UEFI Basic Tutorial (XI) - Reading SMBIOS information under Shell

原创 xiaopangzi313

Posted on 2020-06-07 00:28:07 Read 9k Collection 37 Likes 7

Category columns: 15_Firmware Development

摘要 This article describes how to write an application to read SMBIOS information in a UEFI environment, including BIOS, system information, etc. By locating the SMBIOS protocol and traversing the SMBIOS table, different types of data, such as system manufacturer, product name, etc., can be parsed.

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

1. Write source code

1. Write the UEFI Application
   code C:\edkii\OvmfPkg\MyHelloWorldSMBios\MyHelloWorldSMBios.c,

AI generated projects 登录复制

```
EFI_STATUS
EFIAPI
MyHelloWorldSMBiosEntry(
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
)
{
  DUMP_LOG ("MyHelloWorldSmbiosAppEntry Start\n")
  Print (L"[MyHelloWorldSmbios]  MyHelloWorldSmbiosAppEntry Start..\n");
  //
  // 1. 找到 SMBIOS protocol
  //
  Status = gBS->LocateProtocol (
                  &gEfiSmbiosProtocolGuid,
                  NULL,
                  (VOID**)&Smbios
                  );
  if (EFI_ERROR (Status)) {
    return Status;
  }

  SmbiosHandle = SMBIOS_HANDLE_PI_RESERVED;
  // 2.配合下面while循环，轮询所有SMBIOS Table
  Status = Smbios->GetNext (Smbios, &SmbiosHandle, NULL, &Record, NULL);
  while (!EFI_ERROR(Status)) {
      DUMP_LOG ("SMBIOS Type %d ..\n", Record->Type);
      Print (L"[MyHelloWorldSmbios] SMBIOS Type %d ..\n", Record->Type);

    //SMBIOS_TYPE_BIOS_INFORMATION 0
    if (Record->Type == SMBIOS_TYPE_BIOS_INFORMATION) {
       Type0Record = (SMBIOS_TABLE_TYPE0 *) Record;

       ....
    //SMBIOS_TYPE_SYSTEM_INFORMATION 1
    // 3. 指定SMBIOS type过滤 SMBIOS table
    }else if (Record->Type == SMBIOS_TYPE_SYSTEM_INFORMATION){
       Type1Record = (SMBIOS_TABLE_TYPE1 *) Record;
       DUMP_LOG ("Manufacturer |ProductName |Version |SerialNumber |WakeUpType |SKUNumber |Family\n")
       DUMP_LOG ("%04x         |%04x        |%04x    |%04x         |%04x       |%04x      |%04x  \n",
            Type1Record->Manufacturer ,
            Type1Record->ProductName ,
            Type1Record->Version ,
            Type1Record->SerialNumber ,
            Type1Record->WakeUpType ,
            Type1Record->SKUNumber ,
            Type1Record->Family
        )
    //SMBIOS_TYPE_SYSTEM_ENCLOSURE 3
    }else if (Record->Type == SMBIOS_TYPE_SYSTEM_ENCLOSURE)
      ......
    //SMBIOS_TYPE_PHYSICAL_MEMORY_ARRAY 16
    }else if (Record->Type == SMBIOS_TYPE_PHYSICAL_MEMORY_ARRAY){
      ......
    //SMBIOS_TYPE_MEMORY_DEVICE 17
    }else if (Record->Type == SMBIOS_TYPE_MEMORY_DEVICE){
      ......
    //SMBIOS_TYPE_MEMORY_ARRAY_MAPPED_ADDRESS 19
    }else if (Record->Type == SMBIOS_TYPE_MEMORY_ARRAY_MAPPED_ADDRESS)  {
      ......
    //SMBIOS_TYPE_SYSTEM_BOOT_INFORMATION 32
    }else if (Record->Type == SMBIOS_TYPE_SYSTEM_BOOT_INFORMATION){
      ......
    }
    Status = Smbios->GetNext (Smbios, &SmbiosHandle, NULL, &Record, NULL);
  }

  DEBUG ((EFI_D_ERROR, "[MyHelloWorldSmbios] MyHelloWorldSmbiosAppEntry End..\n"));

}
```

收起 ^

2. Modify C:\code\local_edkrepo_10nm\Edk2\OvmfPkg\MyHelloWorldSMBios\MyHelloWorldSMBios.inf

AI generated projects 登录复制

```
[Defines]
  INF_VERSION = 0x00010006
  BASE_NAME = MyHelloWorldSmbios
  FILE_GUID = 69A69543-FA9F-485E-9A3E-EA70FDCFC82F
  MODULE_TYPE = UEFI_APPLICATION
  VERSION_STRING = 1.0
  ENTRY_POINT = MyHelloWorldSmbiosAppEntry

[Sources]
  MyHelloWorldSmbios.c
  ....

[Protocols]
  gEfiSmbiosProtocolGuid                      # PROTOCOL ALWAYS_CONSUMED

[Depex]
  gEfiSmbiosProtocolGuid

```

收起 ^

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

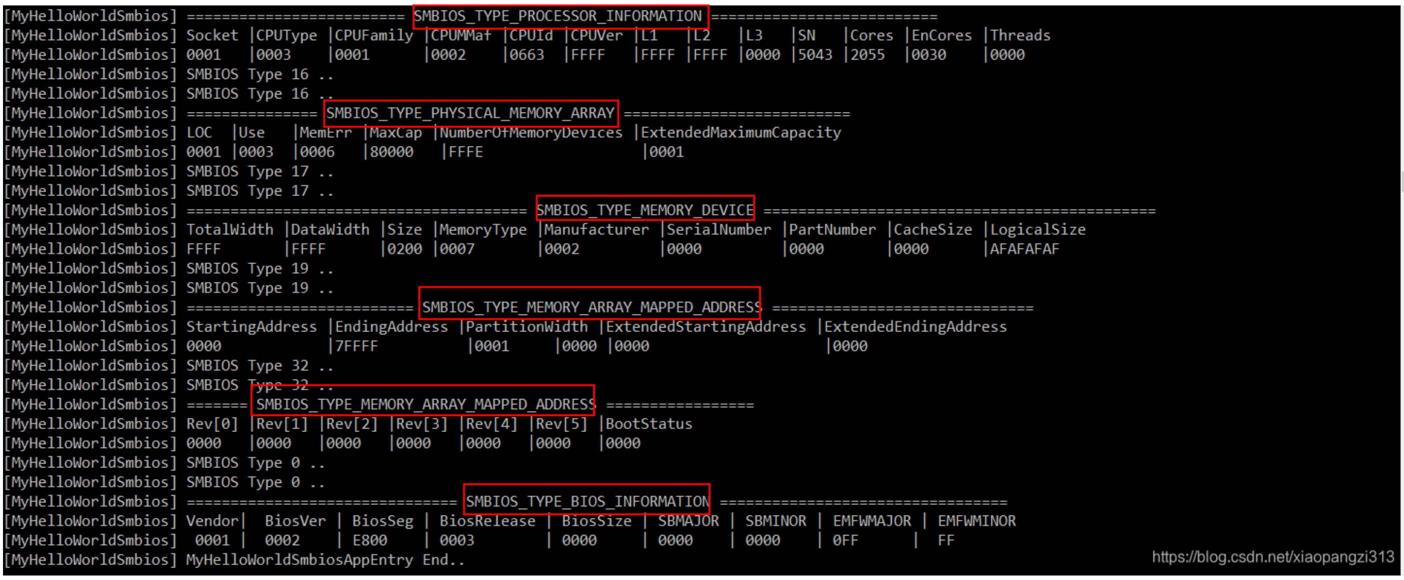
AI generated projects 登录复制

```
...
#
[Components]
  OvmfPkg/MyHelloWorldSmbios/MyHelloWorldSmbios.inf
...
```

2. Compile and generate EFI files

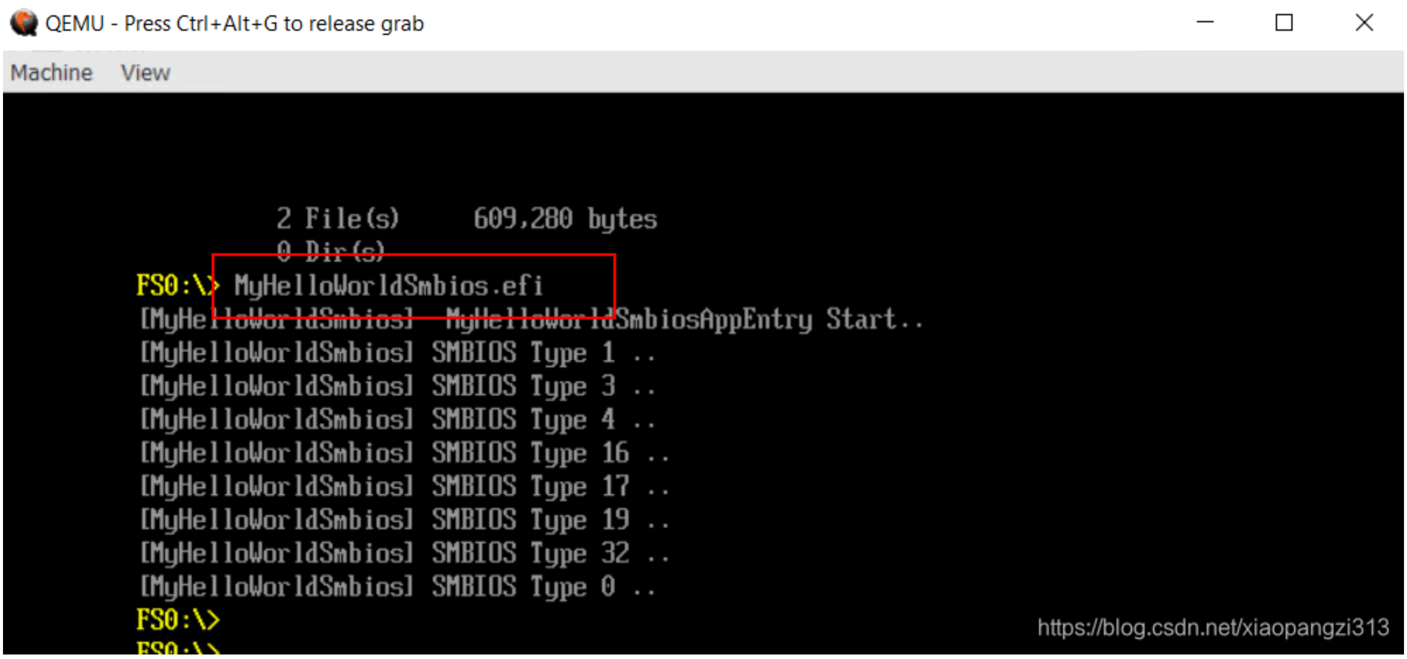
Run `edksetup.bat` && `build -a X64 -p OvmfPkg\OvmfPkgX64.dsc -D DEBUG_ON_SERIAL_PORT -t VS2013x86` compile the entire OvmfPkg Package

3. Run UEFI APP `MyHelloWorldSmbios.efi`



```
[MyHelloWorldSmbios] ========================= SMBIOS_TYPE_PROCESSOR_INFORMATION =========================
[MyHelloWorldSmbios] Socket |CPUType |CPUFamily |CPUMMaf |CPUId |CPUVer |L1   |L2   |L3   |SN   |Cores |EnCores |Threads
[MyHelloWorldSmbios] 0001   |0003    |0001      |0002    |0663  |FFFF   |FFFF |FFFF |0000 |5043 |2055  |0030    |0000
[MyHelloWorldSmbios] SMBIOS Type 16 ..
[MyHelloWorldSmbios] SMBIOS Type 16 ..
[MyHelloWorldSmbios] =============== SMBIOS_TYPE_PHYSICAL_MEMORY_ARRAY =========================
[MyHelloWorldSmbios] LOC  |Use   |MemErr |MaxCap |NumberOfMemoryDevices |ExtendedMaximumCapacity
[MyHelloWorldSmbios] 0001 |0003  |0006   |80000  |FFFE                   |0001
[MyHelloWorldSmbios] SMBIOS Type 17 ..
[MyHelloWorldSmbios] SMBIOS Type 17 ..
[MyHelloWorldSmbios] ======================================= SMBIOS_TYPE_MEMORY_DEVICE ============================================
[MyHelloWorldSmbios] TotalWidth |DataWidth |Size |MemoryType |Manufacturer |SerialNumber |PartNumber |CacheSize |LogicalSize
[MyHelloWorldSmbios] FFFF       |FFFF      |0200 |0007       |0002         |0000         |0000       |0000      |AFAFAFAF
[MyHelloWorldSmbios] SMBIOS Type 19 ..
[MyHelloWorldSmbios] SMBIOS Type 19 ..
[MyHelloWorldSmbios] ========================== SMBIOS_TYPE_MEMORY_ARRAY_MAPPED_ADDRESS ==============================
[MyHelloWorldSmbios] StartingAddress |EndingAddress |PartitionWidth |ExtendedStartingAddress |ExtendedEndingAddress
[MyHelloWorldSmbios] 0000            |7FFFF         |0001      |0000 |0000                    |0000
[MyHelloWorldSmbios] SMBIOS Type 32 ..
[MyHelloWorldSmbios] SMBIOS Type 32 ..
[MyHelloWorldSmbios] ======= SMBIOS_TYPE_MEMORY_ARRAY_MAPPED_ADDRESS ================
[MyHelloWorldSmbios] Rev[0] |Rev[1] |Rev[2] |Rev[3] |Rev[4] |Rev[5] |BootStatus
[MyHelloWorldSmbios] 0000   |0000   |0000   |0000   |0000   |0000   |0000
[MyHelloWorldSmbios] SMBIOS Type 0 ..
[MyHelloWorldSmbios] SMBIOS Type 0 ..
[MyHelloWorldSmbios] =============================== SMBIOS_TYPE_BIOS_INFORMATION ================================
[MyHelloWorldSmbios] Vendor|  BiosVer | BiosSeg | BiosRelease | BiosSize | SBMAJOR | SBMINOR | EMFWMAJOR | EMFWMINOR
[MyHelloWorldSmbios]  0001 |  0002    | E800    | 0003        | 0000     | 0000    | 0000    | 0FF       |  FF
[MyHelloWorldSmbios] MyHelloWorldSmbiosAppEntry End..
```

https://blog.csdn.net/xiaopangzi313



```
          2 File(s)     609,280 bytes
          0 Dir(s)
FS0:\> MyHelloWorldSmbios.efi
[MyHelloWorldSmbios] MyHelloWorldSmbiosAppEntry Start..
[MyHelloWorldSmbios] SMBIOS Type 1 ..
[MyHelloWorldSmbios] SMBIOS Type 3 ..
[MyHelloWorldSmbios] SMBIOS Type 4 ..
[MyHelloWorldSmbios] SMBIOS Type 16 ..
[MyHelloWorldSmbios] SMBIOS Type 17 ..
[MyHelloWorldSmbios] SMBIOS Type 19 ..
[MyHelloWorldSmbios] SMBIOS Type 32 ..
[MyHelloWorldSmbios] SMBIOS Type 0 ..
FS0:\>
FS0:\>
```

https://blog.csdn.net/xiaopangzi313

IV. Summary

After the motherboard is powered on, UEFI will store HW information including CPU, Memory, FW, PM and other information in a memory area in the form of SMBIOS table. After entering the OS, the OS can obtain relevant configurations by parsing the memory.

In UEFI Shell, the entire SMBIOS table can be accessed through SmbiosProtocol. This article provides a demo implementation.

Smbios DEMO source code

about Us Careers Business Cooperation Seeking coverage ☎ 400-660-0108 ✉ kefu@csdn.net Online Customer Service Working hours 8:30-22:00

Public Security Registration Number 11010502030143 Beijing ICP No. 19004658 Beijing Internet Publishing House [2020] No. 1039-165

Commercial website registration information Beijing Internet Illegal and Harmful Information Reporting Center Parental Control

Online 110 Alarm Service China Internet Reporting Center Chrome Store Download Account Management Specifications

Copyright and Disclaimer Copyright Complaints Publication License Business license