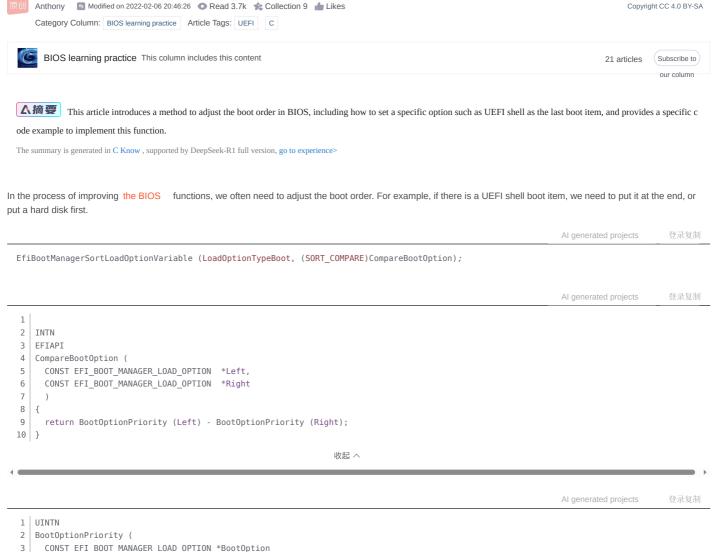
BIOS Practice: Adjusting the Boot Order



```
CONST EFI_BOOT_MANAGER_LOAD_OPTION *BootOption
4
5
6
     EFI_DEVICE_PATH_PROTOCOL *DevPath;
     DevPath = BootOption->FilePath;
     while (!IsDevice Path End Type ((void*) Dev Path))\\
8
9
         \  \  \text{if ( DevPath->Type == MSSAGING\_DEVICE\_PATH) \&\& (DevPath->SubType == MSG\_SATA\_DP) )} \\ 
10
11
12
        }else if( (DevPath->Type == MESSAGING DEVICE PATH) && (DevPath->SubType == MSG USB DP) ){
13
        }else if((DevPath->Type == MESSAGING_DEVICE_PATH) && ((DevPath->SubType == MSG_IPv4_DP) || (DevPath->SubType == MSG_IPv6_DP))){
14
15
         return 30:
16
17
       DevPath = NextDevicePathNode((void*)DevPath):
18
19
     if (StrCmp (BootOption->Description, L"UEFI Shell") == 0) {
20
21
        return 40;
22
23
24
      return 1000;
25 | }
                                                                    收起 へ
```

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

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.

收起 へ

about Us Careers Business Seeking 2400-660- 108 kefu@csdn.net Customer 8:30-22:00 Service Public Security Registration Number 11010502030143 Beijing ICP No. 19004658 Beijing 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 ©1999-2025 Beijing Innovation Lezhi Network Technology Co., Ltd.