

## 安装Ubuntu子系统

对生成linux平台的游戏有用，对Android系统没用，这里只为测试cmake工程。

```
ubuntu@LAPTOP-HH9HSFUJ: ~$  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
This message is shown once a day. To disable it please create the  
/home/ubuntu/.hushlogin file.  
ubuntu@LAPTOP-HH9HSFUJ:~$  
ubuntu@LAPTOP-HH9HSFUJ:~$
```

参考VRTemplate工程创建Android的编译环境

[yuanjianking/VRTemplate \(github.com\)](https://github.com/yuanjianking/VRTemplate)

使用vs2022创建cmake工程，并尝在window和Ubuntu子系统两侧执行，输出Hello CMake。



用vs2022

上一步(B)

下一步(N)

查看DLL的函数命令，主要查看生成的dll是否正确。

dumpbin /exports "CommonDll.dll"

生成window用的库

设置成release模式

本地计算机 x64 Release 默认值

右键点击CMakeLists.txt，选择配置CommonLib，然后点击生成。

生成的库文件

Unreal Projects > UseThirdPartySample > ThirdParty > Common		
名称	修改日期	
Share	2023/8/11 11:00	
Static	2023/8/11 11:00	
CommonDll.dll	2023/8/11 14:42	
CommonDll.dll.manifest	2023/8/11 14:42	
CommonDll.exp	2023/8/11 14:42	
CommonDll.lib	2023/8/11 14:42	
CommonLib.lib	2023/8/11 14:42	staitc目录

生成android用的库

因为ABI（Application Binary Interface）接口的问题，上面ubunt生成的.a和.so在Android系统上面没法使用，所以要用android studio自带的cmake编译生成。

AppData > Local > Android > Sdk > cmake > 3.10.2.4988404 > bin

名称	修改日期	类型
cmake.exe	2023/1/25 18:02	应用程序
cmcldeps.exe	2023/1/25 18:02	应用程序
cpack.exe	2023/1/25 18:02	应用程序
ctest.exe	2023/1/25 18:02	应用程序
ninja.exe	2023/1/25 18:02	应用程序

C:\Users\MI\AppData\Local\Android\Sdk\cmake\3.10.2.4988404\bin>cmake -H"C:\Users\MI\Documents\Unreal Projects\UseThirdPartySample\Common" -B"C:\Users\MI\Documents\Unreal Projects\UseThirdPartySample\Common\generated\ninja\project\debug\arm64-v8a" -DANDROID\_ABI=arm64-v8a -DANDROID\_PLATFORM=android-26 -DANDROID\_NDK=C:\Users\MI\AppData\Local\Android\Sdk\ndk\25.1.8937393

C:\Users\MI\source\repos\Common是cmake工程根目录

C:\Users\MI\source\repos\Common\generated\ninja\project\debug\arm64-v8a是cmake生成的ninja工程目录

C:\Users\MI\AppData\Local\Android\Sdk\ndk\25.1.8937393是ndk目录，要跟ue工程配置的ndk版本保持一致

进入到ninja工程目录执行

C:\Users\MI\Documents\Unreal Projects\UseThirdPartySample\Common\generated\ninja\project\debug\arm64-v8a>C:\Users\MI\AppData\Local\Android\Sdk\cmake\3.10.2.4988404\bin\ninja.exe

生成的.a和.so

› Unreal Projects › UseThirdPartySample › Common › generated › ThirdParty › Common

名称	修改日期	类型	大小
 libCommonDll.so	2023/8/11 15:09	SO 文件	6 KB
 libCommonLib.a	2023/8/11 15:09	A 文件	4 KB

将.so移动到share文件夹，将.a移动到static文件夹








› Unreal Projects › UseThirdPartySample › ThirdParty › Common

名称	修改日期	类型
 Share	2023/8/11 11:00	文件夹
 Static	2023/8/11 15:15	文件夹

a可以直接打包到apk

so必须通过编译脚本才能打包到apk

› Unreal Projects › UseThirdPartySample › ThirdParty › Common › Share

名称	修改日期	类型
 CommonDll.dll	2023/8/11 10:59	DLL 文件
 CommonDll.dll.manifest	2023/8/11 10:59	MANIFEST 文件
 CommonDll.exp	2023/8/11 10:59	EXP 文件
 CommonDll.lib	2023/8/11 10:59	Object File Librai
 libCommonDll.so	2023/8/11 15:09	SO 文件
 <u>My_APL_armv8.xml</u>	2023/3/15 12:28	XML 源文件
 Sub.h	2023/3/14 22:12	C Header 源文件

```
<root xmlns:android="http://schemas.android.com/apk/res/android">
  <!-- init section is always evaluated once per architecture -->
  <init>
    <setBool result="bSupported" value="false"/>
    <isArch arch="arm64-v8a">
      <setBool result="bSupported" value="true"/>
    </isArch>
  </init>
  <!-- optional files or directories to copy to Intermediate/Android/APK -->
  <resourceCopies>
    <isArch arch="arm64-v8a">
      <copyFile src="$(PluginDir)/libCommonDll.so"
        dst="$(BuildDir)/libs/arm64-v8a/libCommonDll.so" />
    </isArch>
  </resourceCopies>
  <!-- optional libraries to load in GameActivity.java before libUE4.so -->
  <soLoadLibrary>
    <if condition="bSupported">
      <true>
        <loadLibrary name="myso" failmsg="Failed to load myso library" />
      </true>
    </if>
  </soLoadLibrary>
</root>
```

编译运行window平台和android平台的应用  
在ue编辑器里执行打包