Mmo框架模块添加

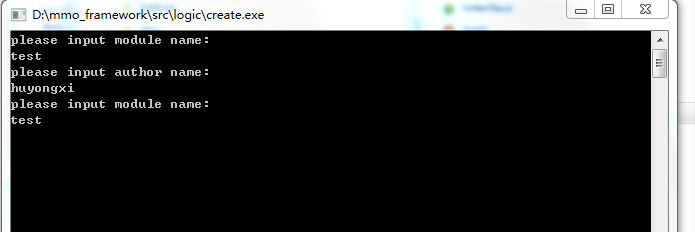
## 模块介绍

Mmo框架中，每个模块都对应一个类，类的继承关系是，class M 继承class IM，class IM继承 class IModule（假设模块名是M）。class IModule，定义了所有模块都需要的方法和属性。class M和 class IM是我们自己去写的，我们把需要给其他模块使用的函数，在class IM中声明为纯虚函数，在class M中实现。

多个模块可以放在一个动态链接库中，服务器进程从中加载模块。首先会调用模块的Initialize方法，初始化模块的数据和读取模块自己的配置文件。然后调用模块的Launchd方法，启动模块。一般我们在启动时候，会注册网络消息的回调函数，事件的回调函数。

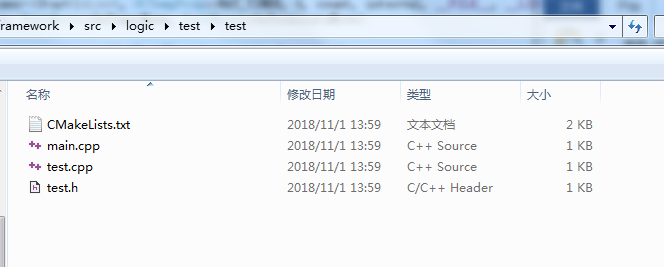
## 如何加入新模块

在src/logic目录下，有一个create.exe程序。用它可以生成一个模块的基本代码。如下图所示：

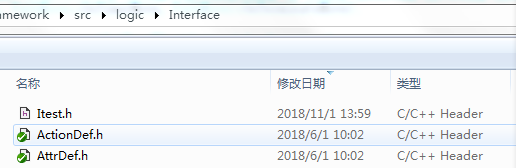


我们根据提示输入模块名字和作者名字，按回车键会自动生成./test/test目录。会生成一个模块的基本的代码。

主要内容有：



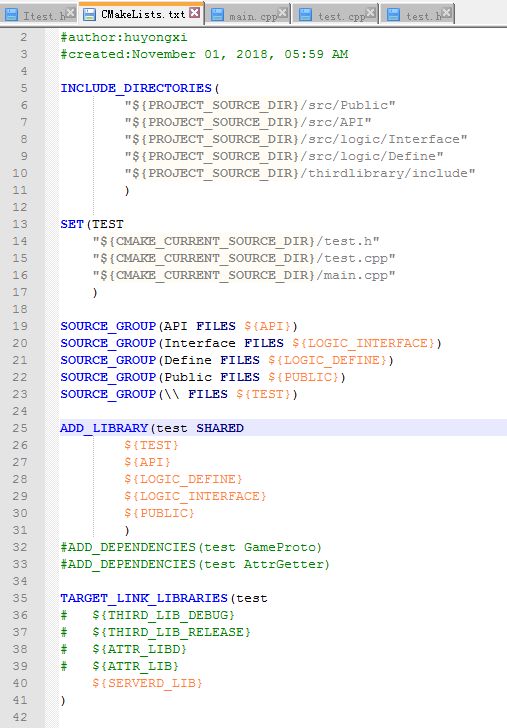
还会在 ./Interface目录下生成Itest.h文件



### 模块基本的文件组成

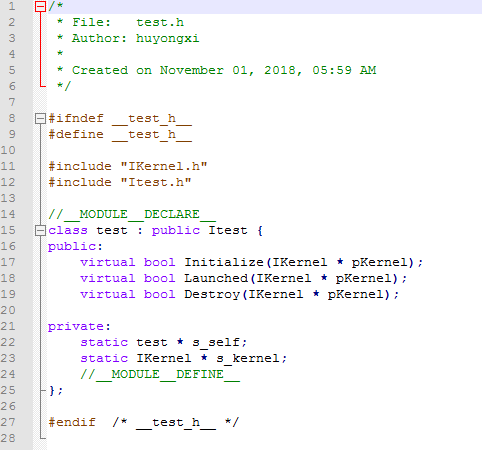
下面介绍一下每个生成的文件

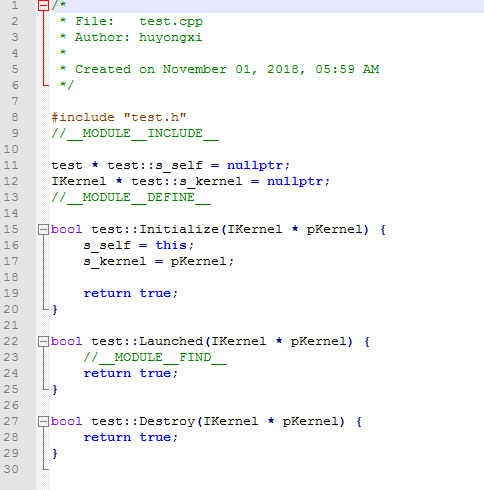
1. 会生成该模块CMakeList.txt文件，主要内容如下：



如果我们的模块需要增加新的cpp文件或者增加include目录，我们都可以在上面基础上进行修改。可以看出我们每个模块都会作为一个动态链接库。

1. 生成的Test.h 和 test.cpp文件：





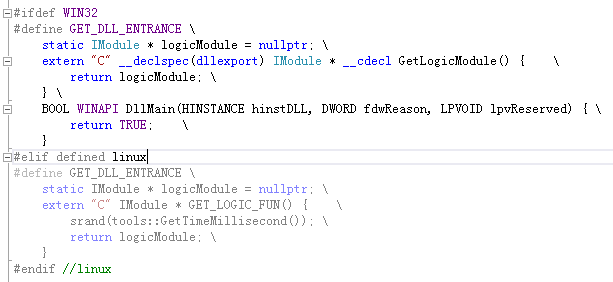
这两个文件主要声明了一个最基本的class，每个模块都对应一个class。Initialize, Launched, Destroy这些是必须要重写的虚函数。主要负责模块的初始化，启动，销毁。

它们都有一个参数pKernel, 这参数是服务器进程核心的部分，可以通过它获取到其它模块指针，增删定时器，异步任务还有网络等一些基本的组件。

我们可以给这个类增加新的成员变量和成员函数，增加模块的功能。

## 一些重要的宏

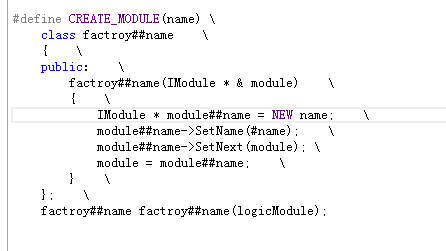
宏 GET\_DLL\_ENTRANCE



这个宏的意思是，创建一个全局变量模块指针logicModule在动态链接库中, IModule是一个纯虚类，它有一个重要的成员 IModule\* \_nextModule，后面会用到。我们模块的继承顺序是这样的，假设模块名字叫A，class A 会继承 class IA（纯虚类） class IA 继承 IModule。

这个宏定义了一个导出函数GetLogicModule()，它返回模块指针，所以我们可以从动态链接库中获取到模块。

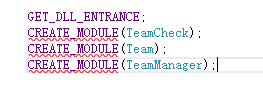
宏 CREATE\_MODULE



这个宏的意思是，它根据参数name声明了一个类，然后定义该类的一个对象，利用全局变量的构造函数来给logicModule变量赋值。我们new一个模块时，可以设置它的name, \_ nextModule。

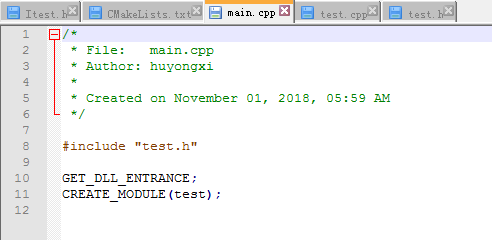
我们可以在一个dll里，创建很多模块，通过\_nextModule指针，像链表一样，把所有的模块链起来，通过logicModule这一个指针，就可以访问到所有模块。可以看出我们创建的顺序和这个链的顺序相反的。

比如这种形式：



它的意思就是，创建了3个模块，这些宏的参数是类名。我们可以通过函数GetLogicModule()获取到logicModule变量，从而能获取到每个模块。

1. 再来看一下我们生成的main.cpp



它的意思就是创建了test模块。（就是class Test 的实例）我们可以从test.dll中拿到这个实例。

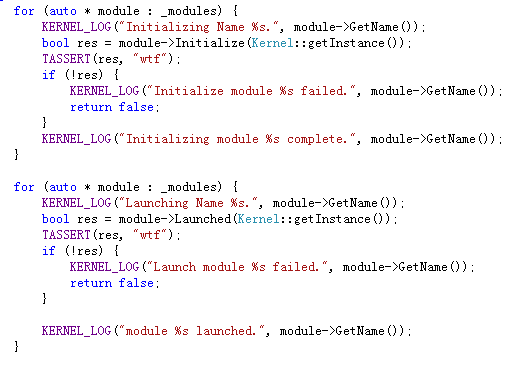
Serverd.exe会根据配置文件加载不同的dll，形成不同的服务器进程如：gate，scene，logic等不同的进程。

Serverd.exe会把dll中的模块，都保存在map中：

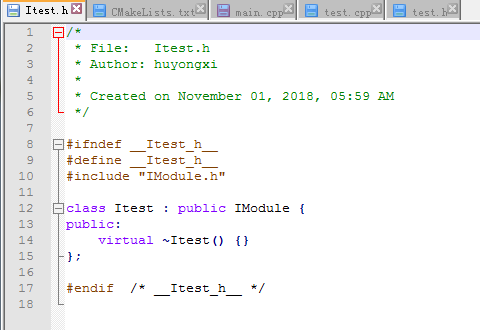


logic变量值就等于dll中各个模块实例的地址，name就是模块的名字。

然后就是初始化Initialize()，和启动Launched()：

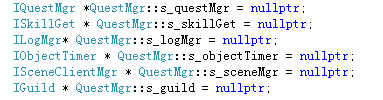


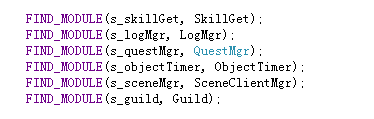
1. 最后看一下./Interface目录下生成Itest.h文件



这个文件作用是接口，作为include包含的头文件，其他的模块可以包含这个头文件，我们将需要提供给其他模块使用的函数，在这里声明为纯虚函数。实现放在模块的定义中。

比如我们可以在其他模块经常看到这样的语句：





宏FIND\_MODULE用来查找其他模块的，其实就是利用\_nameModules这个map。



宏TMODULE也可以返回模块指针。

模块添加完成之后，在需要这个模块的进程的module.xml配置文件中，加入要加载的模块。