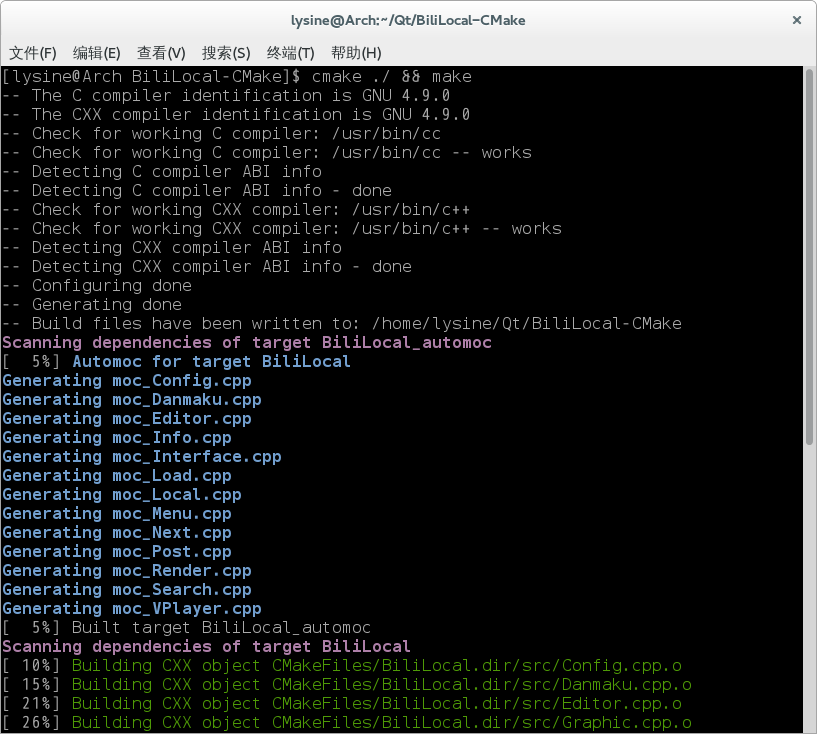
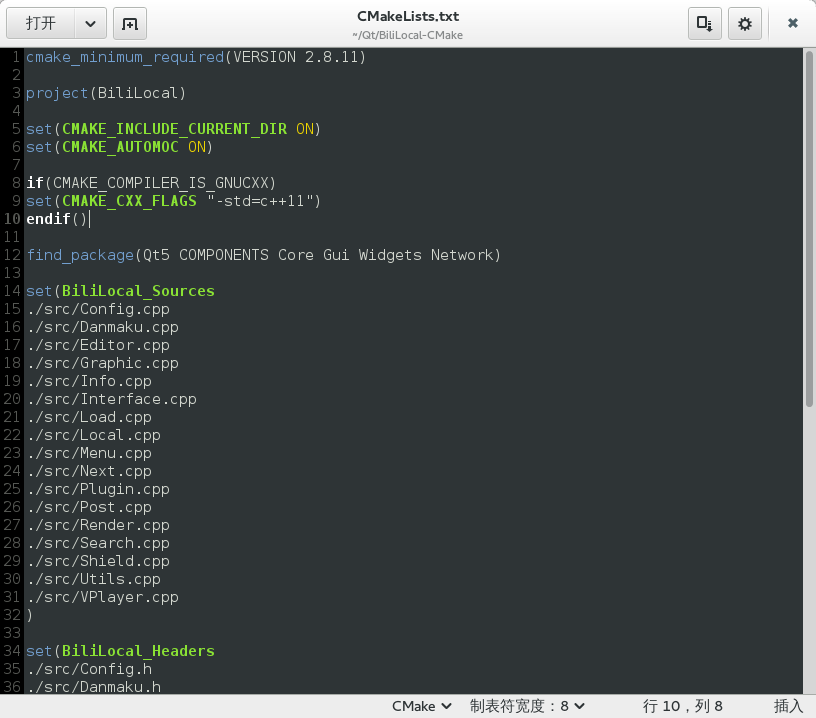
技术短文

baraplayer作为一款跨平台的播放器，虽然处于性能和兼容性考虑自己实现了如xml/json解析、shadow/glow/stroke特效等小算法，但是在诸如视频解码、图像缩放、ssl加密链接、用户界面等方面还是无可避免的使用了大量的第三方跨平台库，这些库在不同的平台上又有不同的依赖关系，这给baraplayer在不同平台上的构建带来了很大的困难。最终我们决定将项目转移到cmake构建系统。

我们认为cmake的核心优势在于他要求每个包自行管理自己的依赖关系，存储在独立的文件中，这使得通过不同参数构建出的不同依赖的库可以使用统一的CMakeLists.txt不需要任何改动，链接一个库也只需要add对应的package，而在使用vs等ide进行构建的过程中则需要针对不同的平台手动填写所有依赖的库以及这些依赖库的依赖库……

编写CMakeLists使用的语法近似于shell，cmake提供了大量的预设函数和变量对编译选项进行调整，加入所有的依赖关系和源文件后使用cmake命令即可生成Makefile或.sln文件，之后再调用编译器进行编译即可得到最终的可执行文件。



相对于更加复杂的且必须运行在类unix环境下的autotools，cmake的跨平台性更加强大、编写更加简单。但是由于使用的库必需要自带描述文件，封闭平台上的很多库的编写又根本不考虑cmake，很多时候都需要手工补全描述，cmake的实用性被大大削弱。