UEFI Basic Tutorial (Part 9) - Simple Use of EVENT

原创 xiaopangzi313 Posted on 2020-06-06 10:21:05 Read 3.8k Collection 8 Likes 2

Category columns: 15_Firmware Development

摘要 This article describes how to use the EVENT API to implement key and timer event monitoring in a UEFI environment. By creating and waiting for events, it demonstrates the application of asynchronous mechanisms in limited multitasking support.

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

UEFI Basic Tutorial (VIII) - Simple Use of EVENT

1. Write source code

1. Write the UEFI Application
   code C:\edkii\OvmfPkg\MyHelloWorldEvent\MyHelloWorldEvent.c,

C AI generated projects 登录复制 run

```c
EFI_STATUS MyHelloWorldEventEntry(
        IN EFI_HANDLE           ImageHandle,
        IN EFI_SYSTEM_TABLE     *SystemTable
        )
{
    EFI_STATUS Status;

    UINTN        Index=0;
    EFI_INPUT_KEY      Key;
    EFI_EVENT myEvents[2] = {0};


    Print (L"[MyHelloWorldEvent] MyHelloWorldEventEntry Start..\n");
    // 1. 生成事件
    // 生成按键事件
    myEvents[0] = gST->ConIn->WaitForKey;



    // 生成Timer事件
    Status = gBS->CreateEvent(EVT_TIMER , TPL_CALLBACK, (EFI_EVENT_NOTIFY)NULL, (VOID*)NULL, &myEvents[1]);
    if(EFI_ERROR(Status)){
      Print (L"[MyHelloWorldEvent] CreateEvent %r ...\n",Status);
      return Status;
    }

    Status = gBS->SetTimer(myEvents[1],TimerPeriodic , 100 * 1000 * 1000);//设置10秒定时
    if(EFI_ERROR(Status)){
      Print (L"[MyHelloWorldEvent] SetTimer %r ...\n",Status);
      return Status;
    }

    while (1){
        //2. 阻塞并等待事件被触发
        Status = gBS->WaitForEvent(2, myEvents, &Index);
```

```
        if(EFI_ERROR(Status)){
          Print (L"[MyHelloWorldEvent] WaitForEvent %r ...\n",Status);
          return Status;
        }

        if (Index == 0){
            // 读取按键键值并显示
            Status = gST->ConIn->ReadKeyStroke (gST->ConIn, &Key);
            switch (Key.ScanCode){
              case SCAN_UP:
                Print (L"[MyHelloWorldEvent]  Key UP is Pressed..\n");
                break;

              case SCAN_DOWN:
                Print (L"[MyHelloWorldEvent]  Key Down is Pressed..\n");
                break;

              ...

              case SCAN_ESC:
                Print (L"[MyHelloWorldEvent]  Key ESC is Pressed..\n");
                goto End;

              default:
                Print (L"[MyHelloWorldEvent] Key %a is Pressed..\n",(CHAR8 *)&Key.UnicodeChar);
                break;
            }
        }else{
            Print (L"[MyHelloWorldEvent]  Timer event is trigered ..\n");
        }
    }

End:
    //销毁事件
    Status = gBS->CloseEvent(myEvents[0]);
    Status = gBS->CloseEvent(myEvents[1]);

    Print (L"[MyHelloWorldEvent] MyHelloWorldEventEntry End..\n");
    return Status;
}
```

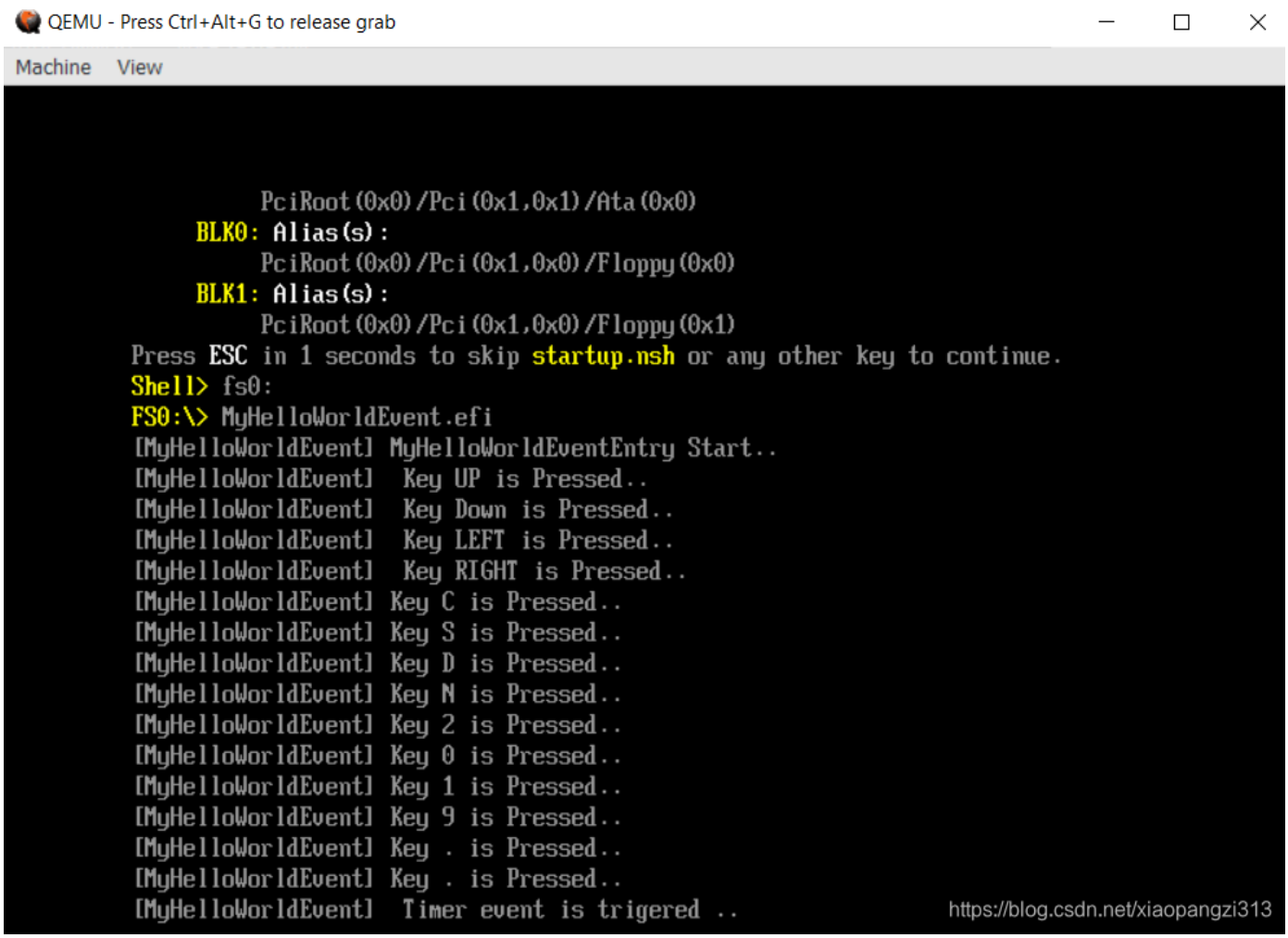
收起 ^

2. Compile and generate EFI files

Run and `edksetup.bat` compile the entire OvmfPkg Package

3. Run UEFI APP `MyHelloWorldEvent.efi`

1. Copy `C:\edkii\Build\OvmfX64\DEBUG_VS2013x86\FV\OVMF.fd` to `C:\qemu` ; Copy `C:\edkii\Build\OvmfX64\DEBUG_VS2013x86\X64\OvmfPkg\MyHelloWorldEvent\MyHelloWorldEvent\OUTPUT\MyHelloWorldEvent.efi` to virtual disk `HDD_BOOT.img`
2. Execute , and then execute in , the result is as follows, `setup-qemu-x64.bat | findstr MyHelloWorldEvent UEFI SHELL MyHelloWorldEvent.efi`



IV. Summary

Events is another type of object managed by UEFI Service, which can provide an asynchronous mechanism to support limited multitasking, including keyboard, mouse, timer, protocol, etc. This article uses some APIs of Event to monitor key and timer events and make simple processing. Among them, CreateEvent is used to generate events, WaitForEvent is used to block and wait for events to be triggered, and CloseEvent is used to destroy events.

Event DEMO source code