

**Проектирование системы сборки кроссплатформенного компонента инсталляции навигационных карт с использованием CMake**

**Аржевитин Б.К.,**

**Научный руководитель – к.т.н, Захаров А.С.**

(АО «Вяртсиля Цифровые Технологии», Санкт-Петербург)

В работе предлагаются методы, позволяющие понятным образом описывать сборку модулей системы в CMake-файлах для сборки компонентов в крупных проектах. На основе разработанных методов описан процесс сборки компонента инсталляции карт и его тестирования с визуализацией результатов запуска тестов в системе непрерывной интеграции Jenkins. В настоящее время продукты, производимые разработчиками, включают в себя большое число различных компонентов, модулей, зависимостей – подобное особенно хорошо наблюдается в проектах крупных компаний. Так как управление процессами сборки и тестирования подобного числа компонентов вручную в сложных проектах сопряжено с большими неудобствами, появляется необходимость в разработке механизмов автоматизированного построения модулей проекта и их последующего тестирования. На сегодняшнее время наибольшее распространение получила система автоматизированной сборки проектов CMake. В крупных командах разработки обычно не все сотрудники досконально знают как устроены процессы сборки продукта, поэтому важно реализовать систему таким образом, чтобы разработчикам было удобно ее использовать без глубокого понимания деталей реализации. Кроме того, система сборки должна быть расширяемой, то есть поддерживать простое добавление новых компонентов и модулей в продукт. Отметим, что в крупных проектах применяется модульное тестирование, поэтому разрабатываемая система должна поддерживать возможность работы с тестовыми микропрограммами и анализировать результаты их работы.

При работе над продуктом разработчик сталкивается с такими проблемами, как время компиляции, организация совместной работы над проектом, грамотная структуризация исходного кода, повторное использование чужого и своего кода. Для решения подобных задач применяется подход – модульное программирование. На основе этого подхода исходный код проекта делится на части – транслируемые модули, при этом между модулями устанавливаются зависимости. Подобный процесс можно автоматизировать. Для этого применяются автоматические системы сборки проектов. Среди наиболее известных: Make, Automake, qmake, CMake, SCons.

Проведенный анализ используемой в настоящее время системы сборки компонента инсталляции показал, что для функционирования этой системе необходима поддержка кросскомпиляции под мобильные платформы, организация управления модульными тестами, кроссплатформенная сборка модулей. Кроме того, при сборке компонента инсталляции генерация проекта для интегрированных сред разработки (IDE) реализована только под Visual Studio. Это во много связано с тем, что в рассматриваемой системе механизмы сборки построены с применением SCons. Кроме того, с использованием SCons многие механизмы сборочной системы требуется реализовывать вручную, что усложнило сборочную систему. Отметим, что применение SCons также сказалось на сложности реализации механизмов сборки под различные платформы; продолжительном времени построения графа зависимостей; невозможности отделения этапа конфигурирования от этапа сборки компонента, что повлекло за собой увеличение времени сборки.

На основании отмеченных недостатков системы сборки компонента инсталляции предлагается разработать механизмы сборки с применением CMake. Причинами выбора CMake является простота синтаксиса описания cmake-файлов сборки модулей; возможности поддержки кросскомпиляции под большинство современных платформ «из коробки» или с удобными возможностями по настройке специальных программ сборки (компилятора, линкера и т.п.); кроссплатформенность системы сборки; поддержка организации модульного тестирования. Отметим, что в силу распространённости системы CMake среди крупных компаний, таких как Netflix, LLVM, MySQL, OpenCV, QGIS, Qt и др. обеспечивается высокий

уровень разработки и удобная документация по сравнению с SCons системой. На основании приведенных аргументов предлагается использовать CMake для улучшения системы сборки компонента инсталляции навигационных карт.

Для удобства создания новых модулей компонента инсталляции и поддержки уже разработанных был подготовлен набор вспомогательных методов, упрощающих описание сборки модулей. Отметим также, что для компонента инсталляции требуется выполнять постепенный переход между устаревшей и разработанной системами; для решения этой задачи были разработаны механизмы сборки, позволяющие выполнять построение части модулей компонента средствами SCons, а другой части – механизмами CMake .

В целях контроля качества компонента инсталляции следует осуществлять модульное тестирование, с представлением результатов в понятной человеку форме. В устаревшей системе сборки и тестирования компонента инсталляции управление модульными тестами осуществлялась с помощью разработанных вспомогательных утилит, так как средствами SCons это выполнить затруднительно. Так как CMake позволяет управлять процессом тестирования, то в новой разработанной системе предлагается построить процесс тестирования на его основе, и практически полностью заменить устаревшие утилиты организации тестирования.

Для удобства разработки проектов важно, чтобы в разрабатываемой системе сборки присутствовали возможности представления исходных файлов проекта в популярных IDE. Устаревшая система сборки позволяла выполнять генерацию проекта только для среды Microsoft Visual Studio. Так как требуется поддержка кроссплатформенной сборки, то необходимо расширить поддержку IDE для остальных платформ. Поэтому разрабатываемая система осуществляет генерацию проектов для таких IDE, как Visual Studio, Xcode, CodeBlocks и др., таким образом предоставляя инструменты удобной разработки компонента для всех необходимых платформ. В силу того, что CMake активно развивается, существуют как устаревшие, так и современные подходы по организации stake-файлов. Поэтому на основании анализа литературы, был подготовлен набор рекомендаций по написанию stake-файлов с применением современного подхода «Modern CMake».

Реализован набор методик организации проекта с большим количеством компонентов и модулей. На основе разработанных методик осуществляется сборка компонента инсталляции навигационных карт и его тестирование. На основании анализа литературы были разработаны вспомогательные функции для написания CMake-файлов, подобные функции позволили в значительной степени упростить написание CMakeLists.txt файлов для модулей системы. Разработанная структура директорий проекта позволит успешно расширять в дальнейшем количество модулей в проекте. В ходе работы была осуществлена поддержка тестирования. Для этого был выбран инструмент CTest. Реализован подход по представлению результатов тестов в сервисе непрерывной интеграции Jenkins во время удаленной сборки проекта.

Реализация разработанной системы сборки будет внедрена в процессы разработки программного обеспечения картографической библиотеки компании Wärsilä Digital Technologies.