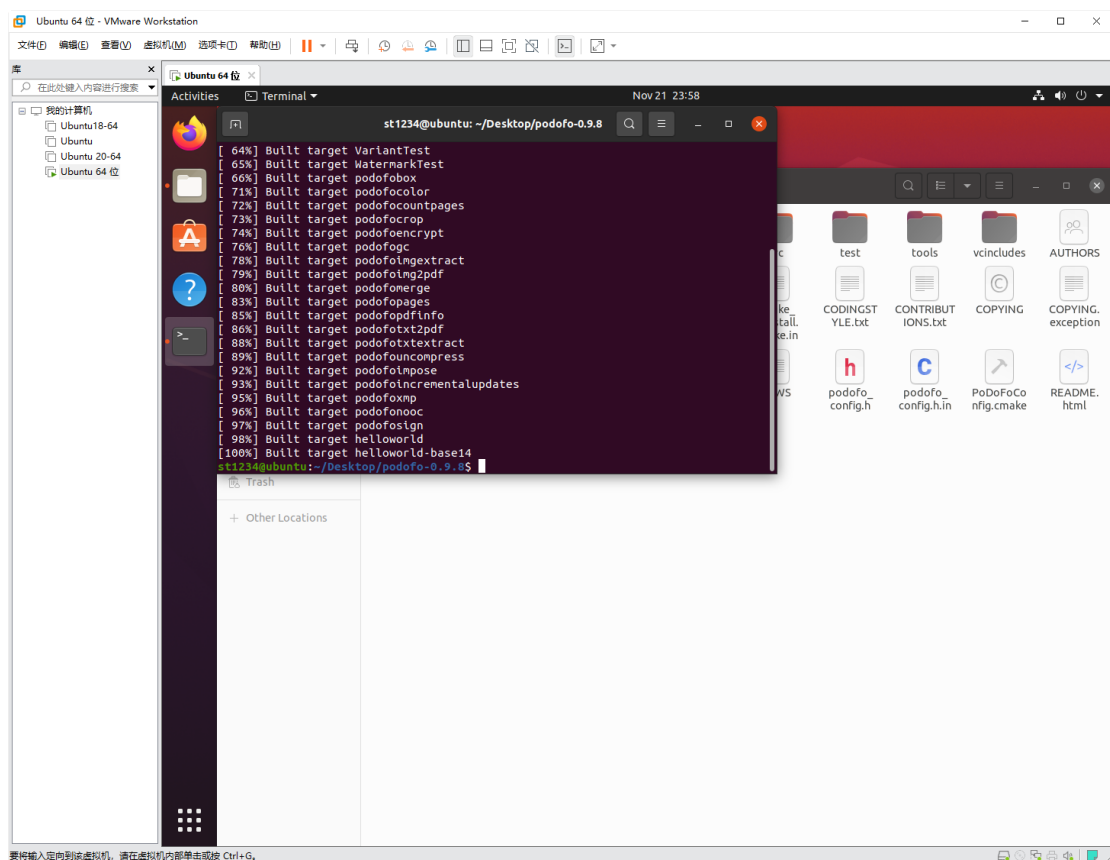
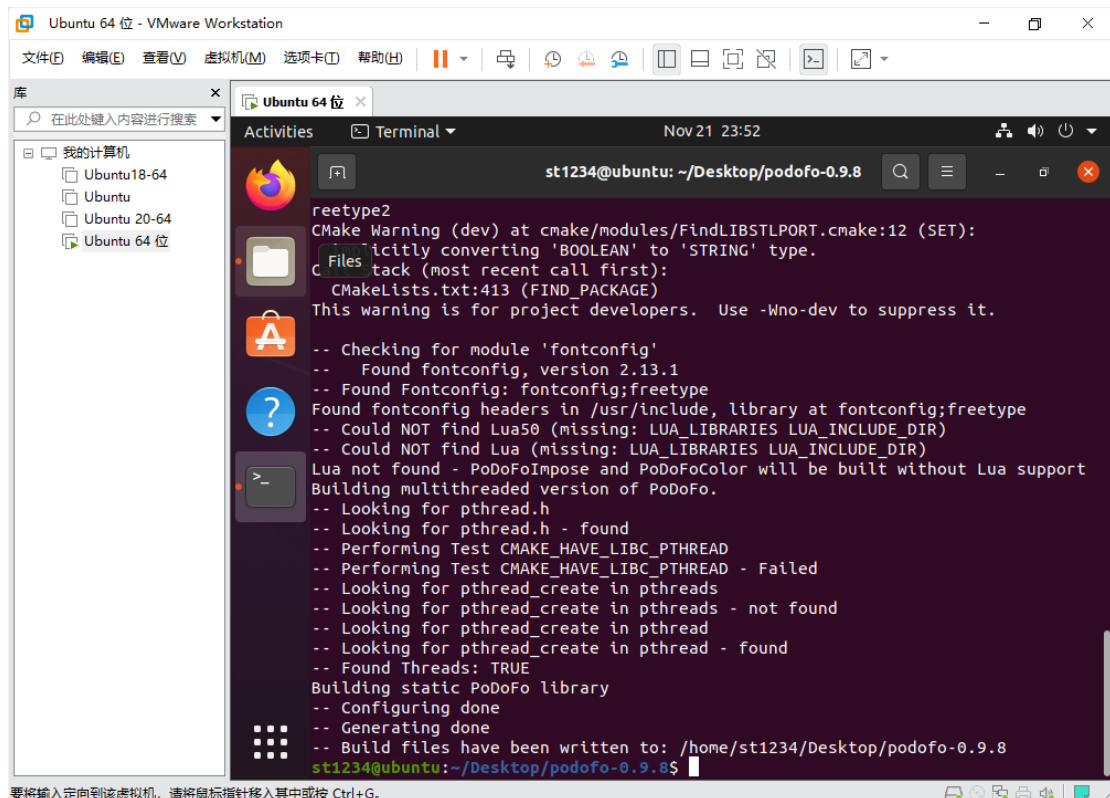
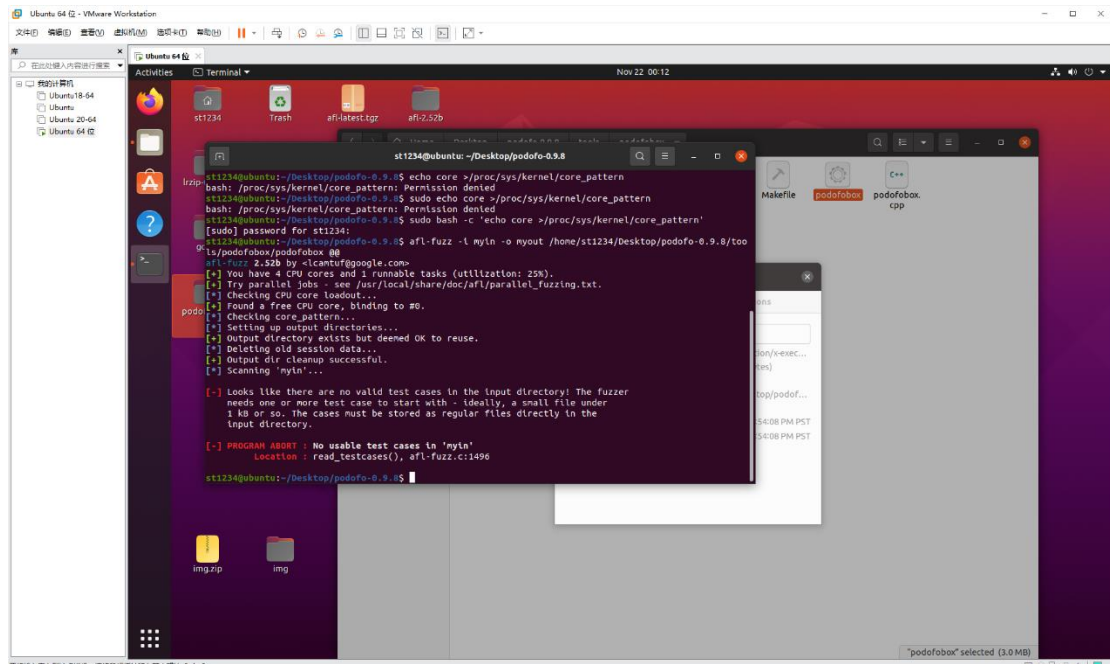


以 podofo 为例
教程建立在能够用传统 gcc 编译的基础上（即你的依赖都已经成功安装）



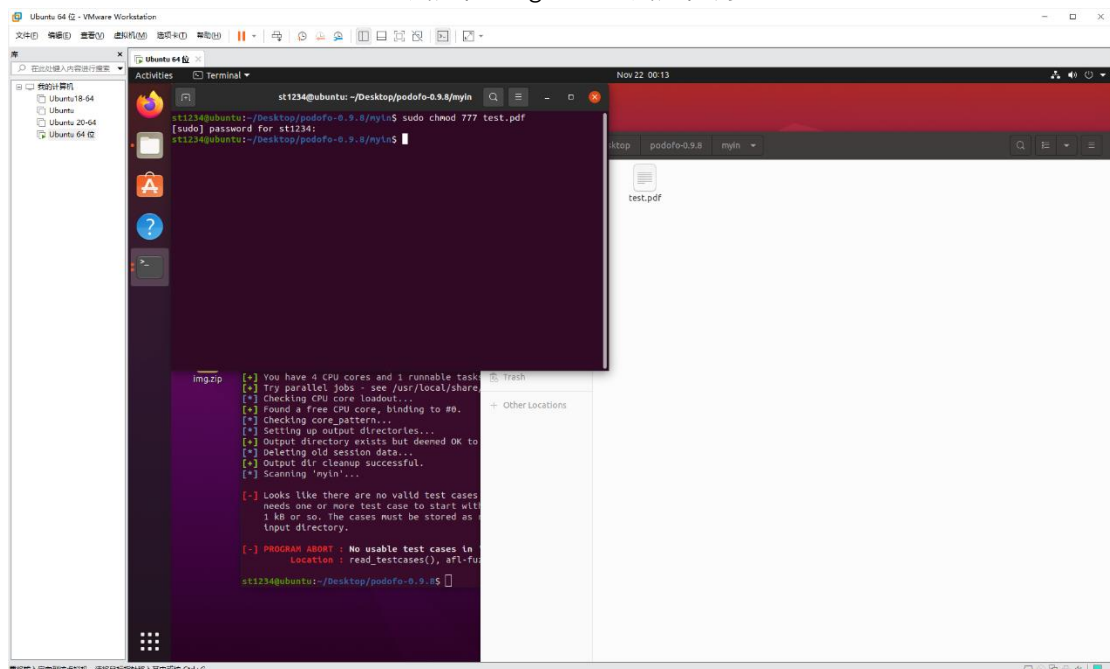
一、使用 afl 前，遇到的一些 bug

1.测试用例 bug——no usable test cases in myin



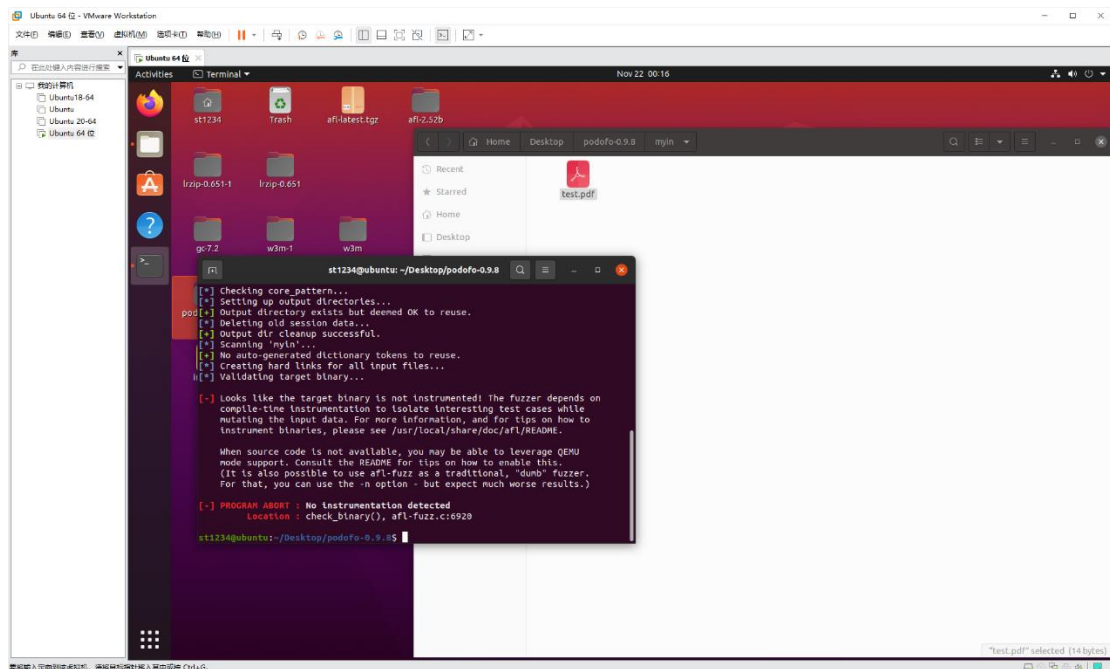
解决方法：myin 中此时已经有了测试用例，但是是完全空的，要换成一个不那么空的测试用例

2.测试用例 bug——测试用例为空



解决方法：你可能是使用了命令行+sudo 来创建测试文件，此时的文件带锁，在终端输入 sudo chmod 777 文件名 即可

3.binary is not instrumented!

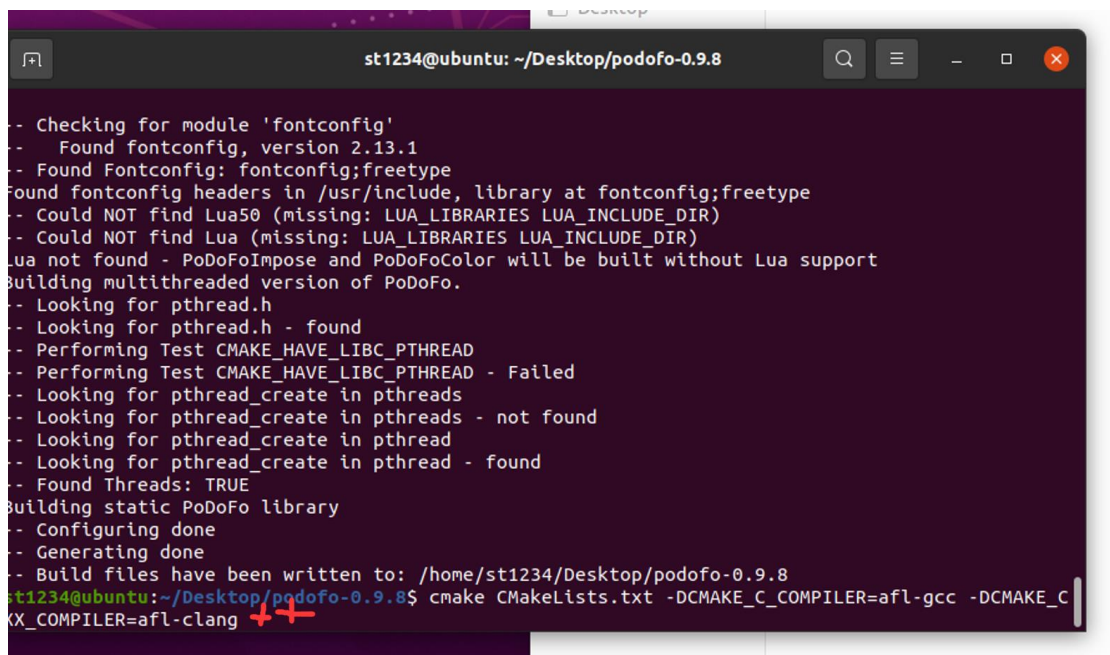


这时的问题就是你没有用 afl-gcc 编译，导致出现了用例正确但是找不到的问题。

解决方法：详见二、

二、只有 cmake 怎么使用 afl?

1.参考



cmake CMakeLists.txt -DCMAKE_C_COMPILER=afl-gcc -DCMAKE_CXX_COMPILER=afl-clang++

2.make clean all 时报错

```
clang: warning: -lm: linker input unused [-Wunused-command-line-argument]
afl-as 2.52b by <lcantuf@google.com>
[+] Instrumented 522 locations (64-bit, non-hardened mode, ratio 100%).
[ 53%] Linking CXX static library libpodof0.a
[ 53%] Built target podof0_static
[ 53%] Building CXX object test/ContentParser/CMakeFiles/ContentParser.dir/main.cpp.o
afl-cc 2.52b by <lcantuf@google.com>
clang: warning: -lm: linker input unused [-Wunused-command-line-argument]
afl-as 2.52b by <lcantuf@google.com>
[+] Instrumented 347 locations (64-bit, non-hardened mode, ratio 100%).
[ 54%] Linking CXX executable ContentParser
afl-cc 2.52b by <lcantuf@google.com>
/usr/bin/ld: /home/st1234/Desktop/podof0-0.9.8/src/podof0/libpodof0.a(PdfObject.cpp.o): undefined reference to symbol 'floor@@GLIBC 2.2.5'
/usr/bin/ld: /lib/x86_64-linux-gnu/libm.so.6: error adding symbols: DSO missing from command line
clang: error: linker command failed with exit code 1 (use -v to see invocation)
make[2]: *** [test/ContentParser/CMakeFiles/ContentParser.dir/build.make:100: test/ContentParser/ContentParser] Error 1
make[1]: *** [CMakeFiles/Makefile2:917: test/ContentParser/CMakeFiles/ContentParser.dir/all] Error 2
make: *** [Makefile:130: all] Error 2
st1234@ubuntu: ~/Desktop/podof0-0.9.8$
```

对标示部分进行搜索，比如这个的结果是因为 gcc 编译时缺少 -lm 选项

解决方法：

打开 CMakeLists.txt

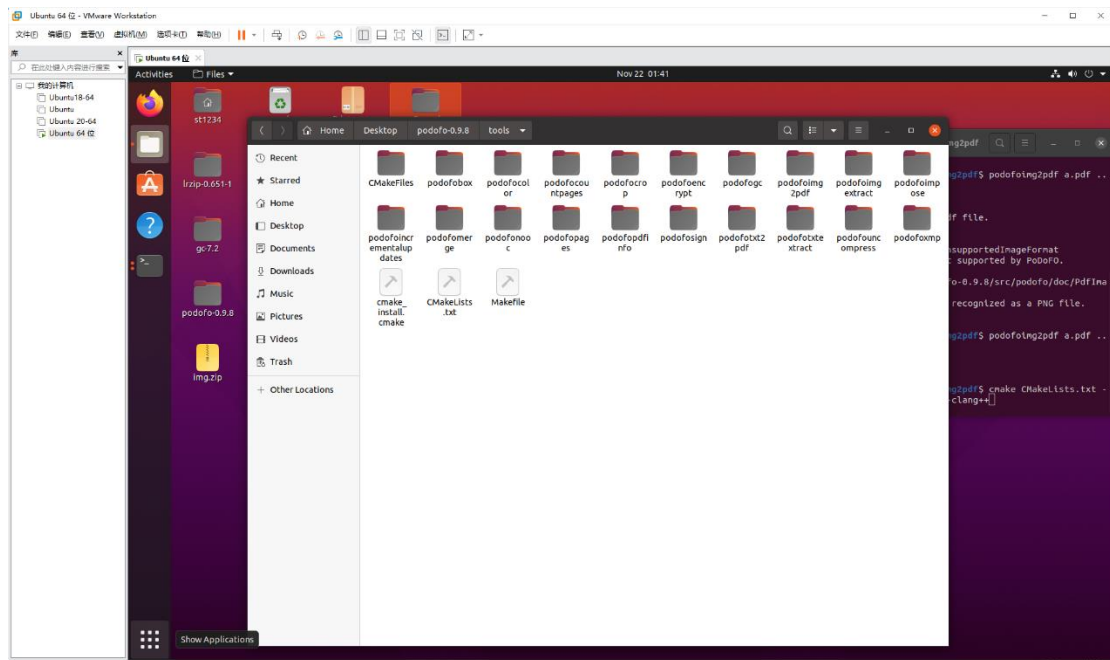
在第二行输入

add_compile_options(-lm)

```
1 cmake_minimum_required(VERSION 2.6)
2
3 ***** IMPORTANT *****
4 # Look at http://www.vtk.org/Wiki/CMake_HowToPlatformChecks
5 # and the other wiki entries before you add anything. You might not need to.
6 *****
7
8 #
9 # Project name and version
10 #
11 PROJECT(Podof0)
12
13 SET(PODOFO_VERSION_MAJOR "0" CACHE STRING "Major part of Podof0 version number")
14 SET(PODOFO_VERSION_MINOR "9" CACHE STRING "Minor part of Podof0 version number")
15 SET(PODOFO_VERSION_PATCH "8" CACHE STRING "Patchlevel part of Podof0 version number")
16 SET(PODOFO_SOVERSION "${PODOFO_VERSION_MAJOR}.${PODOFO_VERSION_MINOR}.${PODOFO_VERSION_PATCH}")
17 SET(PODOFO_LIBVERSION "${PODOFO_SOVERSION}")
18
19 #
20 #
21 # Main includes
22 #
23 INCLUDE(CheckCXXSourceCompiles)
24 INCLUDE(CheckIncludeFile)
25 INCLUDE(CheckLibraryExists)
26 INCLUDE(CheckBigEndian)
27 INCLUDE(CheckTypeSize)
28
29 #
30 # Setup CMake Policies
31 #
32
33 # Prefer files in CMAKE_MODULE_PATH over shipped ones in module directory
34 CMAKE_POLICY(SET CMP0017 NEW) # https://cmake.org/cmake/help/v3.0/policy/CMP0017.html
35 # Do not use export_library_dependencies() anymore
36 IF(POLICY CMP0033)
37 CMAKE_POLICY(SET CMP0033 NEW) # https://cmake.org/cmake/help/v3.0/policy/CMP0033.html
38 ENDIF()
39
40 # Load modules from our source tree too
41 SET(CMAKE_MODULE_PATH "${CMAKE_CURRENT_SOURCE_DIR}/cmake/modules")
42
43
44 # Build the main executable. Do not build the main executable.
45
```

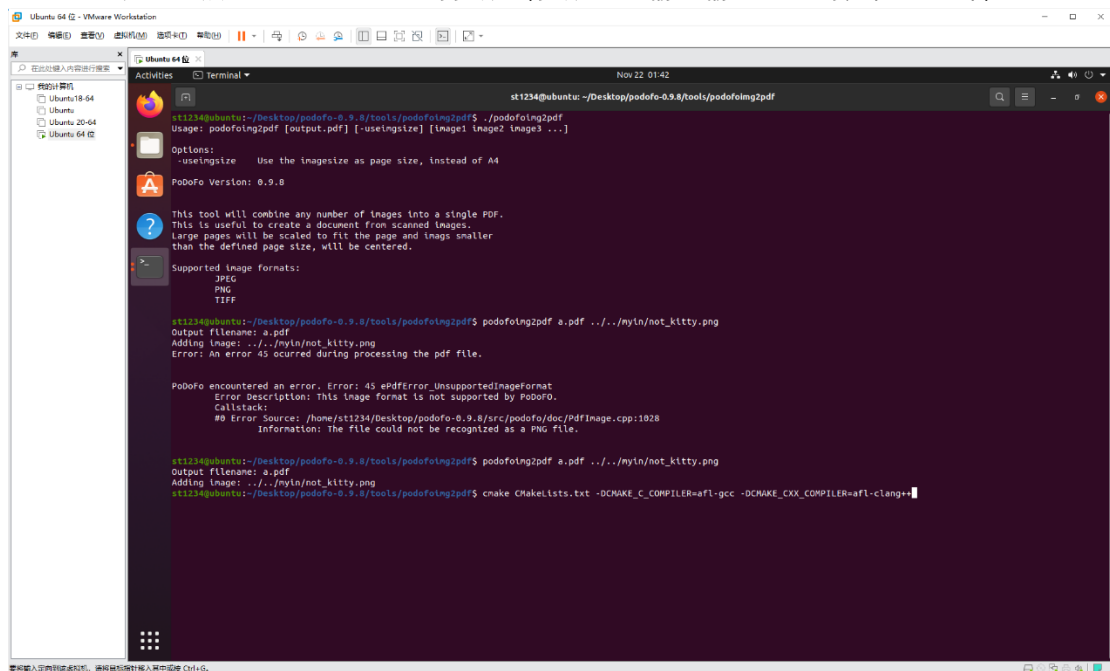
括号里面是什么依据错误而异

3.make 成功后，解决无参数的参数问题



要安装人定内到该文件夹，请按照以下步骤输入其中内容 Ctrl+G

我们有很多个装有可执行文件的文件夹，进入文件夹打开终端输入可执行文件名字，便可以看到该可执行文件对应的参数（参数没有输入输出文件的，优先选择）



要安装人定内到该文件夹，请按照以下步骤输入其中内容 Ctrl+G