# Проблемы мобильности программного обеспечения

Выполнил Ядрихинский Николай Васильевич
Группа 16212

# Мобильность программного обеспечения

Наиболее распространенными контекстами рассмотрения мобильности ПО являются следующие:

– переносимость ПО между различными аппаратными платформами;

– переносимость ПО между различными операционными системами (ОС);

– комбинация вышеперечисленных вариантов.

# Переносимость между различными аппаратными платформами

# Переносимость между различными аппаратными платформами

Наиболее часто эта проблема решается на уровне исходных кодов приложения. Если приложение написано на интерпретируемых языках или языках программирования высокого уровня (Java, C#) приложение практически автоматически переносится на любую систему, где реализована соответствующая среда исполнения. При использовании промежуточного представления, такого как, Java байт-код, это можно делать даже без доступа к исходному коду.

# Переносимость между различными аппаратными платформами

Если приложение написано на языке программирования более низкого уровня (Си, C++), то для обеспечения переносимости простой перекомпиляцией разработчик должен аккуратно учитывать возможные отличия между платформами, такие как, размеры базовых типов данных, их представление, неявные преобразования между типами и так далее. Это не является чрезмерно сложной задачей при достаточной культуре программирования, хотя и требует дополнительных усилий на проведение тестирования на целевой платформе.

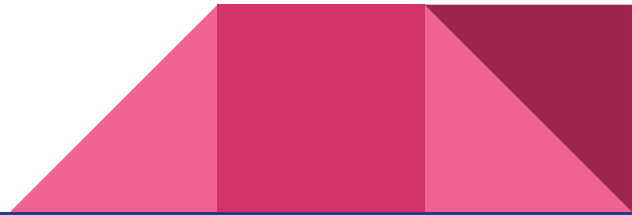
# Переносимость между различными аппаратными платформами

В случае когда переносимость на уровне исходных кодов недоступна возможно использовать переносимость на бинарном уровне, которая достигается при помощи эмуляции целевой аппаратной платформы. Эта эмуляция может быть реализована в операционной системе (например, динамический транслятор Rosetta, используемый в Mac OS X, работающих на компьютерах Apple с процессорами Intel, для выполнения программ, предназначенных для машин с процессорами PowerPC ) или в виде независимого приложения (например, эмулятор Hercules для эмуляции мейнфреймов IBM S370, S390).

# Переносимость между различными операционными системами (ОС)

# Переносимость между различными операционными системами (ОС)

Интерпретируемые языки и языки высокого уровня сами по себе уже не могут полностью решить эту проблему, так как большинству приложений не хватает базовых библиотек времени исполнения своего языка, а также им приходится взаимодействовать с множеством объектов, специфичных для различных ОС. Для языков низкого уровня ситуация еще более сложная ввиду ограниченности их библиотек времени исполнения.

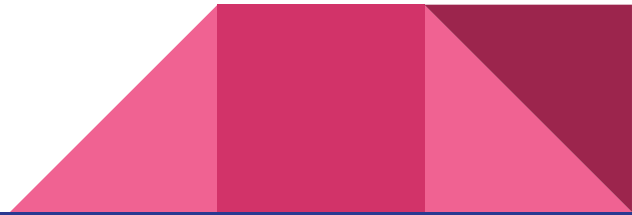


# Переносимость между различными операционными системами (ОС)

На помощь здесь приходит стандартизация интерфейсов взаимодействия между приложениями и ОС (например, стандарт POSIX ), которая определяет контракт между приложениями и совместимыми ОС. К сожалению, в настоящее время стандарты такого рода поддерживаются далеко не всеми ОС, да и охватывают они лишь небольшую часть необходимой функциональности.

# Переносимость между различными операционными системами (ОС)

Для разрешения проблемы возможно использовать дополнительные кроссплатформенные библиотеки-медиаторы, которые реализуют требуемую функциональность на различных ОС и предоставляют приложению общий для всех ОС программный интерфейс. Конечно, библиотека должна удовлетворять требованиям приложения не только по предоставляемой функциональности, но и по многим других характеристикам.



# Переносимость между различными операционными системами (ОС)

Виртуализация позволяет запустить копию одной ОС внутри другой ОС, и, таким образом, любое приложение, предназначенное для первой ОС, может работать внутри второй ОС. Данный способ достаточно неудобен, как в смысле производительности, так и ввиду необходимости иметь лицензии на обе ОС. Еще один подход к обеспечению мобильности ПО — это разработка Веб-приложений, в которых клиентская часть ПО взаимодействует с серверной посредством стандартов HTTP и HTML, что делает возможным работу с ПО на любой ОС, предоставляющей Веб-браузер.

# Переносимость между различными операционными системами (ОС)

Альтернативным вариантом решения проблемы переносимости является использование эмуляторов и слоев совместимости исходной ОС на целевой ОС. В качестве примеров такого инструментария можно привести:

– wine — слой совместимости с WinAPI для Linux, MacOS, Solaris, FreeBSD и др.;

– cygwin — слой совместимости с Linux для семейства ОС Microsoft Windows.

# Недостатки обеспечения мобильности ПО

# Недостатки обеспечения мобильности ПО

– снижение производительности ПО;

– удорожание процесса разработки ПО;

– усложнение архитектуры ПО;

– дополнительные требования к квалификации разработчиков;

– осложнение использования возможностей, доступных только на некоторых платформах.

# Недостатки обеспечения мобильности ПО

Это означает, что разработчикам и заказчикам ПО следует принимать решение о необходимости обеспечения переносимости в зависимости от условий конкретного проекта. Для краткосрочных проектов потребности в переносимости ПО, как правило, лежат на поверхности. Если же результаты проектов предполагается использовать в длительной перспективе, то анализ рисков должен быть более тщательным.

# Недостатки обеспечения мобильности ПО

В частности, нельзя оставить вне рассмотрения следующие риски:

– устаревание и выход из строя аппаратных и программных платформ, на которых базируется целевое ПО;

– зависимость от поставщиков базового и промежуточного ПО, требуемого как для работы целевого ПО, так и для его разработки;

– недоступность целевого ПО для части потенциальных потребителей.

# Недостатки обеспечения мобильности ПО

Заказчики ПО при принятии решения о необходимости обеспечения переносимости ПО в конкретном проекте должны учитывать следующие факторы:

– собственный парк аппаратного и программного обеспечения, а также перспективы его развития;

– риски попадания в зависимость от поставщиков аппаратного и программного обеспечения;

– дополнительные затраты на обеспечение переносимости.