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

Category Column: BIOS learning practice Article Tags: UEFI C

摘要 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 HandleProtcol 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:

```
EFI_LOADED_IMAGE_PROTOCOL *LoadedImage;

  Status = gBS->HandleProtocol(
                  gImageHandle,
                  &gEfiLoadedImageProtocolGuid,
                  (void **)&LoadedImage
                  );
  if(EFI_ERROR(Status)) {
    Print(L"Application path error!\n");
    return Status;
  }

  EFI_DEVICE_PATH_PROTOCOL  *ptFilePath;

  ptFilePath  = LoadedImage->FilePath;

  if(!ptFilePath) {
    Print(L"Unkown media!\n");
    return Status;
  }
```

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

```
while(!IsDevicePathEnd(ptFilePath)){
    if(DevicePathType(ptFilePath)==MEDIA_DEVICE_PATH && DevicePathSubType(ptFilePath)==MEDIA_FILEPATH_DP){
      break;
    }
    ptFilePath = NextDevicePathNode(ptFilePath);
  }
  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:

```
EFI_STATUS
GetAPPFolderPathStr(CHAR16 **Path)
{
  EFI_STATUS                Status;
  EFI_DEVICE_PATH_PROTOCOL  *ptFilePath;
  CHAR16                    *PathName, *TmpPath;
  UINTN                     StrSize;

  Status = EFI_SUCCESS;
  PathName    = NULL;

  EFI_LOADED_IMAGE_PROTOCOL *LoadedImage;

  Status = gBS->HandleProtocol(
                  gImageHandle,
                  &gEfiLoadedImageProtocolGuid,
                  (void **)&LoadedImage
                  );
  if(EFI_ERROR(Status)) {
    Print(L"Application path error!\n");
    return Status;
  }
  ptFilePath  = LoadedImage->FilePath;

  if(!ptFilePath) {
    Print(L"Unkown media!\n");
    return Status;
  }

  while(!IsDevicePathEnd(ptFilePath)){
    if(DevicePathType(ptFilePath)==MEDIA_DEVICE_PATH && DevicePathSubType(ptFilePath)==MEDIA_FILEPATH_DP){
      break;
    }
    ptFilePath = NextDevicePathNode(ptFilePath);
  }
  TmpPath = ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName;

  while ((TmpPath = StrStr (TmpPath+1, L"\\")) != NULL) {
    PathName = TmpPath;
  }

  if(PathName != NULL){
    StrSize = (PathName - ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName)*sizeof(CHAR16);
  }else{
    StrSize = 2;
  }

  PathName = (CHAR16 *)AllocateZeroPool(StrSize + 2);
  if(PathName==NULL) {
    Status = EFI_OUT_OF_RESOURCES;
    goto ProcExit;
  }

  TmpPath = ((FILEPATH_DEVICE_PATH*)ptFilePath)->PathName;
  CopyMem (PathName, TmpPath, StrSize);

  if(PathName[0] != L'\\'){
    Status = EFI_UNSUPPORTED;
    goto ProcExit;
  }
  *Path = PathName;

ProcExit:
  if(EFI_ERROR(Status) && PathName != NULL){
    FreePool(PathName);
    PathName = NULL;
  }
  return Status;
}
```

收起 ^

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

```
EFI_STATUS
GetFullPathBaseOnAPPFolder (
  IN       CHAR16 *FileName,
  IN OUT   CHAR16 **FullPath
  )
{
  EFI_STATUS  Status;
  CHAR16      *AppFolder;
  UINTN       FullPathSize;
  CHAR16      *AppFullPath;
  CHAR16      Slash = L'\\';

  Status      = EFI_SUCCESS;
  AppFolder   = NULL;
  AppFullPath = NULL;

  if(FileName == NULL || FullPath == NULL){
    Status = EFI_INVALID_PARAMETER;
    goto ProcExit;
  }

  GetAPPFolderPathStr(&AppFolder);//上个函数的返回路径
  if (AppFolder == NULL) {
    Status = EFI_OUT_OF_RESOURCES;
    goto ProcExit;
  }
  FullPathSize = strsize(AppFolder) + strsize(FileName);
  AppFullPath  = AllocateZeroPool(FullPathSize);
  if(AppFullPath == NULL){
    Status = EFI_OUT_OF_RESOURCES;
    goto ProcExit;
  }
  if (strsize(AppFolder) > 4) { // '\' and '\0'
    CopyMem (AppFullPath, AppFolder, strsize(AppFolder) - 2); // skip '\0'
    CopyMem (AppFullPath + strsize(AppFolder)/2 - 1, &Slash, 2);
    CopyMem (AppFullPath + strsize(AppFolder)/2, FileName, strsize(FileName));
  } else {
    CopyMem (AppFullPath, FileName, strsize(FileName));
  }
  *FullPath = AppFullPath;
```

收起 ^

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

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

AI generated projects 登录复制

```
EFI_STATUS
ReadFile (
  IN     CHAR16   *FileName,
  IN OUT UINT8    **FileBuffer,
  OUT    UINTN    *FileSize
)
{
  EFI_STATUS  Status;
  CHAR16      *FullPath;

  // get File full path
  Status = GetFullPathBaseOnAPPFolder(FileName, &FullPath);
  if(EFI_ERROR(Status)) {
    return Status;
  }

  Status = ReadFileInFS(FullPath, FileBuffer, FileSize);
  FreePool (FullPath);

  return Status;
}
```

收起 ^