二次开发详细设计报告

HearthStone Team

**1 cppcheck简介**

cppcheck是一种C/C++代码缺陷静态检查工具。不同于C/C++编译器及其它分析工具，cppcheck只检查编译器检查不出来的bug，不检查语法错误。cppcheck通过自身具有的功能对C/C++语言代码进行错误检查。

**1.1 cppcheck检查功能**

通过查阅资料[1]，我们了解到cppcheck具有如下的检查功能：

1. 自动变量检查

2. 数组的边界检查

3. class类检查

4. 过期的函数，废弃函数调用检查

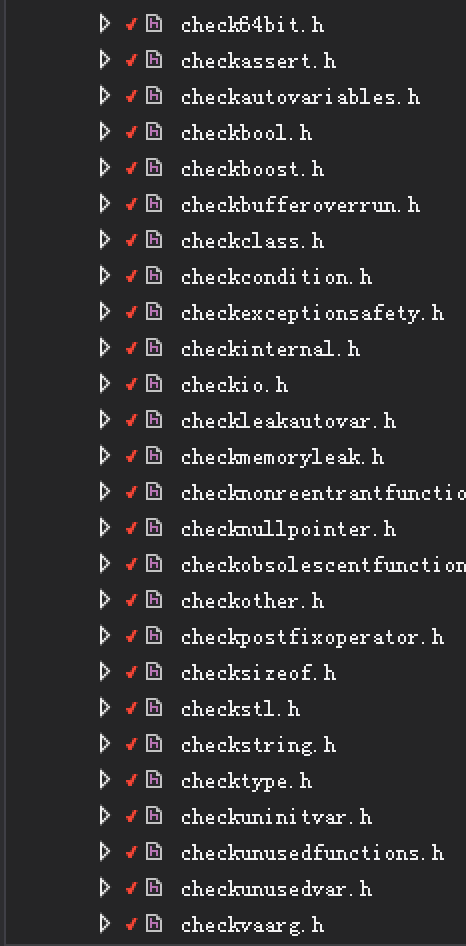
5. 异常内存使用，释放检查

6. 内存泄漏检查，主要是通过内存引用指针

7. 操作系统资源释放检查，中断，文件描述符等

8. 异常STL 函数使用检查

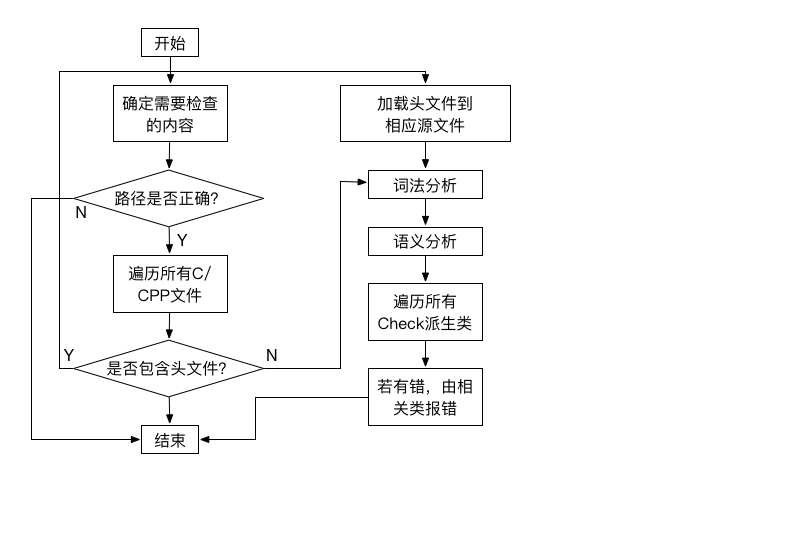
9. 代码格式错误，以及性能因素检查

 左图为cppcheck项目“检查”类的目录，即上述“不同派生类”的具体目录，它们具有不同的对代码进行分析的功能。此外，cppcheck还留出了扩充接口，可以在现有的代码基础上进行二次开发，进一步提高缺陷检查能力，这也是本次开发的基础。通过对代码的分析，我们详细的了解了cppcheck各个模块的功能，这也对我们进行二次开发提供了很大的帮助。当然，在这里我们不会过多的关注已有功能，而是对我们要开发的方向进行反复的测试，决定出我们要开发的方向，攻克进行二次开发可能遇到的技术难题，完善cppcheck，使其具有更强大的检查功能。

**1.2 cppcheck检查流程**

cppcheck内置完善的检查类，CppCheckExecutor类作为整个程序的接入，进入检查阶段，CppCheck类处理单个文件，加入预定义宏使用Preprocessor预处理源文件，去除注释及宏命令和无效代码等，在文件内容前添加文件自身信息，一同进行tokenize，Token模块负责语意划分，进入CppCheck::checkFile方法进行主要处理工作，SymbolDatabase模块负责缓存语义的定义域，建立SymbolDatabase符号库，建立运算符关系，将运算符操作数与运算符token链接 以供定义域内按照名称索引语义元素并进行匹配，最后使用所有Check规则对现有的token进行测试分析，不同派生类执行不同的检查工作，将检查出的错误交给ErrorLogger统计并输出。

**1.3 cppcheck处理流程图**

****

通过对代码的分析、黑箱测试、白盒测试、以及参考了一些文档[2]，我们总结出了如上图的处理流程，首先程序要求待处理项目的目录，再依次遍历待处理的源文件，并且在有头文件时智能的进行匹配，从而实现了对整个项目目录下所有源文件中缺陷的检测与报错，方便使用者调试。

**1.4 cppcheck涉及到的设计模式**

cppcheck使用。

**2 cppcheck二次开发**

这里是内容

**2.1 缺陷分析**

这里是内容

**2.2 缺陷模式添加**

这里是内容

**3 cppcheck二次开发结果测试**

这里是内容

**参考文献**

[1] Cppcheck百度百科. <http://baike.baidu.com/link?url=LQR9Zfwn8mKnkZwzJH95o9TyYoh7BkZ-y5vrqsu4Yd4ThtjIsC-I4EUN-1oxdF9E4QgXhZepjcqhiapLQ2OZo_>

[2] cppcheck的整体架构. <http://wenku.baidu.com/view/7d9804c7bb4cf7ec4afed0aa.html>

[3]张仕金，尚赵伟 Cppcheck的软件缺陷模式分析与定位[A].重庆:重庆大学,2015,51(3):69-73