C/C+程序检查工具调研

1. 常见静态检查工具

* **Cppcheck:** **(开源)**

**介绍**：cppcheck是一款历史悠久的小型C++源码静态检查工具，可以嵌入到VS编译工具，也可以再CMD命令行下操作静态检查源码，可以查找出可能的空指针、成员变量未初始化这些警告，还有一些风格上的校验，C类风格的写法也会做提醒，用户根据检查结果和自身的需要进行修改。不同于 C/C++ 编译器及很多其它分析工具，它不检查代码中的语法错误。Cppcheck 只检查编译器检查不出来的 bug 类型。

**检查信息**：

可检测的问题包括：

* Dead pointers
* Division by zero
* Integer overflows
* Invalid bit shift operands
* Invalid conversions
* Invalid usage of STL
* Memory management
* Null pointer dereferences
* Out of bounds checking
* Uninitialized variables
* Writing const data

并将问题分为以下6类：

错误（error）：bug。

警告（warning）：预防性编程方面的建议。

风格警告（style）：出于对代码简洁性的考虑（函数未使用、冗余代码等）。

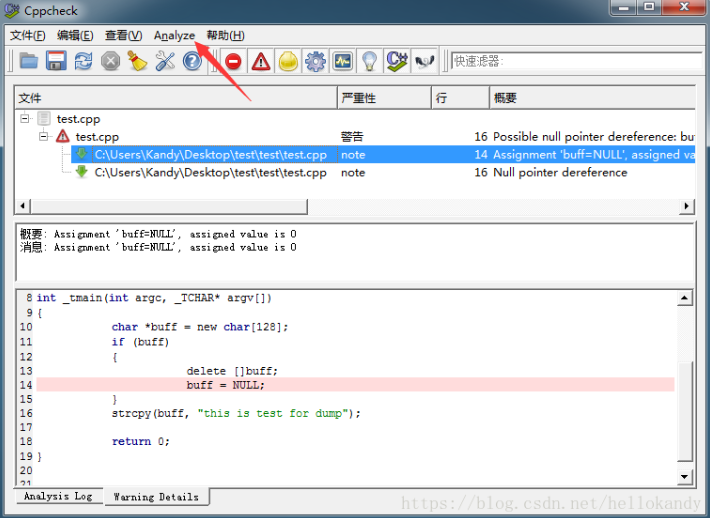
可移植性警告（portability）：64/32位可移植性、编译器通用性等。

性能警告（performance）：使代码更高效的建议，但不保证一定有明显效果。

信息消息（information）：条件编译方面的警告。

**版本信息**：支持windows版本和Linux版本

安装十分简便，只需在官网下载最新的可执行安装包（本文目前为cppcheck-1.83-x86-Setup.msi）跟着向导「下一步」即可。



Linux版本下载地址：<https://sourceforge.net/projects/cppcheck/files/cppcheck/>

Window版本下载地址：<http://cppcheck.sourceforge.net/>

**使用教程**：<https://blog.csdn.net/hanxv_1987/article/details/77803837>

<https://blog.csdn.net/akof1314/article/details/7477014>

<https://www.cnblogs.com/lvdongjie/p/4187144.html>

* [**TscanCode**](http://code.tencent.com/tscancode.html)**: (开源)**

**介绍:** TscanCode是腾讯的开源项目，为此次调研的唯一一款本土工具，起初构建于Cppcheck的基础之上，后来进行了重新实现，并加入了对C#和Lua的支持。

**检查信息**：

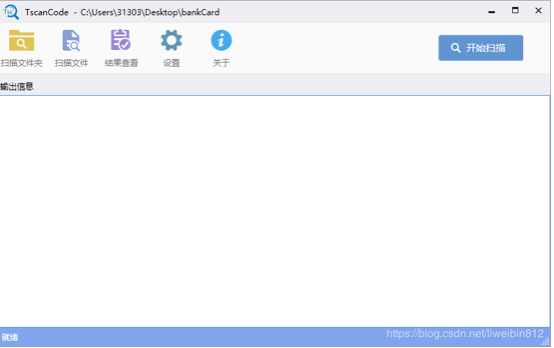
可检测的问题包括:

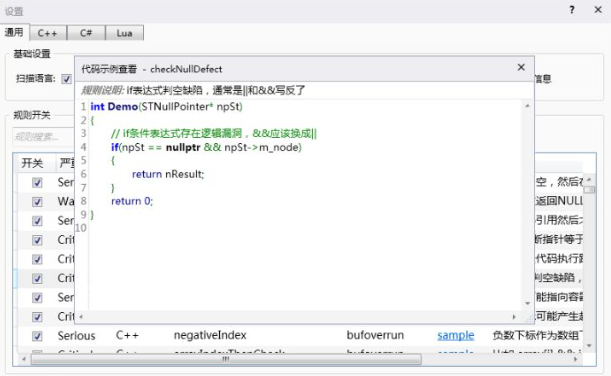
* 空指针检查，包含可疑的空指针，判空后解引用比如Crash等共3类subid检查
* 数据越界，Sprintf\_S越界共1类subid检查
* 内存泄漏，分配和释放不匹配同1类subid检查
* 逻辑错误，重复的代码分支，bool类型和INT进行比较，表达式永远True或者false等共18类检查
* 可疑代码检查，if判断中含有可疑的=号，自由变量返回局部变量等共计15类检查
* 运算错误，判断无符号数小于0,对bool类型进行++自增等，共计11类检查

并将问题分为致命、严重、警告、提示、风格5类。

**版本信息：**支持Windows、Linux和Mac OS

安装同样便捷，下载安装包（本文目前为TscanCodeV2.14.24.windows.exe）跟着向导「下一步」即可。





下载地址：<https://github.com/Tencent/TscanCode>

**测试结果：**<https://github.com/Tencent/TscanCode/tree/master/samples/cpp>

**使用教程：**https://blog.csdn.net/liweibin812/article/details/84544750

* **Cpplint: (开源)**

**介绍**：cpplint 是google使用python脚本实现的用于检查C++代码风格的工具。使用的是Google C++代码规范。如果是遵循google code style的，可以使用cpplint作为代码规范的一个检查工具。

google C++风格：https://blog.csdn.net/shendl/article/details/6762538

**源码下载**：<https://github.com/google/styleguide/blob/gh-pages/cpplint/cpplint.py> (python2.7)

**使用教程**：<https://blog.csdn.net/yelu_hong/article/details/83856306>

https://blog.csdn.net/fengbingchun/article/details/47341765

* **PC-Lint: (付费)**

**介绍**：PC-Lint是一个历史悠久，功能异常强劲的静态代码检测工具。它不但能够监测出许多语法逻辑上的隐患，而且也能够有效地帮你提出许多程序在空间利用、运行效率上的改进点，对于小公司和个人开发而言，PC-Lint也非常重要，因为基于开发成本考虑，小公司和个人往往不能拿出很多很全面的测试，这时候，PC-Lint的强劲功能可以很好地提高软件的质量。

**功能**：

1) PC-lint不但可以检测单个文件，也可以从整个项目的角度来检测问题，因为C语言编译器固有的单个编译，这些问题在编译器环境下很难被检测，而PC-Lint在检查当前文件的同时还会检查所有与之相关的文件。

2) PC-lint支持几乎所有流行的编辑环境和编译器，比如Borland C++从1.x到5.x各个版本、Borland C++ Build、GCC、VC，VC.net、watcom C/C++、Source insight、intel C/C++等等。

3) 支持Scott Meyes的名著（Effective C++/More Effective C++）中说描述的各种提高效率和防止错误的方法。

4) 错误信息编号：

C C++

告警级别语法错误(Syntax Errors) 1 - 199 1001 - 1199 1

内部错误(Internal Errors) 200 - 299 0

致命错误(Fatal Errors) 300 - 399 0

告警(Warnings) 400 - 699 1400 - 1699 2

提示(Informational) 700 - 899 1700 - 1899 3

可选信息(Elective Notes) 900 - 999 1900 - 1999 4

对于C语言，1~199是与语法错误；200~299是PC-lint内部错误，应该决不会发生的；300~399是致命错误，通常是由于超越了某些限制；400~699是警告消息，提示被检查的程序中可能存在错误；700~899是提示信息，这些提示信息可能有错误，也可能是合法的程序，取决于个人的编程风格；900~999则是一些称为可选信息，一般不会自动输出。

**版本信息**：主要应用于windows平台

**下载地址**：https://www.gimpel.com/html/index.htm

**使用教程**：<https://blog.csdn.net/kingofbirdzjy/article/details/1727835>

<https://blog.csdn.net/scucj/article/details/4409735>

<https://blog.csdn.net/TrueZq/article/details/652104>

**SpecChecker: (付费, 北京轩宇信息有限公司)**

**介绍**：SpecChecker是北京轩宇信息技术有限公司自主研发的静态分析工具，采用Java语言，在中国航天系统广泛应用，替代了国外同类商业工具。支持安全编码标准符合性检查、运行时缺陷检测和代码质量度量等。

**功能：**

1) 编码标准符合性检查：支持300+条C++规范、700+条C规范和本地化规则，对违背规则的语句会在结果中给出，并指出违背的规则项。

2) 代码度量分析：支持代码行规模类、圈复杂度、逻辑复杂度和预处理类等50余项代码度量项。

3) 软件质量度量。软件质量度量的分析依赖标准规则检查和代码度量分析。衡量的指标有：可靠性、可维护性、可移植性、效率性。试用版本上无软件质量度量功能。自动生成的报告中虽有质量度量的描述，但质量度量的分析图不可见。

**检查信息：**

可检测的问题包括:

* 除零(CWE-369)
* 数组越界(CWE-129)
* 缓冲区溢出(CWE-121/122/124)
* 内存泄漏(CWE-401)
* 内存拷贝未检查长度(CWE-120)
* 空指针解引用(CWE-467)
* 变量读取前未初始化(CWE-453)
* 内存释放后使用(CWE-416)
* 返回栈地址(CWE-562)
* 释放非堆内存(CWE-590)

**版本信息：**主要应用于windows平台



静态检测工具对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **工具** | **语言** | **平台** | **授权** | **优劣** |
| [Cppcheck](http://cppcheck.sourceforge.net/) | C, C++ | Windows,Linux | 开源 | 界面简单，检测准确 |
| [TscanCode](http://code.tencent.com/tscancode.html) | C,C++,C#,Lua | Windows,Linux,Mac | 开源 | 界面简单，易用性强 |
| Cpplint | C, C++ | Windows,Linux | 开源 | 规范受限，友好型差 |
| [PC-Lint](http://www.gimpel.com/html/pcl.htm) | C, C++ | Windows | 付费 | 传统较准确，费用低 |
| SpecChecker | C, C++ | Windows | 付费 | GUI人性化，分析速度快  分析详细，可生成报告 |
|  |  |  |  |  |
|  |  |  |  |  |

**[参考博客]**

1. [C/C++代码静态分析工具调研](https://www.jianshu.com/p/92886d979401)