



Технологии QNX и КПДА

в России

Санкт-Петербург, 23 ноября 2016

Выявление программных ошибок при помощи инструментов статического анализа

Игорь Рондарев, ООО «СВД Встраиваемые Системы»

- Статический анализ — что это?
- Инструменты статического анализа
- Пример использования
- Пример интеграции в среду разработки

- «Статический анализ кода (англ. static code analysis) — анализ программного обеспечения, производимый (в отличие от динамического анализа) без реального выполнения исследуемых программ.»*
 - позволяет выявлять ошибки, трудно поддающиеся локализации с помощью других методик (отладчиков, динамических анализаторов и т.д.)
 - позволяет осуществлять проверку на соответствие стандартам безопасного программирования
 - Помогает по-новому взглянуть на разработанные ранее приложения, в том числе на используемые приёмы и привычки программирования.

* [http://ru.wikipedia.org/Статический анализ кода](http://ru.wikipedia.org/Статический_анализ_кода)

Пример (базовый статический анализ)

Исходный код (фрагмент):

```
int main(void) {
    char a;
    char array[16];

    /* some long-running-code WITH NO INITIALIZATION */

    printf("result = %d\n", a + array[8]);

    return EXIT_SUCCESS;
}
```

Вывод компилятора (gcc -Wall -Wextra):

```
static_analysis.c:10: warning: 'a' is used uninitialized in this
function
```

Вывод приложения:

```
result = 42
```

Пример (базовый статический анализ)

Исходный код (фрагмент):

```
int main(void) {
    char a;
    char array[16];

    /* some long-running-code WITH NO INITIALIZATION */

    printf("result = %d\n", a + array[8]);

    memset(array, 16, 0xFF); return EXIT_SUCCESS;
}
```

Вывод компилятора (gcc -Wall -Wextra):

```
static_analysis.c:10: warning: 'a' is used uninitialized in this
function
```

Вывод приложения:

```
result = 42
Segmentation fault
```

Пример (базовый статический анализ)

Исходный код (фрагмент):

```
int main(void) {
    char a;
    char array[16];

    /* some long-running-code WITH NO INITIALIZATION */

    printf("result = %d\n", a + array[8]);

    memset(array, 0xFF, 16); return EXIT_SUCCESS; /* Correct! */
}
```

Вывод компилятора (gcc -Wall -Wextra):

```
static_analysis.c:10: warning: 'a' is used uninitialized in this
function
```

Вывод приложения:

```
result = 42
```

Пример (базовый статический анализ)

- Алгоритмические ошибки

- «*off-by-one*»

- `char a[4]; int i; for(i=0; i<=4; i++) {...}`

- Синтаксические ошибки

- `memset(dst, val, len) <=> memset(dst, len, val)`

- **Смысловые ошибки**

- Ошибки «Copy-Paste»

- `if(obj.x == x || obj.y == y || obj.z == y) {...}`

- Неверные функции

- Класс `std::string`

- `str.empty()` ВМЕСТО `str.clear()`

- И т.д.

- Свободные/открытые:
 - **cppcheck, cpp lint, splint, Frama-C** и т. д.
- Коммерческие:
 - **Coverity, LDRA TestBed, PC-Lint, PVS-Studio** и т.д.

- Продукт с открытым исходным кодом
 - Возможность использования на разных платформах
- Гибкая настройка
 - Используемый стандарт языка
 - **POSIX, C99, C++11** и т.д.
 - Используемые библиотеки
 - **Qt, GTK, SDL** и т. д.
- Возможность запуска в многопоточном режиме
 - Сокращает время анализа больших проектов

- Варианты использования
 - **CLI (Command Line Interface)**
 - Запуск для анализа проектов или индивидуальных файлов
 - Автоматизированный запуск для построения отчётов
 - **GUI (Graphical User Interface)**
 - сppcheck-gui (входит в состав сppcheck)
 - QNX Momentics IDE
 - Qt Creator

- Анализатор **crrcheck**
 - Установка на инструментальную систему
 - Базовая настройка
 - Стандарты языка
 - Формат и содержимое вывода
 - Запуск анализатора
 - Использование **crrcheck-gui** для просмотра результатов анализа
- Средства интеграции в QNX Momentics IDE* (модуль **crrcheclipse**)
 - Установка
 - Базовая настройка
 - Использование
 - импорт проекта, запуск анализатора и просмотр результатов анализа

* рекомендуемая версия QNX Momentics IDE - 5.0 и выше

- Статический анализатор **cppcheck**
 - Проект активно развивается
 - <https://sourceforge.net/p/cppcheck/news/>
 - Поддерживаются различные инструментальные платформы
 - **Windows-based**
 - **Linux-based**
 - **QNX Self-hosted**
 - Развитый пользовательский интерфейс
 - **Интеграция с IDE**
 - Возможность использования в **системах автоматизированной сборки и тестирования ПО**
 - Подробная документация и поддержка со стороны сообщества

Спасибо за внимание

Игорь Рондарев

инженер-программист

ООО «СВД Встраиваемые Системы»

тел.: +7 (812) 346-8956

факс: +7 (812) 346-8953

<http://www.kpda.ru> | <http://forum.kpda.ru> |

<http://www.swd.ru>