

УДК 519.685

РАЗРАБОТКА МУЛЬТИПЛАТФОРМЕННОЙ УТИЛИТЫ АВТОМАТИЗАЦИИ СБОРКИ ПРОГРАММ ДЛЯ ООО «ИТТАС»

Никитин Д.А., Парафиянович Т.А.

*Белорусский государственный университет информатики и радиоэлектроники, г.Минск, Беларусь,
den221001@gmail.com*

Аннотация. Рассмотрены способы сборки программного средства itVPN, методы автоматизации сборки, определены недостатки разработки программ с высокой детализацией выполняемых команд, предложен подход повышения абстракции для возможности конфигурирования всех инструментов на одном уровне.

Ключевые слова. Автоматизация, сборка программных средств, кроссплатформенность, конфигурирование инструментов.

Идея разработки программы по автоматизации сборки других программ появилась при выполнении сборки программного средства itVPN в рамках прохождения учебной производственной практики в компании ООО «ИТТАС».

Программный продукт itVPN состоит из следующих компонентов:

Программное средство openVPN [1];

Библиотека openssl;

Движок для расширения функционала openssl (движок является сложным программным средством, которое включает более 5 модулей) [2].

Под сборкой программного средства подразумевается полная или частичная компиляция его модулей. Сборка программы необходима для проведения различного рода тестирования и сертификации с целью дальнейшей реализации продукта. При сборке программного средства itVPN для сертификации необходимо выполнять сборку под операционные системы на базе Windows и Linux, а также для нескольких архитектур (x86, ARM и MIPS).

Процесс сборки программы itVPN для сертификации состоит из следующих этапов:

1. Компиляция openssl;

2. Компиляция движка openssl;

3. Компиляция openVPN;

4. Формирование дополнительных файлов;

5. Установка иерархии каталогов и файлов по определенным правилам.

Сборка itVPN осуществляется для 32 и 64 разрядных операционных систем Windows, Debian, CentOS и архитектур ARM и MIPS, исходя из вариаций систем и архитектур сборка программного средства itVPN повторяется шесть раз. Важность автоматизации сборки позволит повысить эффективность и в дальнейшем она будет увеличиваться относительно добавлению новых систем для сборки.

Перед разработкой утилиты для автоматизации сборки необходимо определить и изучить текущие инструменты и методы сборки программы itVPN. Текущим инструментом сборки выступают скрипты командной строки, недостатком которых является необходимость их разделения для UNIX подобных операционных систем и систем семейства Windows. При рассмотрении физической структуры скриптов оказывается, что если необходимо разрабатывать более сложные скрипты, то их необходимо больше разде-

лять на файлы. Таким образом, у сложного продукта может быть огромный набор из скриптов сборки под Windows и UNIX системы.

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.

Одним из более важных недостатков выступает чрезмерный поток информации, отображающийся в окне терминала – речь идет о логах выполнения утилиты Make и CMake. Решением задачи может выступить перенаправление потока вывода в null, однако, при необходимости организации смешанного вывода от определенных команд или некоторые из них выполнять без вывода на экран, то придется вернуться и править исходные коды скриптов.

На основании рассмотренных проблем сборки с использованием скриптов принято решение переписать скрипты на язык Python, с дальнейшей его компиляцией в исполняемый файл для уменьшения количества используемых файлов.

После переработки механизма сборки выявлена возможность одновременной сборки различных версий itVPN за счет запуска нескольких экземпляров программ, такого нельзя было достичь с использованием скриптов командной строки. Для конфигурирования собираемых модулей выбран способ вынесения опций в конфигурационный файл формата JSON. Более гибкий подход и поддержка объектно-ориентированной парадигмы программирования, позволили разработать гибкую архитектуру с взаимозаменяемыми сущностями: набор команд; логгер; конфигуратор и опции сборки; сборщик.

Переработка метода сборки помогла избавиться от чрезмерного количества логов и множества файлов, однако при разработке данной утилиты не была предусмотрена гибкость сборщика, т.е. осталась привязка к выполняемым командам и при желании их изменить необходимо было обращаться к исходным кодам. Проблема была выявлена при переходе на новую версию openssl, что потребовало изменить имена файлов библиотеки.

Текущая утилита сборки привязана к конкретным командам, что являлось ошибочным решени-



ем – такой подход работает только в случае, если дальнейшие изменения или расширения продукта, не подразумеваются.

С учетом выявленных ошибок и недостатков определено, что новая утилита должна быть более абстрактной и все преимущества в виде конфигурационных файлов и разделения на независимые модули должны быть перенесены в нее.

Повышение абстракции заключается в необходимости утилиты выполнять терминальные команды, которые определены в конфигурационном файле. Перенос выполняемых команд в файл позволит избавиться от перекомпиляции утилиты. При выполнении команд программа должна руководствоваться опциями, конфигурационного файла. Опции должны быть привязаны к каждой команде, также должна быть реализована возможность группировки команд, это позволит реализовать возможность одновременного выполнения команд. Таким образом, после сборки openSSL можно одновременно запустить сборку openVPN и движка openSSL, т. к. они напрямую не связаны друг с другом и такое решение значительно экономит временные ресурсы.

Принято решение разделить утилиту на три модуля:
Библиотека с основным функционалом;

Консольная утилита для использования функционала библиотеки;

Пользовательский интерфейс для удобного редактирования конфигурационного файла.

В качестве языка программирования библиотеки и консольной утилиты выбран C++ – он является кроссплатформенным, что позволит запустить утилиту на необходимых системах.

Необходимость разработки консольной утилиты и пользовательского интерфейса как надстройки над библиотекой с основным функционалом обусловлена различным способом и местом применения программ. В задачи консольной утилиты входит только выполнение команд из конфигурационного файла. Пользовательский интерфейс же позволяет не только выполнять команды, но и в более удобном формате составлять и редактировать конфигурационный файл. Не все операционные системы, на которых происходит сборка itVPN, имеют графический интерфейс, поэтому запуск программы с пользовательским интерфейсом невозможен.

Пользовательский интерфейс будет разработан на языке Python. Совместимость библиотеки, реализованной на C++ и пользовательского интерфейса

Python будет осуществляться с использованием библиотеки sturpes.

Разрабатываемое программное средство будет базироваться на конфигурационном файле формата JSON. Таким образом, происходит скрещивание первых двух подходов. Для каждой системы если она имеет особенности будет необходимо добавить команды в конфигурационный файл, за счет чего они будут разными. Однако, теперь все команды можно будет регулировать с помощью пользовательского интерфейса.

Каждая команда рассматривается как объект, для которого можно устанавливать различные опции. Команды можно объединять в группы и настраивать параметры для целых групп.

Помимо этого, необходимо разработать секцию, которая будет отвечать за настройку среды операционной системы, перед выполнением основных команд, этим действием может быть установка путей к компиляторам в переменную окружения.

В ходе исследования был рассмотрен процесс сборки программного обеспечения itVPN компании ООО «ИТТАС». Определены недостатки существующего подхода в виде дублирования файлов и необходимости изменения исходных кодов скриптов сборки. На основе этих недостатков спроектировано и разработано программное средство на языке Python, которое устраняет вышеописанные недостатки, но еще не является универсальным решением, в связи с установкой правил выполнения внутри исходных кодов программы. С учетом нюансов обоих вариантов сборки спроектировано программное средство, позволяющее вынести правила сборки в отдельный конфигурационный файл, сделать программу устойчивой к изменениям процесса сборки, выполнять сборку быстрее и эффективнее, абстрагироваться от собираемого программного средства, сделать разрабатываемую программу универсальным инструментом.

Литература

1. Reference manual for OpenVPN 2.0 [Электронный ресурс]. – Режим доступа: <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-0>. – Дата доступа: 27.02.2024.
2. Dynamically loading the engine(s) [Электронный ресурс]. – Режим доступа: https://openssl-cng-engine.readthedocs.io/en/latest/using/dynamic_loading.html. – Дата доступа: 27.02.2024.

DEVELOPMENT OF CROSS-PLATFORM AUTOMATIZATION SOFTWARE BUILDING TOOL FOR LLC «ITTAS»

D.A. Nikitin, T.A. Parafiyonovich

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus, den221001@gmail.com

Abstract. This topic speaks about build ways for software itVPN, automatization building methods, detect disadvantages of automatization utils with high detalizing based on used commands, propose approach of abstract scaling for setup toolset at the same layer.

Keywords. Automatization, software building, cross-platform, toolset configuring.