Совместное использование нескольких автоматизированных систем сборки в проекте

Д. А. Никитин
Белорусский государственный университет информатики и радиоэлектроники, Минск, Республика Беларусь

Научный руководитель: Парафиянович Т.А. – канд.пед.наук, доцент, доцент кафедры ИРТ

В тезисах представлена информация об использовании нескольких систем сборки программных средств, на примере программного средства itVPN ООО «ИТТАС». Рассмотрены недостатки формирования общего алгоритма выполнения сборки через скрипты командной строки, а именно привязка к используемым программам. Определены способы решения выявленных проблем — взаимодействие с командами, а не конкретными программами. Спроектирована и разработана программа на языке программирования Python, решающая выявленные проблемы. Проведен анализ разработанной программы и определены пути ее улучшения.

Ключевые слова: автоматизированные системы сборки программ, конфигурирование систем сборки, построение алгоритма сборки, CMake, GNU Autotools.

При прохождении учебной производственной практики в компании ООО «ИТТАС» стояла задача сборки программного средства itVPN. Данное программное средство разработано на базе openVPN с модифицированной библиотекой openSSL и модулем, реализующем криптографические алгоритмы [1].

При сборке itVPN в качестве точки входа, которая определяет и регулирует последовательность выполнения сборки выступают скрипты командной строки. В качестве автоматизированных систем сборки выступают:

– GNU Autotools и Make: для itVPN и openSSL;

– CMake: для криптографического модуля.

При взаимодействии с процессом сборки itVPN выявлены следующие недостатки:

1) при рассмотрении физической структуры скриптов оказывается, что если необходимо разрабатывать более сложные скрипты, то их необходимо больше разделять на файлы. Таким образом, у сложного продукта может быть огромный набор из скриптов сборки;

2) при сборке itVPN для тестирования есть необходимость выполнять сборку его конкретных модулей, а не всей системы в целом, и чтобы решить эту задачу с использованием скриптов необходимо комментировать строки кода, отвечающие за компиляцию конкретного модуля или модулей, что не является удобным для текущей реализации механизма сборки;

3) одним из более важных недостатков выступает чрезмерный поток информации, отображающийся в окне терминала — речь идет о логах выполнения утилиты Make и CMake.

В связи с этим принято решение изменить скрипты командной строки на программное обеспечение с учетом рассмотренных недостатков.

Основной проблемой больших проектов является использование множества утилит, в том числе систем сборки. Так, например, для изменения состояния логов или добавления других опций необходимо взаимодействовать с конкретными утилитами, в рассматриваемой ситуации — это системы сборки. При повышении уровня абстракции происходит взаимодействие с терминальными командами. Такой подход стандартизирует конфигурирование процесса сборки, однако не является достаточно мобильным, т.к. необходимо изменять исходный код скриптов.

Для решения проблемы определено, что необходимо вынести необходимые опции в конфигурационный файл, который регулирует процесс выполнения сборки. Чтобы

поддерживалась кроссплатформенность необходимая для itVPN принято решение использовать язык программирования Python.

При проектировании программного модуля выделены следующие сущности:

1) этапы сборки — модуль отвечает за включение конкретных шагов сборки;

2) конфигурация сборки — отвечает за взаимодействие с параметрами сборки, такими как: тип компиляции, архитектура, система и др., включает в себя этапы сборки;

3) инструмент логирования — отвечает за вывод информации выполнения сборки;

4) сборщик — выполняет терминальные команды в соответствии с опциями, указанными в конфигурации;

5) прослойка основных терминальных команд, которые могут быть реализованы в языке программирования.

В качестве утилиты для формирования исполняемого файла из скриптов языка Python выбран модуль pyinstaller, он позволяет не только преобразовать исходные коды в исполняемый файл, но и сделать его независимым от системных библиотек.

В процессе разработки было выявлено, что программу необходимо разрабатывать на версии языка 3.4, в связи с необходимостью развертки программы на операционных системах Debian 8 и CentOS 7. Помимо этого необходимо проводить ручное тестирование в связи с тем, что pyinstaller не является компилятором и перед преобразованием исходного кода в исполняемый файл отсутствует проверка корректности написанного кода.

На момент завершения разработки программного средства были решены основные проблемы совместного использования систем сборки в проекте: вывод логов был перенесен в файл, который формируется в случае неудачного процесса сборки; добавлена возможность включения и отключения шагов сборки в более простом виде чем изменение исходного кода.

Однако не было предусмотрено абстрагирование от выполняемых команд, таким образом программа работает только для конкретной версии программного средства itVPN и при переходе с версии openSSL 1.1.1 на версию 3 возникает потребность изменения исходного кода программы. Исходя из этого необходимо также вынести выполняемые команды в конфигурационный файл и продумать его структуру.

Список источников

[1] Reference manual for OpenVPN 2.0 / https://openvpn.net/community-resources/reference-manual-for-openvpn-2-0/

Sharing multiple automated и build systems in a project

D. A. Nikitin

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Scientific leader: T. A. Parafiyanovich – Ph.D. of Pedagogy, Professor of the Department of IRT

Annotation

The article provides information on the using of several software build systems, based on example of the itVPN software tool of ITTAS LLC. The disadvantages of forming a general algorithm for executing a build through command-line scripts, namely binding to the programs used, are considered. The ways to solve the identified problems have been identified — interaction with comand line instructions, not specific programs. A program in the Python programming language has been designed and developed to solve the identified problems. The analysis of the developed program has been carried out and ways to improve it have been identified.

Keywords: software build systems, configuration of build systems, design building algorithm, CMake, GNU Autotools.