

UEFI Principles and Programming Practice--Hard Disk Related Protocols: BlockIo/DiskIo/DiskInfo/PassThrough

原创

Anthony


Posted on 2021-11-30 16:42:45

Read 2.9k

Collection 6

Likes

Category Column: [UEFI](#) Article Tags: [UEFI](#) [Linux](#) [C](#)

UEFI This column includes this content

23 articles

Subscribe to our column

摘要 This article reveals the difference between BlockIo and DiskIo in hard disk operations, introduces their functional limitations and applications in obtaining NVMe hard disk information, and deeply explores how to perform advanced operations through PassThrough.

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

The hard disk is a block device, so each hard disk device controller is installed with a BlockIo instance, a BlockIo2 instance, and then a DiskIo instance and a DiskIo2 instance. The difference between the two is that BlockIo can only read and write devices by block, while DiskIo can read and write disks from any cheap source and can read any number of bytes. BlockIo is the abbreviation of EFI_BLOCK_IO_PROTOCOL, and DiskIo is the abbreviation of EFI_DISK_IO_PROTOCOL.

Although BlockIo and DiskIo greatly facilitate our disk operations, their functions are very limited. Through BlockIo and DiskIo, we can only read, write, and flush the hard disk device. If we want to perform more operations on the hard disk, we need to send commands to the hard disk through PassThrough.

Take obtaining NVME hard disk information as an example:

AI generated projects

登录复制

```
1 # define TEMP_POOL_SIZE 300
2
3 VOID
4 UpdateNvmePortInfo ()
5 {
6     EFI_STATUS Status;
7     UINTN HandleCount;
8     EFI_HANDLE *HandleBuffer;
9     EFI_DISK_INFO_PROTOCOL *DiskInfo;
10    UINTN Index;
11    CHAR16 *UnicodeStr;
12    EFI_IDENTIFY_DATA *IdentifyDriveInfo;
13    NVME_ADMIN_CONTROLLER_DATA NvmeIdentifyControllerData;
14    //找出所有支持BlockIo的设备
15    Status = gBS->LocateHandleBuffer (ByProtocol, &gEfiBlockIoProtocolGuid, NULL, &HandleCount, &HandleBuffer);
16    //遍历所有支持BlockIo设备, 获取他们全部的信息
17    for (Index = 0; Index < HandleCount; Index++) {
18        Status = gBS->HandleProtocol (HandleBuffer [Index], &gEfiDiskInfoProtocolGuid, (VOID **)&DiskInfo);
19
20        if (!EFI_ERROR (Status)) {
21            // if Nvme HDD devices
22            UnicodeStr = (CHAR16*)AllocateZeroPool (TEMP_POOL_SIZE);
23            IdentifyDriveInfo = (EFI_IDENTIFY_DATA*)AllocatePool (sizeof (EFI_IDENTIFY_DATA));
24            //通过CompareGuid, 定位NVME硬盘设备
25            if (CompareGuid (&DiskInfo->Interface, &gEfiDiskInfoNvmeInterfaceGuid)) {
26                //获取NVME硬盘设备全部信息
27                Status = NvmeIdentifyController (HandleBuffer[Index], &NvmeIdentifyControllerData);
28                if (EFI_ERROR (Status)) {
29                    continue;
30                }
31                //从信息中筛选出设备ModelName, 显示在setup界面
32                NvmeIdentifyControllerData.Mn[40] = 0;
33                UnicodeSprintfAsciiFormat (UnicodeStr, TEMP_POOL_SIZE, "%a", (CHAR8 *)&NvmeIdentifyControllerData.Mn);
34                HiiSetString (gFrontPagePrivate.HiiHandle, STRING_TOKEN(STR_NVME_DISK_INFO), UnicodeStr, NULL) ;
35            }
36        }
37    }
38 }
```

收起 ^

To be written...

