

# LSB SDK - инструментарий разработки переносимых Linux приложений

Денис Силаков, Владимир Рубанов

Москва

ИСП РАН

LSB SDK

<http://ispras.linuxfoundation.org>

## Аннотация

В докладе представлен один из элементов инфраструктуры, разрабатываемой в ИСП РАН совместно с Linux Foundation для независимых разработчиков Linux приложений — LSB Software Development Kit (SDK) — инструментарий, основанный на использовании для компиляции и компоновки приложений специальных заголовочных файлов и библиотек, гарантирующих отсутствие среди зависимостей получаемой программы непереносимых интерфейсов. Также SDK позволяет создавать приложения для «legacy» дистрибутивов, имеющих принципиальные отличия от системы, в которой производится сборка.

## **Контроль внешних зависимостей приложения**

Внешние зависимости программ, собираемых из исходного кода, не всегда полностью контролируются разработчиками — ряд зависимостей могут быть обусловлены переменными окружения, версиями компилятора и библиотек и другими внешними факторами. При этом такие зависимости могут ограничивать переносимость приложения, снижая круг систем, где оно способно функционировать.

Некоторые характеристики среды сборки изменить достаточно сложно — например, при динамической компоновке с библиотекой, использующей версионирование символов, для результирующего файла в качестве зависимостей проставляются символы с версиями «по умолчанию». Однако эти зависимости будут препятствовать запуску приложения на более старых системах, в которых таких версий еще не было. Обойти эту проблему можно, лишь используя более старую библиотеку при компоновке; но в большинстве в системе присутствует только одна версия библиотеки, а ручное добавление более старой версии может быть нетривиально ввиду необходимости удовлетворить зависимости старой библиотеки.

Поэтому даже если разработчики осведомлены о наличии внешних факторов, привносящих нежелательные зависимости в их приложения, избавление от таких факторов и зависимостей может потребовать значительных усилий.

Одним из возможных путей решения проблемы контроля зависимостей и повышения переносимости приложений является использование инструментария LSB SDK, позволяющего с использованием одной и той же среды сборки получать программы с различными характеристиками, предназначенные для запуска в разных поколениях дистрибутивов Linux.

## **Состав LSB SDK**

LSB SDK включает три основных компонента:

- заголовочные файлы;
- библиотеки-заглушки;
- обертки для компиляторов C и C++ (lsbcc и lsbc++ соответственно).

SDK позволяет собирать приложения, отвечающие требованиям любой версии стандарта Linux Standard Base (см. <http://ldn.linuxfoundation.org/lbsb>), начиная с 3.0. Заголовочные файлы LSB SDK содержат декларации только тех функций, которые включены в одну из версий стандарта LSB, а значит только таких, что их использование является переносимым на большинстве популярных дистрибутивов. В файлах определяются все необходимые для использования функций типы данных и константы, а также макросы, которые не приводят к появлению зависимостей от нестандартизованных функций.

Библиотеки-заглушки экспортируют только бинарные символы, включенные в стандарт. Для каждой версии стандарта SDK содержит отдельный набор библиотек-заглушек.

В процессе своей работы `lsbcc` и `lsbcc++` вызывают системный компилятор, но при компиляции и компоновке используются библиотеки и заголовочные файлы из SDK.

На основе опции «`--lsb-target-version`» (задающей целевую версию LSB) выставляется значение переменной среды, влияющей на доступность деклараций внутри заголовочных файлов, а также выбирается набор библиотек-заглушек, с которыми необходимо производить компоновку.

В зависимости от целевой версии LSB, выставляются различные опции компилятора и компоновщика — запрет или разрешение использования «stack protection» в функциях `glibc`, выставление (при необходимости) «`--hash-style`» в «`sysv`» и т.п.

Каждая версия LSB соответствует определенному поколению основных дистрибутивов Linux (LSB 3.0 — RHEL 4.2, SLES 10.0, Mandriva 2006, LSB 3.1 — RHEL 5, Mandriva 2007.0, SLES 10.1, и т.п.). Поэтому разработчики могут выбирать версию LSB, ориентируясь на целевые дистрибутивы, в которых должно функционировать их приложение.

## **Использование LSB SDK**

Использование LSB SDK прозрачно для разработчиков — в большинстве случаев достаточно вместо системного компилятора просто использовать `lsbcc` (`lsbcc++` для программ на C++) — например, посредством присвоения переменной `CC` значения «`lsbcc`» (а переменной `CXX` — значения «`lsbcc++`»). Для проектов, использующих `pkg-config`, предоставляются «`.pc`» файлы, при использовании которых `pkg-config` возвращает информацию, соответствующую LSB SDK, а не системным компонентам. Также поддерживается совместная работа с `libtool`.

## **Использование элементов, не входящих в LSB**

LSB SDK позволяет осуществлять компоновку с библиотеками, не входящими в LSB, с помощью опции «`--lsb-shared-libs`». Опции «`--lsb-libpath`» и «`--lsb-includedir`» позволяют указать пути к библиотекам и заголовочным файлам, которые следует использовать при компиляции и компоновке вместо тех, что входят в состав LSB SDK. В будущем планируется реализация режима «relaxed mode», позволяющего использовать не-LSB символы из библиотек, включенных в стандарт.

По умолчанию программы, собранные с помощью LSB SDK, используют в качестве загрузчика `ld-lsb`. Это поведение может быть изменено с помощью опций «`--lsb-use-default-linker`» и «`--lsb-besteffort`», при использовании которых в качестве требуемого загрузчика указывается `ld-linux`. Для приложений, собранных с опцией «`--lsb-besteffort`», реально используемый загрузчик выбирается при старте приложения —

если возможно, то используется `ld-lsb`, в противном случае — `ld-linux`.

Вместо вышеперечисленных опций возможно использование соответствующих переменных среды, информацию о которых можно найти в справке `lsbcc`.

## **LSB Eclipse Plugin**

Инструментарий LSB SDK может быть интегрирован в среду разработки Eclipse с помощью соответствующего плагина, предоставляемого Linux Foundation ([http://ispras.linuxfoundation.org/index.php/About\\_LSB\\_Eclipse\\_Plugin](http://ispras.linuxfoundation.org/index.php/About_LSB_Eclipse_Plugin)). Плагин добавляет в Eclipse следующие виды проектов:

- LSB Executable program (C/C++);
- LSB Shared Library (C/C++);
- LSB Static Library (C/C++).

Проекты являются аналогами соответствующих стандартных проектов Eclipse, однако сборка осуществляется с помощью `lsbcc` и `lsbcc++`. Поддерживается преобразование проектов из «обычных» в LSB и обратно.

Плагин позволяет осуществлять настройку всех опций `lsbcc` и `lsbcc++` непосредственно из Eclipse. В частности, могут быть заданы версия LSB, указаны не входящие в LSB библиотеки, с которыми следует производить компоновку, заданы дополнительные опции для системного компилятора.

Помимо интеграции инструментария LSB SDK, плагин позволяет с помощью Linux Application Checker анализировать совместимость собранного приложения с различными дистрибутивами Linux непосредственно во встроенном веб-браузере Eclipse.

Для функций, используемых в приложении, плагин позволяет отображать в браузере Eclipse информацию, получаемую от LSB Navigator (<http://linuxfoundation.org/navigator>) — сигнатуры, ссылки на документацию, данные о присутствии в дистрибутивах и использовании в приложениях и другие сведения.

## **Заключение**

Изначально основной целью LSB SDK была помощь разработчикам в создании приложений, совместимых со стандартом LSB. Однако, в настоящее время функциональность, предоставляемая инструментарием, существенно шире. Гибкость и разнообразие режимов работы SDK позволяют разработчикам использовать его для повышения переносимости своих приложений, не ограничиваясь при этом рамками LSB.