引擎内包装了各个平台的API，各种跟平台系统相关的操作，封装到FXXXPlatformXXX的各个类中去了，通过一个统一的父类系FGeneraticPlatformXXX来抽象跟平台相关的各种操作，如文件读写，打印控制台，内存，字符串，线程等等。这些定义都在Engine/Runtime/Core/Public/GenericPlatform/文件夹内。

我们来讨论关于打印

由于引擎跨平台的需求，首先在PlatformOutputDevices.h内根据宏定义包含不同平台下的PlatformOutputDevices.h文件，例如在windows平台下就包含Windows/WindowsPlatformOutputDevices.h,打开这个头文件会看到FWindowsPlatformOutputDevices继承于FGenericPlatformOutputDevices,并用typedef将其重新定义成FPlatformOutputDevices供引擎内的其他类使用。其他平台下的类定义也是一样的，通过这样定义，对外暴露统一的类型。

CoreGlobals.h内定义了许多全局对象，一般来讲Log系统是很常用的全局需要使用的系统，所以其内部定义了GLogConsole的全局声明，在CoreGlobals.cpp内首先初始化为空。在LaunchEngineLoop.cpp内，定义了一个静态的TUniquePtr<FOutputDeviceConsole> GScopedLogConsole指针来指向一个FOutputDeviceConsole, 在FEngineLoop::PreInit()函数内，通过FPlatformOutputDevices::GetLogConsole()静态方法调用，获得一个FOutputDeviceConsole的指针。(4.15版本大约在909行)，随后将其赋给GLogConsole全局变量。(4.15版本大约在1359行)

GConfig在AppInit()内由FConfigCacheIni::InitializeConfigSystem()静态函数赋值。

------------

命令行可传入参数

-GameINI="" //指定Game.ini路径，如果未指定，使用默认

-LOG //指定是否开启命令控制台的Log，如果指定这个参数则开启控制台