

DIIMain和多线程死锁计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_DIIMain\\_E5\\_92\\_8C\\_c97\\_645921.htm](https://www.100test.com/kao_ti2020/645/2021_2022_DIIMain_E5_92_8C_c97_645921.htm)

在Windows操作系统中，DLL（动态链接库）技术有很多优点。例如，多个应用程序可以共享一个DLL文件，真正实现了资源"共享"，大大缩小了应用程序的执行代码，有效地利用了内存，而且DLL文件作为一个单独的模块，封装性、独立性好，有利于提高软件开发和维护的效率。DIIMain是可选择的DLL入口指针，当进程和线程启动和终止时被系统调用，分别进行创建资源和释放资源等操作，特别地，也可以在DLL被装载进进程空间时（即DIIMain响应DLL\_PROCESS\_ATTACH通知时）创建线程，在DLL从进程空间卸载时（即DIIMain响应DLL\_PROCESS\_DETACH通知时）结束线程。但是，在DIIMain中无论是创建线程还是结束线程，都特别要注意一个规则，那就是DIIMain的顺序调用规则。

1、DIIMain的顺序调用规则

Windows操作系统中是顺序调用DLL的入口函数DIIMain的。当进程被创建时，系统也为该进程创建了一个互斥对象。每个进程都有它自己的互斥对象。进程互斥对象的一个作用是，序列化在需要调用DIIMain的4种情况下DIIMain的执行：DLL\_PROCESS\_ATTACH、DLL\_THREAD\_ATTACH、DLL\_THREAD\_DETACH和DLL\_PROCESS\_DETACH。DIIMain函数的第二个参数指示出调用DIIMain的原因。在DIIMain中创建线程或终止线程时，如果违背了DIIMain的这个顺序调用规则，程序就会发生死锁。下面就DIIMain中创建线程和终止线程两种情况下的死

锁分别进行讲述。 2、DllMain中创建和终止线程时的死锁 2.1、装载DLL时创建的线程的为什么没有运行 考虑在一个多线程程序中，某个DLL被加载进程地址空间时，该DLL的DllMain启动了一个线程，然后立即调用一个应答事件对象的 WaitForSingleObject函数，以确认在继续进行其余的DllMain处理之前，新产生的线程能够正确地执行一些操作。类似的代码如下：`//-----start 100Test` 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)