

**Проектирование системы кроссплатформенной сборки
компонентов картографической библиотеки**

Аржевитин Б.К. (федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО», Санкт-Петербург),

Научный руководитель – к.т.н., Вайгандт Н. Ю. (федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО», Санкт-Петербург)

Введение. В настоящее время увеличивается количество разрабатываемых крупных программ (программное обеспечение – ПО) со сложной архитектурой компонентов, различными внешними зависимостями (библиотеками) и большой кодовой базой. Из-за увеличения сложности ПО, вместе с задачами их непосредственной разработки, становятся актуальными вопросы проектирования более сложных систем сборки подобного ПО. Практически любая программа будь то калькулятор или картографическая ГИС система не обходится без системы сборки, однако, например, в случае ГИС системы механизмы, задействованные для сборки подобного ПО, будет существенно сложнее. Подобные крупные программы включают в себя большое число различных компонентов, модулей, зависимостей. 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. В крупных командах разработки обычно не все сотрудники досконально знают как устроены процессы сборки продукта, поэтому подобная система должна быть спроектирована 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. Немаловажным фактором является время сборки ПО, на которое влияет сборочная система. Удобные способы и простые методы взаимодействия разработчиков с системой сборки позволяет им быстрее вносить в ПО новые изменения, что в свою очередь позволяет пользователям подобного ПО, например, той же ГИС системы, быстрее получать новую функциональность.

Основная часть. В работе проводится анализ существующей системы сборки крупной библиотеки с большим количеством модулей и зависимостей, поиск слабых частей существующей системы, анализ альтернативных решений для проектирования сборочных систем.

Проведенный анализ используемой в настоящее время системы сборки показал, что для функционирования этой системе необходима поддержка кросскомпиляции под мобильные платформы, организация управления модульными тестами, кроссплатформенная сборка модулей. Кроме того, в процессе сборки 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 в работе предлагается разработать механизмы сборки с применением CMake. Причинами выбора CMake является простота синтаксиса описания stake-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 проекта и поддержки уже разработанных был разработан ряд методик и рекомендаций, позволивших упростить описание сборки модулей. В целях контроля разработанной сборочной системы следует осуществлять интеграцию модульного тестирования с представлением результатов в понятной человеку форме. В устаревшей системе сборки и тестирования управление модульными тестами осуществлялось 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».

Выводы. Подготовлен набор методик организации проекта с большим количеством компонентов и модулей. На основе разработанных методик реализована сборка крупной картографической библиотеки и ее тестирование. На основании анализа литературы были разработаны вспомогательные функции для CMake-файлов, подобные функции позволили в значительной степени упростить составление подобных файлов для описания сборки модулей системы. Разработанная структура директорий проекта позволит успешно расширять в дальнейшем количество модулей в проекте. В ходе работы была осуществлена поддержка тестирования системы. Для этого был выбран инструмент CTest.

Апробация результатов. Реализация разработанной системы сборки будет внедрена в процессы разработки программного обеспечения картографической библиотеки компании АО «Вяртсиля Цифровые Технологии»