

BIOS practice: reading files---getting file path + final reading

原创

Anthony  Posted on 2022-01-11 14:42:30  Read 2.9k  Collection 7  Likes 3

Copyright CC 4.0 BY-SA

Category Column: BIOS learning practice Article Tags: UEFI C



BIOS learning practice This column includes this content

21 articles

Subscribe to

our column



This article describes how to obtain the full path of an application in an EFI environment and implement the function of extracting the folder path from the path. By using EFI_LOADED_IMAGE_PROTOCOL and traversing DevicePathProtocol, you can accurately locate the folder where the application is located.

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

Continuing from the previous section, getting the file path is actually getting the path of the current APP. How to get the path of this APP? Here we need to use an EFI_LOADED_IMAGE_PROTOCOL. The principle is as follows:

- Get the corresponding Handle according to the ImageHandle of the current Application.
- Use HandleProtocol to directly open the EFI_LOADED_IMAGE_PROTOCOL loaded on the Application.
- There is FilePath on this Protocol, which is the Device Path Protocol of the current Image.

See the following code for a clear explanation:

AI generated projects

登录复制

```
1 EFI_LOADED_IMAGE_PROTOCOL *LoadedImage;
2
3 Status = gBS->HandleProtocol(
4     gImageHandle,
5     &gEfiLoadedImageProtocolGuid,
6     (void **)&LoadedImage
7 );
8 if(EFI_ERROR(Status)) {
9     Print(L"Application path error!\n");
10    return Status;
11 }
12
13 EFI_DEVICE_PATH_PROTOCOL *ptFilePath;
14
15 ptFilePath = LoadedImage->FilePath;
16
17 if(!ptFilePath) {
18     Print(L"Unkown media!\n");
19     return Status;
20 }
```

收起 ^

OK, it's time to go to the Device Path Protocol. In the device path, find the device where we placed the APP:

AI generated projects

登录复制

```
1 while(!IsDevicePathEnd(ptFilePath)){
2     if(DevicePathType(ptFilePath)==MEDIA_DEVICE_PATH && DevicePathSubType(ptFilePath)==MEDIA_FILEPATH_DP){
3         break;
4     }
5     ptFilePath = NextDevicePathNode(ptFilePath);
6 }
7 TmpPath = ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName;
```

Then this TmpPath is the path of the app, such as \xxx\xxx\xxx\xx.efi , but we only need the path, not the name of the app, so we can remove the app name by judging '\', and the final function is as follows:

```
1 EFI_STATUS
2 GetAPPFolderPathStr(CHAR16 **Path)
3 {
4     EFI_STATUS Status;
5     EFI_DEVICE_PATH_PROTOCOL *ptFilePath;
```

```

6 | CHAR16 *PathName, *TmpPath; 7 | UINTN StrSize;
8 |
9 | Status = EFI_SUCCESS;
10 | PathName = NULL;
11 |
12 | EFI_LOADED_IMAGE_PROTOCOL *LoadedImage;
13 |
14 | Status = gBS->HandleProtocol(
15 |     gImageHandle,
16 |     &gEfiLoadedImageProtocolGuid,
17 |     (void **)&LoadedImage
18 | );
19 | if(EFI_ERROR(Status)) {
20 |     Print(L"Application path error!\n");
21 |     return Status;
22 | }
23 | ptFilePath = LoadedImage->FilePath;
24 |
25 | if(!ptFilePath) {
26 |     Print(L"Unkown media!\n");
27 |     return Status;
28 | }
29 |
30 | while(!IsDevicePathEnd(ptFilePath)){
31 |     if(DevicePathType(ptFilePath)==MEDIA_DEVICE_PATH && DevicePathSubType(ptFilePath)==MEDIA_FILEPATH_DP){
32 |         break;
33 |     }
34 |     ptFilePath = NextDevicePathNode(ptFilePath);
35 | }
36 | TmpPath = ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName;
37 |
38 | while ((TmpPath = StrStr (TmpPath+1, L"\\")) != NULL) {
39 |     PathName = TmpPath;
40 | }
41 |
42 | if(PathName != NULL){
43 |     StrSize = (PathName - ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName)*sizeof(CHAR16);
44 | }else{
45 |     StrSize = 2;
46 | }
47 |
48 | PathName = (CHAR16 *)AllocateZeroPool(StrSize + 2);
49 | if(PathName==NULL) {
50 |     Status = EFI_OUT_OF_RESOURCES;
51 |     goto ProcExit;
52 | }
53 |
54 | TmpPath = ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName;
55 | CopyMem (PathName, TmpPath, StrSize);
56 |
57 | if(PathName[0] != L'\\'){
58 |     Status = EFI_UNSUPPORTED;
59 |     goto ProcExit;
60 | }
61 | *Path = PathName;
62 |
63 | ProcExit:
64 | if(EFI_ERROR(Status) && PathName != NULL){
65 |     FreePool(PathName);
66 |     PathName = NULL;
67 | }
68 | return Status;
69 | }

```

收起 ^

After getting this path, we also need to add the name of the file, which is what we ultimately need:

```

1 | EFI_STATUS
2 | GetFullPathBaseOnAPPFolder (
3 |     IN CHAR16 *FileName,
4 |     IN OUT CHAR16 **FullPath
5 | )
6 | {
7 |     EFI_STATUS Status;
8 |     CHAR16 *AppFolder;
9 |     UINTN FullPathSize;

```

```

10 | CHAR16      *AppFullPath; 11 | CHAR16      Slash = L'\\';
12 |
13 | Status      = EFI_SUCCESS;
14 | AppFolder   = NULL;
15 | AppFullPath = NULL;
16 |
17 | if(FileName == NULL || FullPath == NULL){
18 |     Status = EFI_INVALID_PARAMETER;
19 |     goto ProcExit;
20 | }
21 |
22 | GetAPPFolderPathStr(&AppFolder); // 上个函数的返回路径
23 | if (AppFolder == NULL) {
24 |     Status = EFI_OUT_OF_RESOURCES;
25 |     goto ProcExit;
26 | }
27 | FullPathSize = strlen(AppFolder) + strlen(FileName);
28 | AppFullPath = AllocateZeroPool(FullPathSize);
29 | if(AppFullPath == NULL){
30 |     Status = EFI_OUT_OF_RESOURCES;
31 |     goto ProcExit;
32 | }
33 | if (strlen(AppFolder) > 4) { // '\\' and '\0'
34 |     CopyMem (AppFullPath, AppFolder, strlen(AppFolder) - 2); // skip '\0'
35 |     CopyMem (AppFullPath + strlen(AppFolder)/2 - 1, &Slash, 2);
36 |     CopyMem (AppFullPath + strlen(AppFolder)/2, FileName, strlen(FileName));
37 | } else {
38 |     CopyMem (AppFullPath, FileName, strlen(FileName));
39 | }
40 | *FullPath = AppFullPath;

```

收起 ^

The content of this AppFullPath is xxx\xxx\ **xxx.bmp**

For the final read function , we can chain the above functions together:

AI generated projects

登录复制

```

1 | EFI_STATUS
2 | ReadFile (
3 |     IN     CHAR16  *FileName,
4 |     IN OUT UINT8   **FileBuffer,
5 |     OUT    UINTN   *FileSize
6 | )
7 | {
8 |     EFI_STATUS Status;
9 |     CHAR16     *FullPath;
10 |
11 | // get File full path
12 | Status = GetFullPathBaseOnAPPFolder(FileName, &FullPath);
13 | if(EFI_ERROR(Status)) {
14 |     return Status;
15 | }
16 |
17 | Status = ReadFileInFS(FullPath, FileBuffer, FileSize);
18 | FreePool (FullPath);
19 |
20 | return Status;
21 | }

```

收起 ^

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 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

©1999-2025 Beijing Innovation Lezhi Network Technology Co., Ltd.