




方舟编译器的静态 代码检查

PLCT实验室 史宁宁
2020-03-01

方舟编译器的静态代码检查

- 1、方舟编译器的静态代码检查，主要使用了clang-tidy进行检查。
- 2、方舟编译器源码主目录下，可以看到clang-tidy的检查规则配置文件.clang-tidy:

 .clang-tidy	Update the .clang-tidy to add check rules	13天前
 .gitignore	Clean the out directory in doc and gitignore	2天前
 .gn	add lcn headers in gn & script, modify output path	5天前

.clang-tidy的提交记录

2020-02-17 (1)

d052a3a Update the .clang-tidy to add check rules


浏览代码

 shining 提交于 2020-02-17 17:01

2020-02-14 (1)

199ca90 Update the .clang-tidy to add cast rules

浏览代码

 shining 提交于 2020-02-14 11:37

2020-02-12 (1)

914d0eb Update the .clang-tidy to add rules

浏览代码

 shining 提交于 2020-02-12 17:09

2020-02-11 (1)

df0cee7 Update the .clang-tidy to add rules

浏览代码

 shining 提交于 2020-02-11 16:50

2020-02-07 (1)

7711ccb Add clang-tidy config file

浏览代码

 shining 提交于 2020-02-07 12:55

clang-tidy

- 1、clang-tidy是clang的extral tool，它是基于LibTooling的工具。而LibTooling是一个库，这个库主要是为了基于Clang编写单独的工具。
- 2、clang-tidy是基于AST的静态检查工具。因为它基于AST,所以要比基于正则表达式的静态检查工具更为精准，但是带来的缺点就是要比基于正则表达式的静态检查工具慢一点。也是因为它基于AST,所以clang-tidy运行的时候需要知道编译命令。
- 3、clang-tidy通过添加check来添加检查规则。

clang-tidy (续)

4、clang-tidy每次针对一个TU，即单个cpp文件，无法跨TU处理。但是，可以使用clang-tidy/tool/源码目录之下的run-clang-tidy.py去批量检查文件。

5、clang-tidy因为需要知道编译命令，所以必须通过compile_commands.json获取编译命令，或者通过在clang-tidy后通过--添加编译选项。

方舟编译器的clang-tidy使用文档 (Developer_Guide)

✎ 源码静态检查

本部分内容将指导您使用clang-tidy进行源码静态检查。在对源码进行修改之后，对源码进行静态检查，可以检查源码是否符合编程规范，有效的提高代码质量。

静态源码检查之前，需要先编译出OpenArkCompiler。此后，以检查src/maple_driver源码为例，在openarkcompiler目录下执行以下命令：

```
cp output/compile_commands.json ./
./tools/clang_llvm-8.0.0-x86_64-linux-gnu-ubuntu-16.04/share/clang/run-clang-tidy.py -clang-tidy-binary='./tools/clang_llvm-
```

命令说明：

- `cp output/compile_commands.json ./` 将output目录之下的compile_commands.json复制到当前目录之下，它是clang-tidy运行所需要的编译命令；
- `./tools/clang_llvm-8.0.0-x86_64-linux-gnu-ubuntu-16.04/share/clang/run-clang-tidy.py` 调用clang-tidy进行批量检查的脚本run-clang-tidy.py，其中 `./tools/clang_llvm-8.0.0-x86_64-linux-gnu-ubuntu-16.04/` 目录是之前配置的clang编译器的发行包主目录； `-clang-tidy-binary` 是指明clang-tidy的具体位置； `-clang-apply-replacements-binary` 是指明run-clang-tidy.py所依赖的clang-apply-replacements的位置； `src/maple_driver/` 是要进行源码检查的目录。

clang-tidy文档

- 1、官方文档: <http://clang.llvm.org/extra/clang-tidy/>
- 2、深入研究Clang(十三) clang-tidy简介:
<https://zhuanlan.zhihu.com/p/102248131>
- 3、深入研究Clang(十四) clang-tidy的使用:
<https://zhuanlan.zhihu.com/p/105703209>

求关注！ 求四连！