bat | findstr MyHelloWorldPPI`

```
C:\qemu>setup-qemu-x64.bat | findstr MyHelloWorldPPI
WARNING: Image format was not specified for 'HDD_BOOT.img' and probing guessed raw.
         Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.

         Specify the 'raw' format explicitly to remove the restrictions.
[MyHelloWorldPPI] MyHelloWorldInstallPPIEntry Start..
[MyHelloWorldPPI] MyHelloWorldInstallPPIEntry End..
[MyHelloWorldPPI] MyHelloWorldLocatePPIEntry Locate PPI Start..
[MyHelloWorldPPI] PrintHelloMsg : 2019 CSDN Locate PPI Hello World ...
[MyHelloWorldPPI] MyHelloWorldLocatePPIEntry Locate PPI End..
```


IV. Summary

PPI (PEIM-to-PEIM Interface) is a communication method between PEIM drivers. In this article, `MyHelloWorldInstallPPI` the module `PeiServicesInstallPpi` provides the mPpiListCodePpi service, and `MyHelloWorldLocatePPI` the module `PeiServicesLocatePpi` obtains the mPpiListCodePpi object and accesses the member functions of the object to access the driver `MyHelloWorldInstallPPI` resources.

PPI DEMO source code