UEFI Basic Tutorial (Seventeen) - Simple Use of SMM

Category columns: 15_Firmware Development

1. Write SMM driver to register SMM service

1. Use SwDispatch Demo1

```
EFI_STATUS
EFIAPI
CsdnSmiHandler (
  IN EFI_HANDLE DispatchHandle,
  IN CONST VOID *Context         OPTIONAL,
  IN OUT VOID   *CommBuffer      OPTIONAL,
  IN OUT UINTN  *CommBufferSize OPTIONAL
  )
{
   DEBUG((EFI_D_ERROR, "[CSDN] %a Enter\n", __FUNCTION__));
  // 实现SMM服务
   DEBUG((EFI_D_ERROR,"[CSDN] %a Exit\n", __FUNCTION__));
   return EFI_SUCCESS;
}


Status = gSmst->SmmLocateProtocol(
    &gEfiSmmSwDispatch2ProtocolGuid,
    NULL,
    (VOID **)&SwDispatch
    );

SwContext.SwSmiInputValue = CSDN_SW_SMI_NO; //自定义SMI Number
ASSERT_EFI_ERROR(SwContext.SwSmiInputValue < SwDispatch->MaximumSwiValue);

Status = SwDispatch->Register(
      SwDispatch,
      CsdnSmiHandler,
      &SwContext,
      &SwSmiHandle
    );

```

收起 ∧

2. Use gSmst->SmiHandlerRegister Demo2

```
EFI_STATUS
EFIAPI
SmmCsdnSmiHandler (
  IN EFI_HANDLE DispatchHandle,
  IN CONST VOID *Context         OPTIONAL,
  IN OUT VOID   *CommBuffer      OPTIONAL,
  IN OUT UINTN  *CommBufferSize OPTIONAL
  )
{
   DEBUG((EFI_D_ERROR, "[CSDN] %a Enter\n", __FUNCTION__));
    //可以将SMM RAM 内容copy回 communication buffer
   DEBUG((EFI_D_ERROR,"[CSDN] %a Exit\n", __FUNCTION__));
   return EFI_SUCCESS;
}

```

```
16    Status = gSmst->SmiHandlerRegister(
17        SmmCsdnSmiHandler,
18        &gCsdnSmmGuid,
19        &DispatchHandle
20    );
twen
```

收起 ∧

2. Triggering SMM service

1. Use **IO** Demo3

```
1    IoWrite8 (0xB2, CSDN_SW_SMI_NO);
```

2. Use **SmmCommunication** to access **SMM** Demo4

```
1    Status = SmmCommunication->Communicate(SmmCommunication, CommBuffer, &CommSize);
```