UEFI Principles and Programming Practice--File Operation



This article introduces how to use EFI_SIMPLE_FILE_SYSTEM_PROTOCOL to operate the FAT file system in the UEFI environment, including the basic steps of how to obtain the root directory handle, create and read files. It focuses on showing the practical application examples of the Write and Read functions.

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

File operations mainly focus on some key points

UEFI has built-in EFI_SIMPLE_FILE_SYSTEM_PROTOCOL for operating FAT file system. Through OpenVolume in this protocol, we can get the root directory handle on the FAT file system. The directory handle (FEI_FILE_PROTOCOL) contains the file operation interface for operating files in the directory.

Use EFI_SIMPLE_FILE_SYSTEM_PROTOCOL

```
Al generated projects
                                                                                                                                登录复制
1 | EFI_STATUS Status;
   EFI_SIMPLE_FILE_SYSTEM_PROTOCOL
2
                                     *SimpleFileSystem;
3
   EFI_FILE_PROTOCOL *Root;
4
6
   Status = gBS->LocateProtocol (
7
8
                      &gEfiSimpleFileSystemProtocolGuid,
9
10
                      &SimpleFileSystem
11
   if (EFI ERROR(Status)){
12
13
     return Status;
14
15
   Status = SimpleFileSystem->OpenVolume(SimpleFileSystem,&Root);
                                                              收起 へ
```

After getting the pointer to the EFI FILE PROTOCOL instance of the root directory on the partition file system, you can operate the files on the partition

Write a file using the Write function

Al generated projects 登录复制

```
EFI_STATUS Status;
3
   UINTN BufSize;
4
5
   CHAR16 *Textbuf = (CHAR16*)L"this is test file\n";
6
7
   EFI_FILE_PROTOCOL *FileHandle =0;
8
   //创建新文件,如果文件已经存在,则打开
9
10
11
   Status = Root->Open(Root,
12
                            &FileHandle,
                            (CHAR16)L"testfile.txt",
13
14
                             EFI_FILE_MODE_CREATE | EFI_FILE_MODE_READ | EFI_FILE_MODE_WRITE,
15
16
   if(FileHandle && !EFI_ERROR(Status)){
17
18
               BufSize = StrLen(Textbuf) *2;
19
20
   Status = FileHandle->Write(
21
                   FileHandle,&BufSize,Textbuf);
22
23
   Status = FileHandle->Close (FileHandle);
24
```

Al generated projects 登录

```
1 | EFI STATUS
                                    Status:
2
     EFI_SIMPLE_FILE_SYSTEM_PROTOCOL *ptSFS;
     EFI FILE PROTOCOL
                                      *ptRootFile;
3
4
     EFI_FILE_PROTOCOL
                                      *ptFile;
     EFI_FILE_INFO
5
                                      *FileInfo;
                                     FileInfoSize;
6
     UINTN
     UINT8
                                      *Buffer;
7
8
9
    Status
               = EFI_SUCCESS;
     ptRootFile = NULL;
10
               = NULL;
11
     ptFile
12
     FileInfo = NULL:
13
     HandleBuffer = NULL;
14
15
    Status = ptSFS->OpenVolume(ptSFS, &ptRootFile);
16
         if (!EFI_ERROR (Status)) {
17
           Status = ptRootFile->Open(
18
                      ptRootFile,
19
                       &ptFile,
20
                       FileName,//你的文件名,带目录
                      EFI_FILE_MODE_WRITE|EFI_FILE_MODE_READ,
21
22
23
                      );
24
    if (!EFI_ERROR (Status)) {
25
26
       FileInfo = NULL;
27
       FileInfoSize = 0;
28
       Status = ptFile->GetInfo (
29
                          ptFile.
30
                           &gEfiFileInfoGuid,
31
                          &FileInfoSize.
32
                          FileInfo
33
                          ):
34
       if(Status == EFI_BUFFER_T00_SMALL){
35
         FileInfo = AllocateZeroPool(FileInfoSize);
36
         if(FileInfo == NULL){
           Status = EFI_OUT_OF_RESOURCES;
37
38
         } else {
39
           Status = ptFile->GetInfo(
40
                               ptFile,
41
                               &gEfiFileInfoGuid,
                               &FileInfoSize,
42
43
                               FileInfo
44
                               );
45
46
       }
47
48
49
     Status = ptFile->Read(ptFile, &FileInfo->FileSize, Buffer);
50
51
   ptFile->Close(ptFile);
   ptRootFile->Close(ptRootFile);
                                                              收起 へ
```

The use of this read function is different from that in the book. This BIOS practical application will be discussed later.

about Us Careers Business Seeking coverage 2400-660-0108 kefu@csdn.net Customer Service 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

登录复制