cxx-clean-include原理简要说明

洪坤安([wuzili1234@163.com](mailto:wuzili1234@163.com)) 2016年04月19日

cxx-clean-include基于llvm+clang库开发（其中clang是xcode的编译器），相当于clang的一个可执行的插件。

基本原理是先利用clang库的功能分析出每个cpp文件的语法树，再根据语法树统计每个cpp、h文件对其他h文件的依赖，最后统计出哪些#include无效、可用前置声明替换、可被更深层次的#include替换。

clang本身完整支持c++11，且兼容大部分msvc的语法。但是仍有一部分的msvc语法无法支持（比如：引用函数返回的局部变量、定义名称为not的变量如int not;等）。碰到这种语法时，会产生编译错误，相应地也无法获得整个文件的完整语法树。对于这种文件，本工具就不进行清理了。

其中对每个cpp文件的分析原理如下（此外还有较多细节未一一说明，但下面已经涵盖了基本分析流程）：

1. 清除多余的#include（这一步最简单）

以下图为例



上图表示，虽然[主文件.cpp]包含了很多文件，但只用到了B2.h、D2.h、E1.h头文件

处理办法：保留有用文件的所有祖先文件

处理的结果如下图



可以看出，B2.h、D2.h、E1.h的所有祖先文件均被保留下来

1. 采用前置声明替换#include

以下图为例



上图表示，文件B2中的类或结构体仅被C1用来声明指针或引用，除此之外B2文件中的内容根本不被其他文件关心。

处理办法：找到包含B2的那条#include语句，直接把那条语句替换成前置声明。

比如，假设B2.h中定义了class B2\_Class{ int n; }，则把B1文件中的#include “B2.h”语句替换成class B2\_Class;前置声明语句

1. 深层#include取代浅层#include

以下图为例



上图表示，虽然[主文件.cpp]包含了很多文件，但只用到了B2.h、D2.h、E2.h、F2.h头文件

处理办法：对于每个头文件，若其可被后代取代，则进行替换

处理之后的结果如下图



如上图，可以看到：

#include “B.h”被替换成了#include “B2.h”

#include “C.h”被替换成了#include “D1.h”