

PVS-Studio

Информация и лицензия

PVS-Studio — это инструмент для статического анализа исходного кода программ, написанных на языках C, C++.

Для использования в Linux нужно чтобы в каталоге `~/.config/PVS-Studio` находился лицензионный ключ или в начале исходных файлов с расширениями `.c`, `.cpp`, `*.cxx` и т.п. присутствовали строки

```
// This is an independent project of an individual developer. Dear PVS-Studio, please check it.
```

```
// PVS-Studio Static Code Analyzer for C, C++ and C#: http://www.viva64.com
```

Настройка и компиляция проекта

Полное руководство для работы в Linux находится [здесь](#), а ниже приведён список типовых команд.

Настройка проекта для CMake:

```
cmake -DCMAKE_EXPORT_COMPILE_COMMANDS=On <src-tree-root>
```

Сборка проекта с помощью `make`:

```
pvs-studio-analyzer trace -- make -j$(nproc)
```

Настройка и сборка проекта с использованием Ninja:

```
cmake -GNinja -DCMAKE_EXPORT_COMPILE_COMMANDS=On <src-tree-root>  
ninja -t compdb
```

Анализ проекта

Для анализа проектов используется утилита `pvs-studio-analyzer`. По умолчанию включены только диагностики общего назначения (General Analysis, GA). Включить дополнительные правила можно с помощью опции `-a`:

```
-a [MODE], --analysis-mode [MODE]
  MODE defines the type of warnings:
  1 - 64-bit errors;
  2 - reserved;
  4 - General Analysis;
  8 - Micro-optimizations;
  16 - Customers Specific Requests;
  32 - MISRA.
  Modes can be combined by adding the values
  Default: 4
```

Выбор типов предупреждений осуществляется на основе побитовой маски из приведённых выше типов. Чтобы выполнить анализ, исключив проверки MISRA, нужно выполнить

```
pvs-studio-analyzer analyze -a 29 -j$(nproc) -o pvs.log
```

Составление отчёта для просмотра в Qt Creator:

```
plog-converter -t tasklist -o pvs.tasks pvs.log
```

При составлении отчёта уровень детализации можно изменять с помощью ключа `-a` (`--analyzer`), который производит фильтрацию предупреждений согласно маске, которая имеет вид `MessageType:MessageLevels`, где `MessageType` может принимать один из следующих типов: `GA`, `OP`, `64`, `CS`, `MISRA`, а `MessageLevels` может принимать значения от `1` до `3`. Возможна комбинация разных масок через `;`. Например, при составлении отчёта для просмотра в формате HTML можно повысить уровень детализации:

```
plog-converter -t fullhtml -a "GA:1,2,3;64:1;OP:1,2;CS:1,2" -o html pvs.log
```

Составление отчёта для редактора Vim:

```
plog-converter -a "GA:1,2" -t errorfile -o pvs.err pvs.log
```

Для просмотра отчёта в редакторе Vim нужно открыть файл `pvs.err` и выполнить команды:

```
:set makeprg=cat\ % " выбор программы для компиляции в текущей сессии
:silent make " имитация сборки
:cw " открытие результатов в нижнем окне
```