

BIOS Practice: Adjusting the Boot Order

原创

Anthony

Modified on 2022-02-06 20:46:26

Read 3.7k


Collection 9

Likes

Category Column: BIOS learning practice

Article Tags: UEFI C

Copyright CC 4.0 BY-SA



BIOS learning practice This column includes this content

21 articles

Subscribe to our column

摘要

This article introduces a method to adjust the boot order in BIOS, including how to set a specific option such as UEFI shell as the last boot item, and provides a specific code example to implement this function.

The summary is generated in [C Know](#) , supported by DeepSeek-R1 full version, [go to experience](#)>

In the process of improving the BIOS functions, we often need to adjust the boot order. For example, if there is a UEFI shell boot item, we need to put it at the end, or put a hard disk first.

AI generated projects 登录复制

```
EfiBootManagerSortLoadOptionVariable (LoadOptionTypeBoot, (SORT_COMPARE)CompareBootOption);
```

AI generated projects 登录复制

```
1
2 INTN
3 EFIAPI
4 CompareBootOption (
5     CONST EFI_BOOT_MANAGER_LOAD_OPTION *Left,
6     CONST EFI_BOOT_MANAGER_LOAD_OPTION *Right
7 )
8 {
9     return BootOptionPriority (Left) - BootOptionPriority (Right);
10 }
```

收起 ^

AI generated projects 登录复制

```
1 UINTN
2 BootOptionPriority (
3     CONST EFI_BOOT_MANAGER_LOAD_OPTION *BootOption
4 )
5 {
6     EFI_DEVICE_PATH_PROTOCOL *DevPath;
7     DevPath = BootOption->FilePath;
8     while(!IsDevicePathEndType((void*)DevPath))
9     {
10         if ( (DevPath->Type == MESSAGING_DEVICE_PATH) && (DevPath->SubType == MSG_SATA_DP) ){
11             return 10;
12         }else if( (DevPath->Type == MESSAGING_DEVICE_PATH) && (DevPath->SubType == MSG_USB_DP) ){
13             return 20;
14         }else if((DevPath->Type == MESSAGING_DEVICE_PATH) && ((DevPath->SubType == MSG_IPv4_DP) || (DevPath->SubType == MSG_IPv6_DP))){
15             return 30;
16         }
17         DevPath = NextDevicePathNode((void*)DevPath);
18     }
19     //shell
20     if (StrCmp (BootOption->Description, L"UEFI Shell") == 0) {
21         return 40;
22     }
23
24     return 1000;
25 }
```

收起 ^

Adding the above code to the startup sequence adjustment code segment can adjust the default startup order very well, but how to make the shell or an option at the end, you need to use the following code:

```
1 VOID
2 EFIAPI
```

```

3 | EfiBootManagerSetShellAsLast ()
4 | {
5 |     EFI_STATUS Status;
6 |     EFI_BOOT_MANAGER_LOAD_OPTION *LoadOption;
7 |     UINTN LoadOptionCount;
8 |     UINTN Index, BootOrderIndex;
9 |     UINT16 *OptionOrder;
10
11 |     LoadOption = EfiBootManagerGetLoadOptions (&LoadOptionCount, LoadOptionTypeBoot);
12 |     //
13 |     // Create new ***Order variable
14 |     //
15 |     OptionOrder = AllocatePool (LoadOptionCount * sizeof (UINT16));
16 |     ASSERT (OptionOrder != NULL);
17 |     BootOrderIndex = 0;
18 |     for (Index = 0; Index < LoadOptionCount; Index++) {
19 |         if (StrCmp(LoadOption[Index].Description, L"UEFI Shell") != 0)
20 |         {
21 |             OptionOrder[BootOrderIndex] = (UINT16) LoadOption[Index].OptionNumber;
22 |             BootOrderIndex ++;
23 |         }else
24 |         {
25 |             OptionOrder[LoadOptionCount-1] = (UINT16) LoadOption[Index].OptionNumber;
26 |         }
27 |     }
28 |     Status = gRT->SetVariable (
29 |         mBmLoadOptionOrderName[LoadOptionTypeBoot],
30 |         &gEfiGlobalVariableGuid,
31 |         EFI_VARIABLE_BOOTSERVICE_ACCESS | EFI_VARIABLE_RUNTIME_ACCESS | EFI_VARIABLE_NON_VOLATILE,
32 |         LoadOptionCount * sizeof (UINT16),
33 |         OptionOrder
34 |     );
35 |     //
36 |     // Changing the *Order content without increasing its size with current variable implementation shouldn't fail.
37 |     //
38 |     ASSERT_EFI_ERROR (Status);
39 |
40 |     FreePool (OptionOrder);
41 |     EfiBootManagerFreeLoadOptions (LoadOption, LoadOptionCount);
42 | }

```

收起 ^

Through the above two, we can adjust the startup order very well. Of course, if there are special startup order requirements, they can also be written according to the above logic.