【UEFI Practice】BIOS Upgrade (1) -- Upgrade BIOS in shell

原创 Mutouyugi Posted on 2023-10-12 23:32:02 Read 1.8k Collection 6 Likes 1

Category Column: UEFI in Action Article Tags: server

Table of Contents

Preface

BIOS upgrades are divided into three scenarios: Shell, Setup, and OS.
This article will focus on upgrading under the shell. The specific operation is to obtain the BIOS file information that needs to be upgraded under the shell, and then pass the data to flash in SMM mode.
It is similar to the implementation method of capsule, but only uses the simplest method to update BIOS.

部分代码借鉴EDKII源码，如有侵权，请联系删除

---

1. Read data under Shell

1. Create a shell tool. For details, please refer to this article: [UEFI Practice] Create a shell tool with parameters under EDKII
2. Use EDKII to `EFI_SHELL_PROTOCOL` read BIOS data

```
UINTN               BIOSFileSize;
VOID                *BIOSFileData;
//1.LocateProtocol
gBS->LocateProtocol (&gEfiShellProtocolGuid,NULL,(VOID **)&mShellProtocol);
//2.根据FileName打开文件，注意此文件目录在shelltool运行的目录
ShellProtocol->OpenFileByName (FileName,&Handle,EFI_FILE_MODE_READ);
//3.读文件数据，需要的话可以通过GetFileSize读文件大小
ShellProtocol->ReadFile (Handle,&BIOSFileSize,BIOSFileData);
```

2. Data Transfer

在X86的环境下，是不能直接对flash进行擦写的，需要触发SMI中断进入SMM模式下操作。

1. Register for SMM service

`SMM_DRIVER或者DXE_SMM_DRIVER` Register in

```
gSmst->SmiHandlerRegister(UpdateBiosCallback,&GUID,&Handle );
```

After triggering, the UpdateBiosCallback function will be called

2. Trigger SMM mode and transfer data

There are two ways to trigger SMI interrupt to enter SMM mode:
1. Register a smi number, and then write this smi number to IO port 0xB2 when triggered. I will not elaborate on this.
2. `EFI_SMM_COMMUNICATION_PROTOCOL` Pass data through the method. This method is used here.

```
1.定义需要传递的结构体
typedef struct {
  EFI_GUID    HeaderGuid;
  UINTN       Action;
  UINTN       BIOSDataSize;
  UINT8       BIOSData;
} BIOS_INFO;

```

```
2.赋值
BIOS_INFO *buffer;
buffer -> HeaderGuid = GUID;
buffer -> Action = 0x1;
buffer -> BIOSDataSize = BIOSFileSize;
buffer -> BIOSData = (UINT8*)BIOSFileData;
UINTN buffersize = sizeof(buffer);

3.调用EFI_SMM_COMMUNICATION_PROTOCOL
(1)gBS->LocateProtocol (&gEfiSmmCommunicationProtocolGuid, NULL, (VOID **)&SmmCommunication);
(2)SmmCommunication->Communicate (SmmCommunication, buffer, &buffersize);
```

In this way, the BIOS file data is passed into SMM mode under the shell, and the trigger is found through HeaderGuid

3. Flash FLASH

```
EFI_STATUS
UpdateBiosCallback(
IN EFI_HANDLE  DispatchHandle,
IN CONST VOID  *Context         OPTIONAL,
IN OUT VOID    *CommBuffer      OPTIONAL,
IN OUT UINTN   *CommBufferSize  OPTIONAL
)
{
    //CommBuffer即传进来的数据
    //可以通过Action再次判断
    if(CommBuffer == 0x1)
    {
       //更新bios
    }
}
```

Update the BIOS using PCH_SPI_PROTOCOL

```
gSmst->SmmLocateProtocol (&gPchSmmSpiProtocolGuid, NULL, (VOID **) &SmmSpiProtocol);
```

Can only be called in SMM mode

```
1.使能
需要对flash读写位使能，具体操作哪一位以平台为准
2.擦flash，只能以4K或者4K倍数为单位擦除，不先擦除再写会出问题，具体看flash芯片类型
SmmSpiProtocol->FlashErase();
3.写数据
SmmSpiProtocol->FlashWrite();

伪代码:
{
    WriteEnable();
    for(index = 0; ......... ;  index += 0x1000)
    {
        SmmSpiProtocol->FlashErase();
        SmmSpiProtocol->FlashWrite(BIOSFileData+index );
    }

}
```

Summarize

I will continue to add to my understanding later. I am not very skilled, please point out any mistakes;
if there is any infringement, please contact us to delete it.