



— Портирование приложений —

МФТИ
Осень 2024

Терминология

- **Нативная (*родная*) платформа** – платформа, под которую изначально проектировалось и разрабатывалось приложение
- **Портирование** – процесс адаптации приложения под платформу, отличную от нативной с максимальным сохранением всех его пользовательских свойств
- **Порт** – результат портирования
- **Портируемость** – свойство ПО, описывающее легкость его портирования
- ПО называют **портабельным (*portable*)**, если затраты на его портирование ниже, чем разработка с нуля для заданной платформы

Задание: портируйте DOOM

- Запустите DOOM на своем симуляторе

Что является частью платформы?

- ISA?
- OS?
- Периферия?
- ABI?

Что является частью платформы?

- ISA? – да
- OS?
- Периферия?
- ABI?

Что является частью платформы?

- ISA? – да
- OS? – да
- Периферия?
- ABI?

Что является частью платформы?

- ISA? – да
- OS? – да
- Периферия? – да
- ABI?

Что является частью платформы?

- ISA? – да
- OS? – да
- Периферия? – да
- ABI? – да

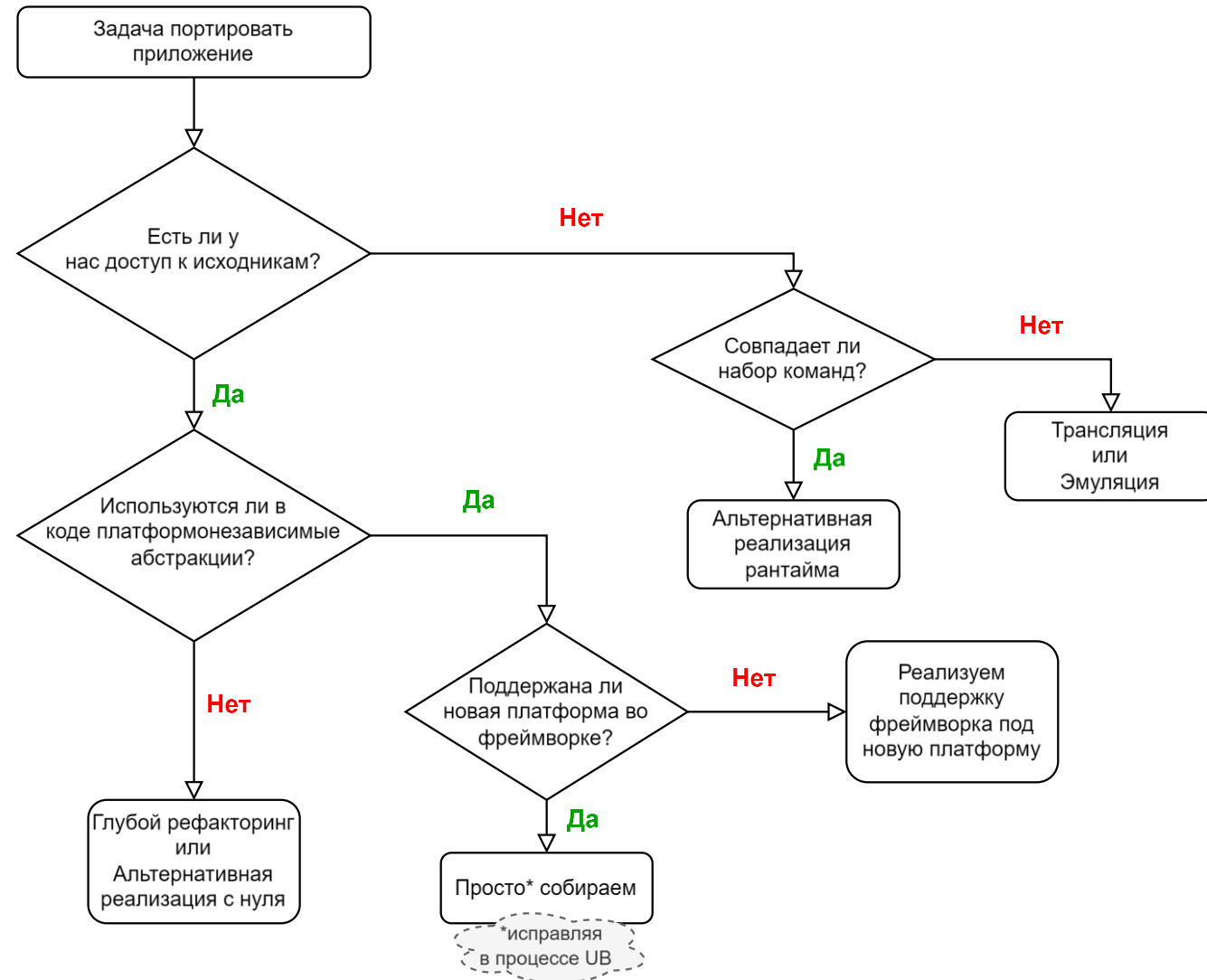
Обсуждение: портирование

Вам поставили задачу портировать приложение

- Ваши действия?

Варианты принятия решений при портировании

- Эмуляция
- Трансляция
- Альтернативная реализация рантайма
- Пересборка
- Поддержка платформы в абстрагирующем фреймворке
- Глубокий рефакторинг с внедрением платформонезависимых абстракций
- Переписывание с нуля



Портирование: Эмуляция

- Симуляция – повторение внешнего поведения
- Эмуляция – повторение внутреннего поведения
- Эмуляция платформы достигается за счет:
 - Симулирования архитектуры
 - Эмулирования периферии
 - Иногда эмулирования ОС
- Примеры:
 - QEMU
 - Различные эмуляторы консолей
- Преимущества:
 - Простое и быстрое решение для простых и сильно отличающихся платформ
 - Не требует доступа к исходному коду
- Недостатки:
 - Самое неэффективное решение по производительности

Портирование: Трансляция

- Если среда исполнения и набор команд примерно схожи, то можно транслировать код одной платформы в код другой платформы на лету (online) или статически (offline)
- Транслятор может быть реализован как программно, так и аппаратно
- Примеры:
 - Apple Rosetta
 - Микропроцессор Эльбрус
- Преимущества:
 - Выше производительность по сравнению с эмуляцией
 - Не требуется доступ к исходному коду приложения
- Недостатки:
 - Требуется больших усилий для реализации
 - Требуется еще больших усилий для получения удовлетворительной производительности

Портирование: Альтернативный рантайм

- Допустим мы хотим запустить Windows приложение на Linux
- У нас есть сборка под тот же процессор
- Отличие только в ABI и OS API
- Можно сделать альтернативную реализацию OS API с необходимым ABI
- Примеры:
 - WINE (**W**ine **I**s **N**ot an **E**mulator)
 - WSL (**W**indows **S**ubsystem for **L**inux)
- Преимущества:
 - Примерно нативная производительность (системные вызовы и так медленные)
 - Не требуется доступ к исходному коду приложения
- Недостатки:
 - Сложно заставить работать такой подход сразу для нескольких приложений
 - Системные вызовы могут иметь незадокументированное поведение
 - Приложения могут закладываться на баги в некоторых системных вызовах

Портирование: Платформонезависимый фреймворк

- Зачем париться с эмуляторами, трансляторами, рантаймами, если можно сразу все хорошо спроектировать?
- Выделить интерфейс, который смогут реализовать все запланированные платформы, в специальный фреймворк
- Примеры:
 - Qt, SDL
 - Electron, Java
 - Linux, Windows
- Преимущества:
 - Почти бесплатное портирование на все поддерживаемые платформы
- Недостатки:
 - Необходимо подумать заранее
 - Во фреймворке могут быть ошибки
 - Фреймворк может быть не оптимально портирован
 - Для портирования требуется исходный код приложения
 - Фреймворк может быть портирован не на все необходимые платформы

Портирование: Реализация с нуля

- Просто пишем одно и то же приложение N раз для N платформ
- Преимущества:
 - Возможность дать самые строгие гарантии по производительности
 - Необязателен доступ к исходному коду
- Недостатки:
 - В процессе разработки может разъехаться функциональность приложения
 - Повышенное количество ресурсов на разработку и поддержку

Задание: запустите DOOM

- Запустите [порт DOOM'а для встраиваемых платформ](#) на вашем симуляторе
- Особенность порта – вся графика выводится в ASCII с цветами
- Перед запуском вам понадобится загрузить Linux =)



To be continued ...

На следующем занятии

- Познакомимся с концепцией векторизации вычислений
- Познакомимся с концепцией Compare-and-Swap циклов
- Узнаем какие особенные возможности для них предоставляет RISC-V