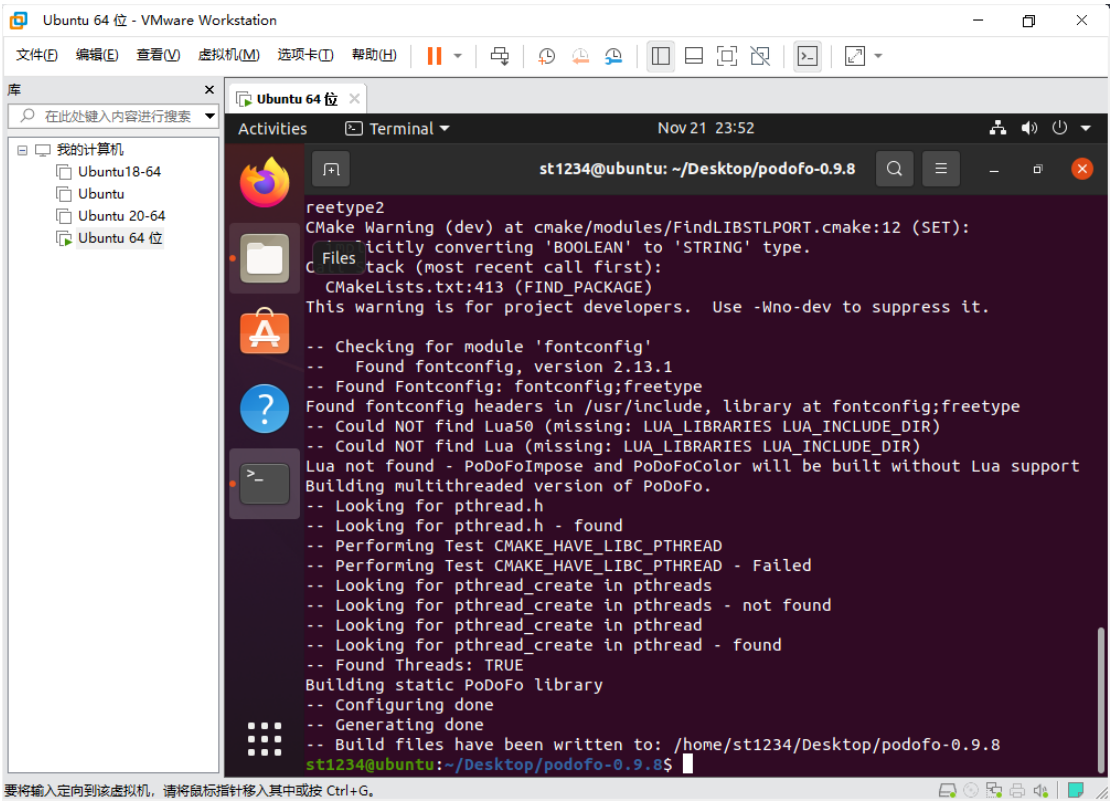
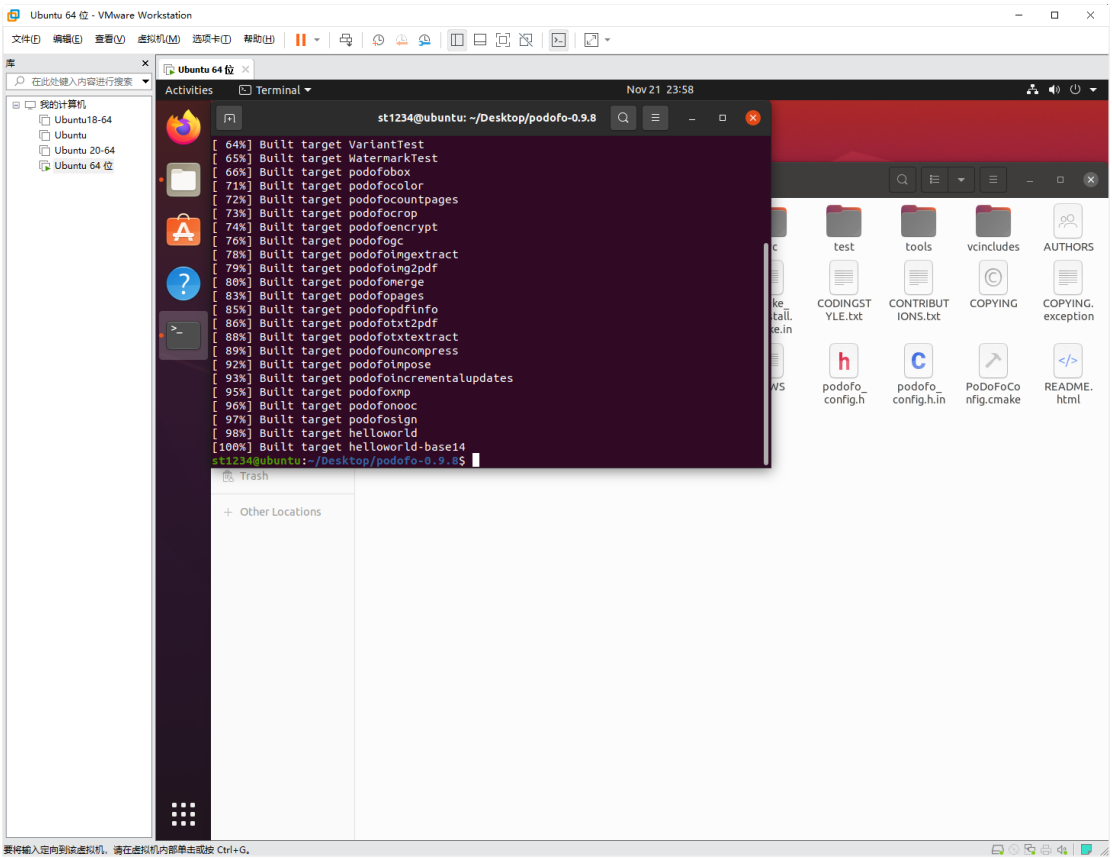
以 podofo 为例

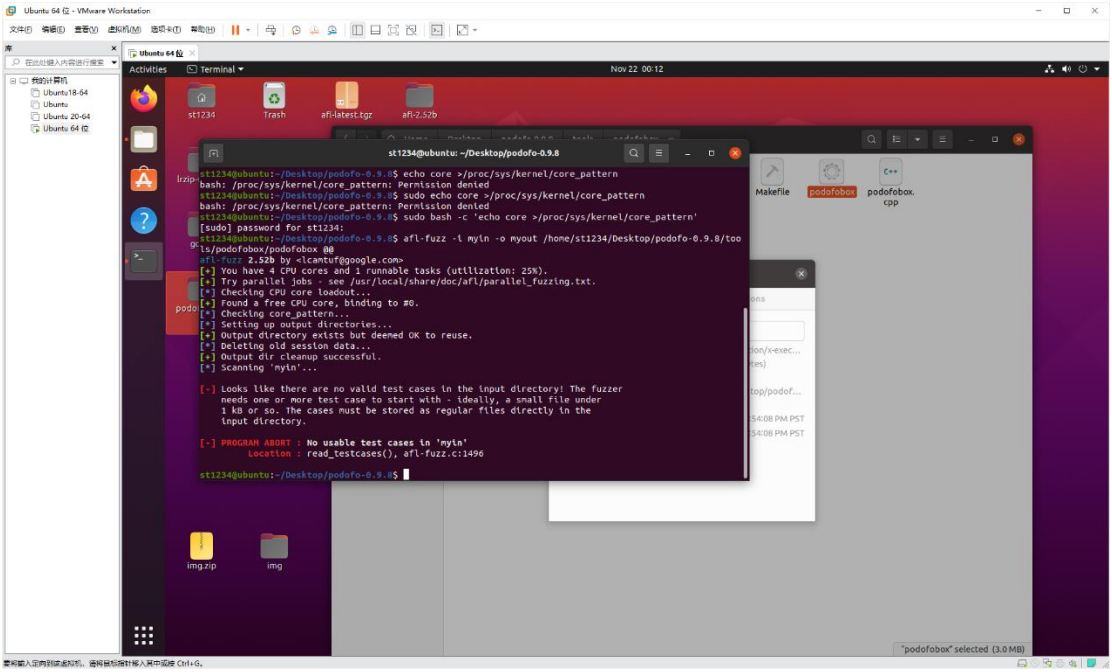
教程建立在能够用传统 gcc 编译的基础上（即你的依赖都已经成功安装）





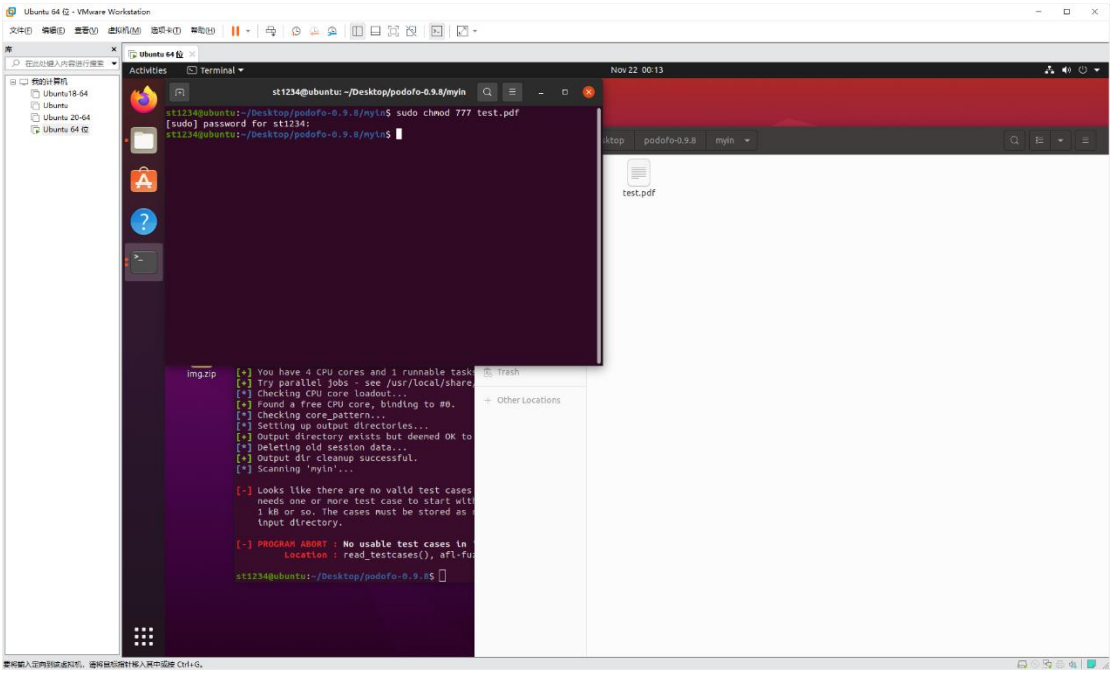
一、使用 afl 前，遇到的一些 bug

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



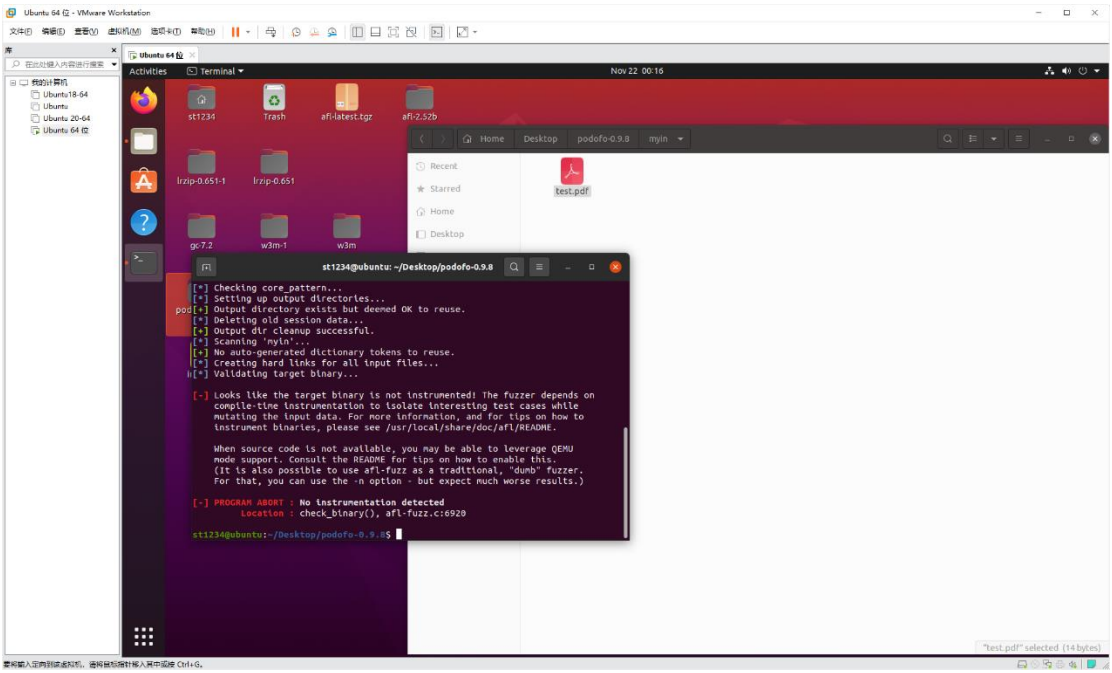
解决方法：myin 中此时已经有了测试用例，但是是完全空的，要换成一个不那么空的测试用例

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!

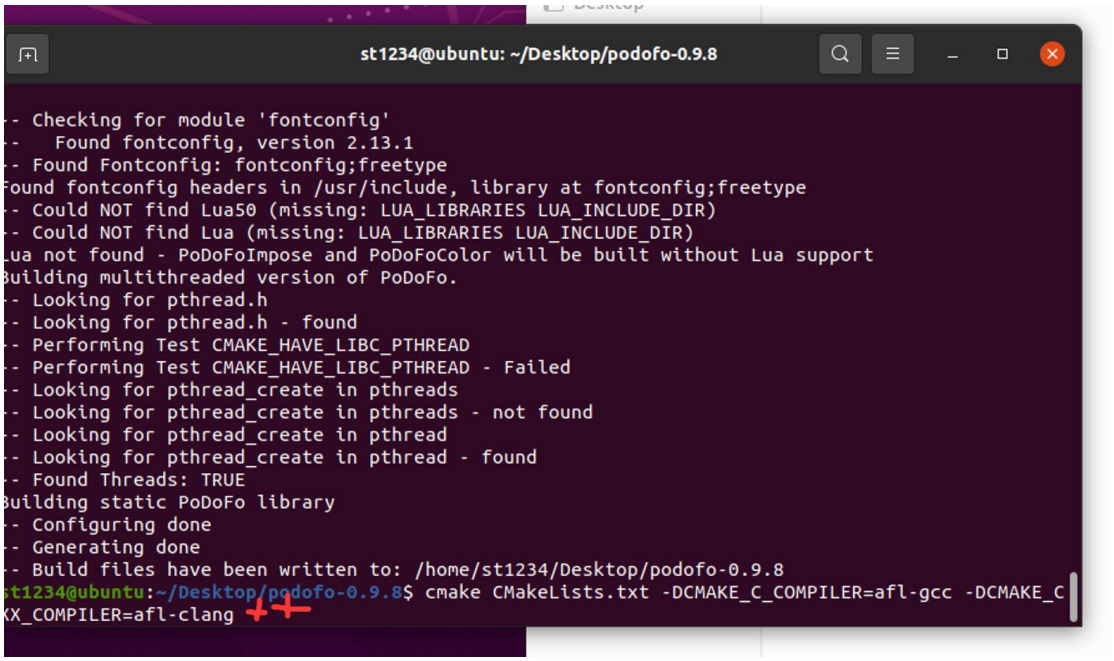


这时的问题就是你没有用 afl-gcc 编译，导致出现了用例正确但是找不到的问题。

解决方法：详见二、

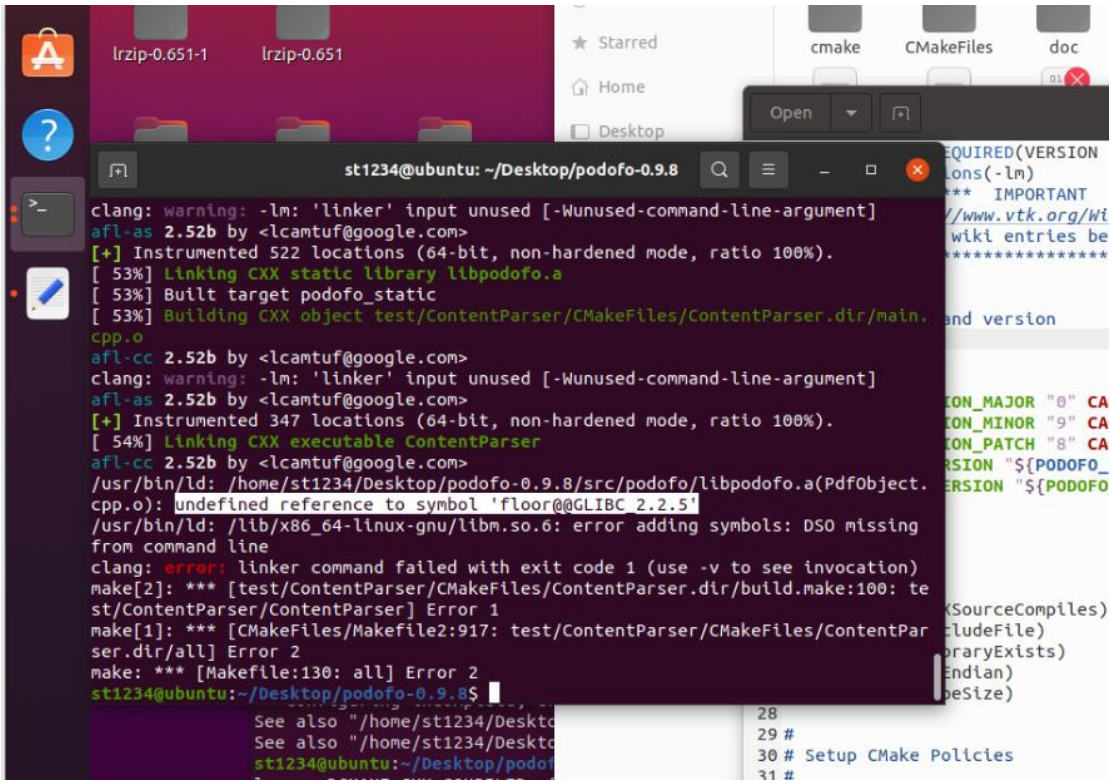
二、只有 cmake 怎么使用 afl?

1.参考



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

2.make clean all 时报错



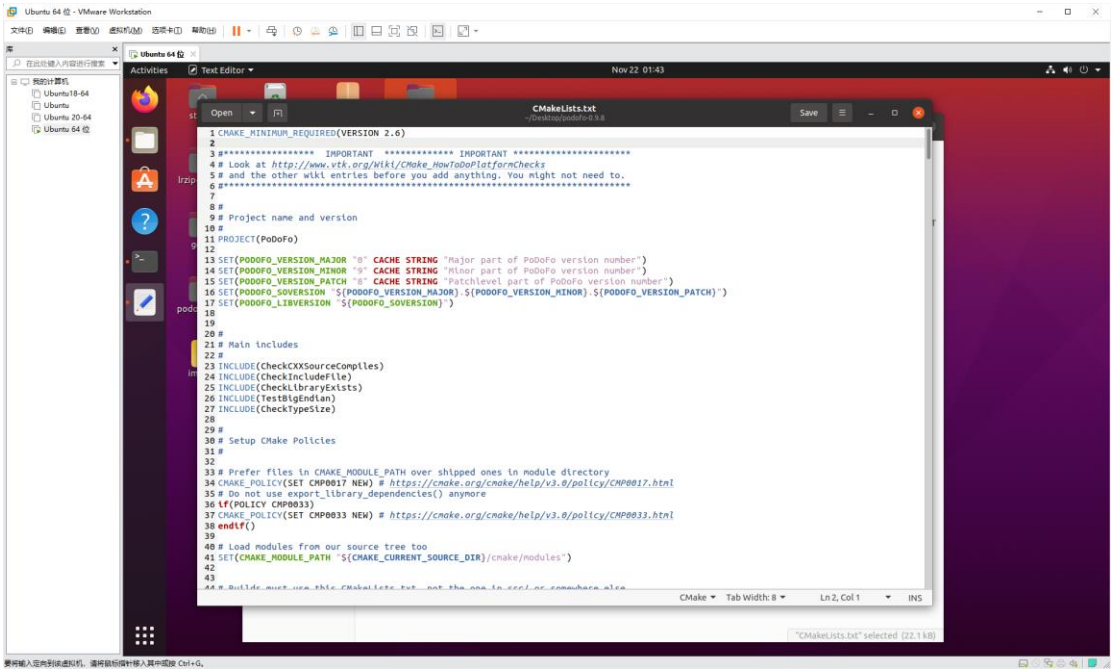
对标示部分进行搜索，比如这个的结果是因为 gcc 编译时缺少 -lm 选项

解决方法:

打开 CMakeLists.txt

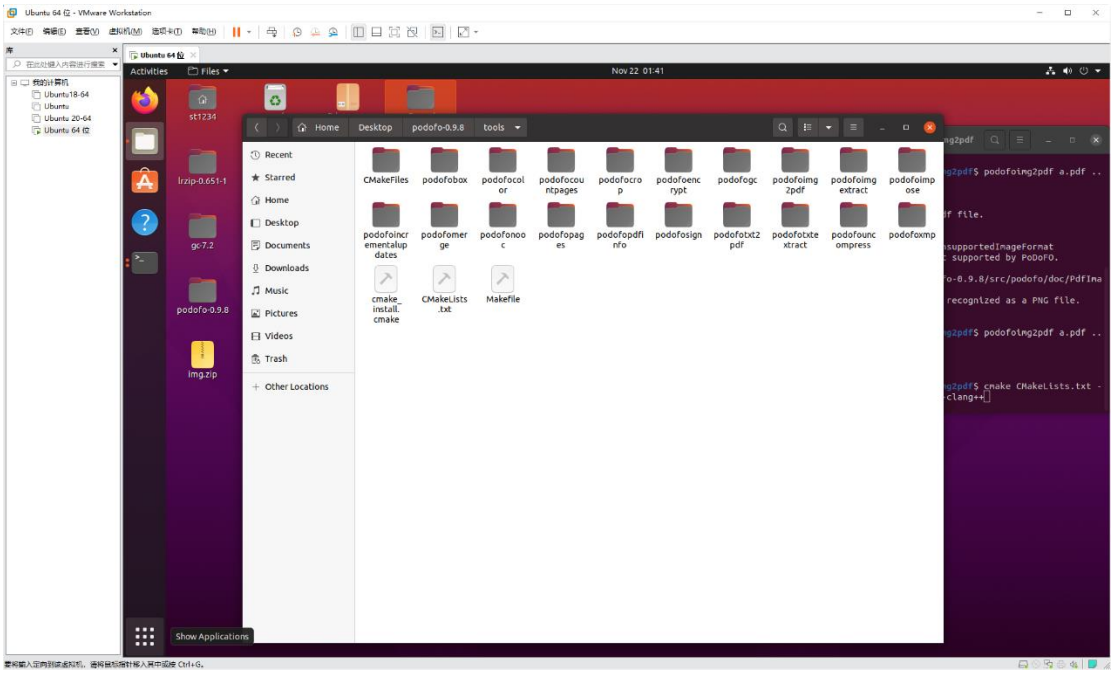
在第二行输入

add_compile_options(-lm)



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



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）