Яндекс

Состояние дел в Fuchsia

Гена Евстратов, энтузиаст

Состояние дел в Fuchsia

01

Введение

Почему про Fuchsia?

> **Почти всегда, когда речь заходит о Flutter так или иначе всплывает ОС Fuchsia, для которой он основной инструмент разработки UI**

При этом мало кто что-то знает про саму ОС, кроме того, что она существует, не говоря уже о том, чтобы попробовать что-то для неё собрать

Поэтому сегодня в развлекательном формате поговорим про то, как она устроена, что нужно, чтобы собрать первое флаттер приложение и что ещё есть вокруг этой ОС

Что такое Fuchsia?

> **Fuchsia это open-source операционная система общего назначения**

При её разработке руководствуются следующими принципами

- Простота
- Безопасность
- Простота обновлений
- Производительность

Об этих принципах и её устройстве мы поговорим чуть позже и чуть подробнее

Где она работает?

Сейчас Fuchsia поддерживает архитектуры arm64 и x86-64

> **Уже сейчас она работает в устройствах Nest (и UI там написан на Flutter)**

Для целей разработки и развлечения её можно запустить

- В эмуляторе
- На chromebook
- На intel NUC

Какие на неё планы?

> **Никто не знает, какие у Google планы на Fuchsia**

При этом

- Разработка Fuchsia идёт достаточно активно
- В ней участвуют не только инженеры гугла
- Ходят слухи, что будет сделан слой совместимости Fuchsia и Android
- И слухи, что гугл заменит Android на Fuchsia

Поживём увидим!

02

Принципы разработки Fuchsia

Принципы разработки Fuchsia

> **Разработчики Fuchsia следуют набору базовых принципов, когда создают эту ОС**

Вот эти принципы сверху вниз

- Простота
- Безопасность
- Обновляемость
- Производительность

Рассмотрим их подробнее

Простота

> **Fuchsia упрощает создание, поддержку и интеграцию программ и железа для большого количества Устройств**

- Простота и минимализм — приоритет при разработке архитектуры и дизайна системы
- Разработчики стараются делать максимально простую для понимания и поддержки систему
- Система надежно предоставляет базовые сервисы (доступ к железу и ресурсам, программные абстракции), становясь таким образом простой и стабильной платформой для разработки сервисов.

Говоря человеческим языком, они стараются применять принцип KISS, и у них это на самом деле получается

Безопасность

Ядро и внутреннее устройство Fuchsia созданы специально для современных задач

- Основанная на понятии «разрешение» система прав доступов по умолчанию полностью изолирует процессы, и предоставляет доступ только к явно запрошенным разрешениям и ресурсам
- Приложения распространяются в полностью изолированных пакетах, которые предоставляют гарантии безопасности и четкие границы софта
- Система сама заставляет использовать sandboxing, что сильно облегчает организацию безопасной среды

Обновляемость

> **Так как Fuchsia является модульной операционной системой, ядро, драйверы и другие компоненты могут быть обновлены независимо друг от друга**

- Стабильные ABI дают возможность ядру, драйверам и прикладным программам оставаться совместимыми в течении долгого времени
- Приложения распространяются в пакетах, которые могут обновляться независимо и даже по запросу, почти как веб-сайты
- Google обещает поддерживать и обновлять эту ОС в течении долго времени

Производительность

> **Дизайн Fuchsia разрабатывается с учетом реальных устройств и оптимизируется в сторону производительности**

- Микроядро эффективно управляет ресурсами, и код планировщиков оптимизирован в сторону производительности
- Система хорошо показывает себя в текущих применениях в продакшене

03

Базовое устройство Fuchsia

Как это устроено в Fuchsia

> **Fuchsia достаточно сильно отличается от более широко распространенных ОС. Перед тем, как что то писать для этой ОС стоит понять, как там всё устроено**

Основные отличия примерно такие

- Отсутствуют пользователи, права на выполнение и доступ раздаются за счёт явных запросов «разрешений»
- Почти всё в системе это «компонент», в том числе и выполняемый код
- Зависимости между компонентами разрешаются в рантайме, а не во время установки, что скорее похоже на веб, чем на другие ОС

Компоненты в Fuchsia

> **Почти всё в системе это «компонент», в том числе и выполняемый код. С ними работает специальная часть ОС – Component Framework**

- Component Framework это то, что управляет и выполняет все компоненты в системе
- Компоненты зависят друг от друга по API. Это абстрактные зависимости по интерфейсу, которые описываются через FIDL. Абстрактные в этом контексте означает, что реализацию зависимости осуществляет другой компонент, а какой именно определяет Component Framework
- FIDL это специальный язык, который описывает IPC компонентов
- Компоненты распространяются в пакетах. Компоненты могут использовать общие библиотеки только из своего собственного пакета. Это называется ABI зависимостью
- В системе НЕТ концепции зависимостей между пакетами

На чём пишут компоненты в Fuchsia

- Dart
- Rust
- C
- C++
- GO
- Python

При этом есть понятный путь добавить поддержку в систему любого языка

Что такое ядро ОС?

> **Ядро — центральная часть операционной системы, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, память, внешнее аппаратное обеспечение.**

Как правило это делается за счет так называемых системных вызовов (syscalls)

Ядра бывают разных типов, и от этого во многом зависит работа ОС

Монолитное ядро

> **Ядро, все компоненты которого являются частью одной программы, используют общие структуры данных и взаимодействуют друг с другом непосредственно вызовом процедур называется монолитным.**

- Монолитным ядром являются старые ядра UNIX и Linux
- Монолитные ядра проще разрабатывать, они работают быстрее
- Из-за того, что всё ядро выполняется в одном адресном пространстве сбой в части ядра может обрушить всю систему

Модульное ядро

> **Это современная и усовершенствованная концепция монолитного ядра**

По большому счету это монолитное ядро, которое научилось подгружать в себя определённые модули, например драйверы устройств. Теперь его не надо пересобирать при каждом обновлении железа, ура!

При этом они продолжают выполняться в одном адресном пространстве, и, если что-то идёт не так, то всё ядро может сломаться.

Микроядро

> **Микроядро представляет только базовые функции управления процессами и минимальный набор абстракций для работы с оборудованием**

Остальное осуществляется внешними программами, которые работают каждая в своём адресном пространстве

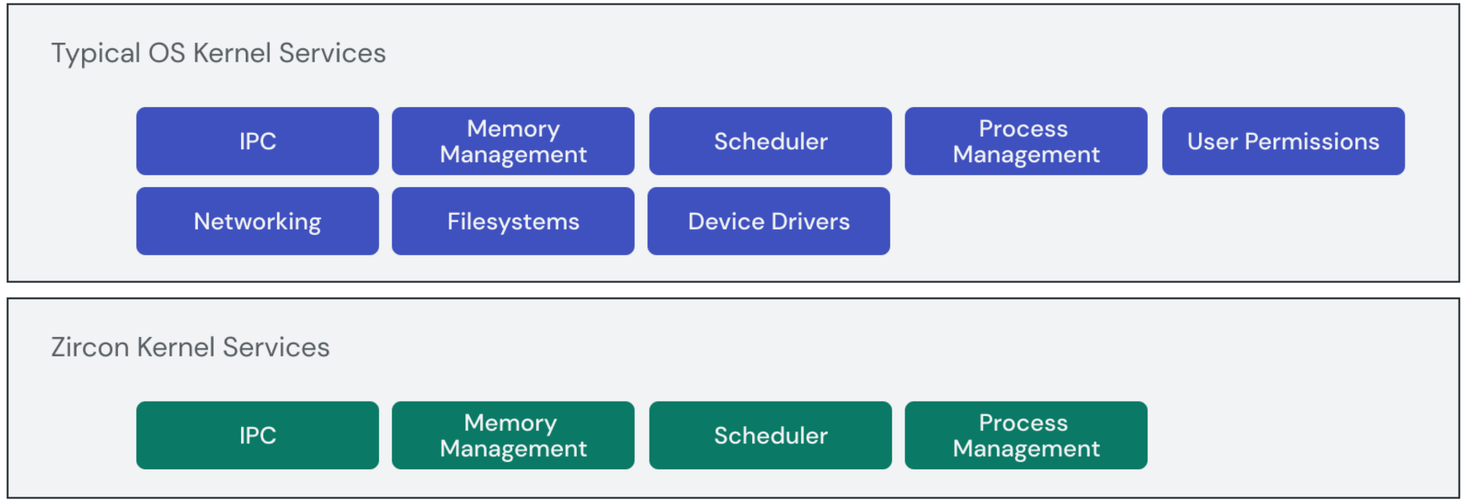
Соответственно, если что-то случается с одной из них, остальные остаются в безопасности

В Fuchsia используется именно такой подход

Микроядро в Fuchsia

Fuchsia использует микроядро, которое называется Zircon

- в нём примерно 170 системных вызовов
- больше, чем в типичном микроядре
- сильно меньше, чем в ядрах других систем



Что из этого следует?

> **Микроядерная архитектура Fuchsia позволяет вынести большинство кода в userspace, тем самым улучшив безопасность и упростив разработку**

Например, все драйверы, файловые системы по сути являются теми же компонентами с соответствующими FIDL описаниями

Это же даёт возможность на лету обновлять и загружать различные компоненты

04

Погружение в Fuchsia и Flutter

Что нужно чтобы начать этот путь?

> **Официальная документация говорит, что нужен x86_64 компьютер с Linux или MacOS**

В реальной жизни у меня не получилось собрать ничего под MacOS, поэтому я использовал Lunix лаптоп

Кроме этого нужен хороший интернет и некоторое количество терпения, потому что собирать ОС это не самое быстрое дело

Какие шаги нужно сделать?

> **Я не буду пересказывать туториал с официального сайта, а расскажу только о проблемах, с которыми столкнулся сам**

Итак, нам нужно научиться

- собирать ОС из исходников
- Запускать собранный образ в эмуляторе
- Загружать и запускать свои компоненты

Что у меня пошло не так?

Примерно всё

Если у вас что-то не заработает после прохождения официального туториала проверьте следующие моменты

- Вы указали все нужные компоненты при сборке
- После конфигурации вы их пересобрали
- У вас запущен сервер компонентов в соседнем терминале

Дополнительные проблемы, которые возникли у меня

- По умолчанию не стояли библиотеки vulkan на ubuntu
- Когда я их поставил, эмулятор с ними всё равно не заработал и пришлось использовать программную графику

Первый код на Flutter

> **Для экспериментов я выбрал уже существующий тестовый компонент на Flutter и начал его модифицировать**

Процесс не быстрый, потому что после каждого изменения нужно

- Пересобрать компонент
- Обновить его в эмуляторе

После hot reload на остальных платформах это достаточно долго и непривычно

Зависимости из pub

> **Fuchsia достаточно неудобно поддерживает зависимости из pub.dev**

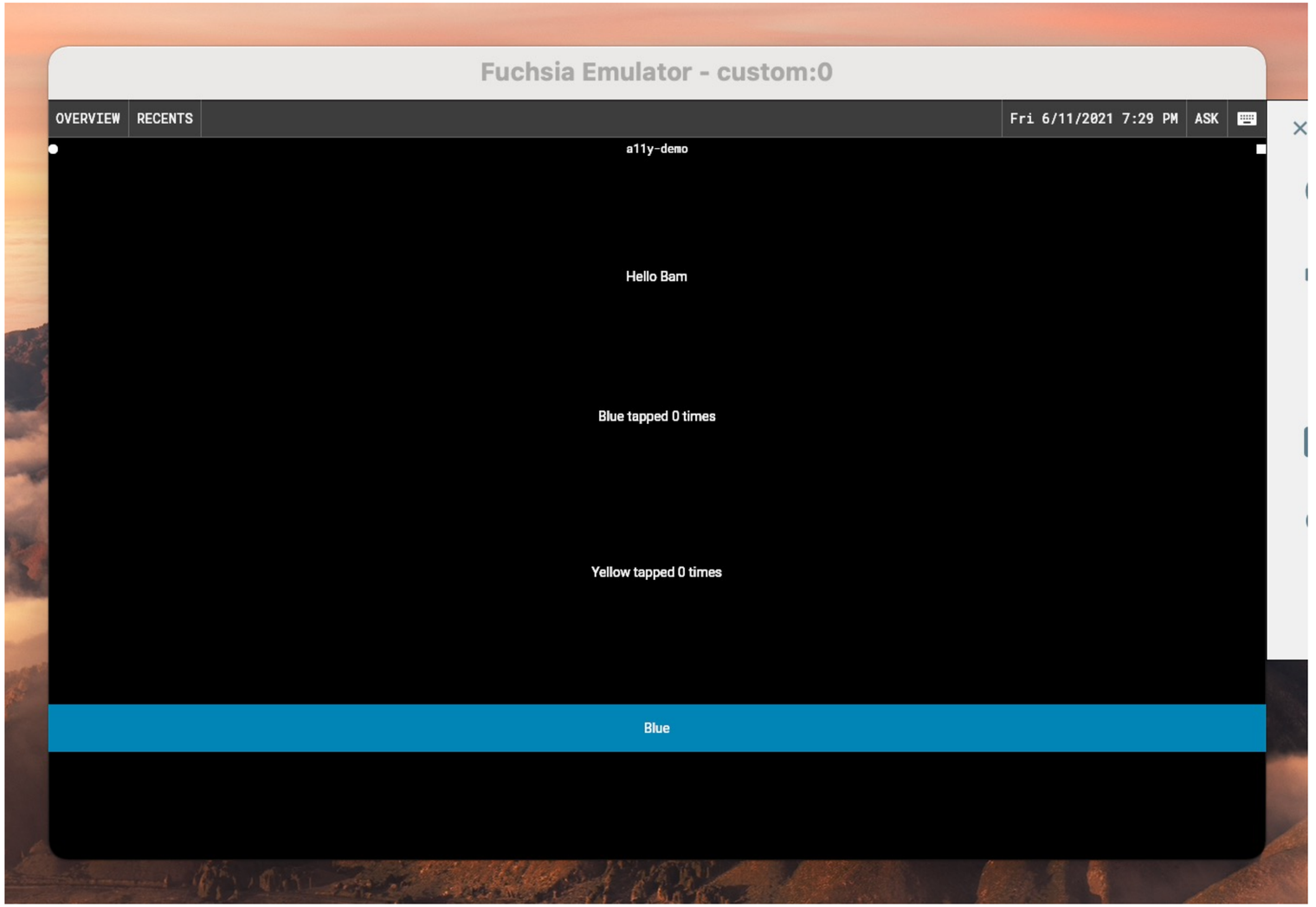
Для использования зависимостей из pub.dev нужно

- Перечислить их в специальном файле в дереве исходников
- Выполнить скрипт, который заберет код из pub и переложит его в нужное место в исходниках ОС
- Пересобрать свой компонент

Понятно, что большинство библиотек из pub не будут работать в Fuchsia, но что-то точно будет

Я пока не смог научиться делать federated plugins для Fuchsia, но думаю, что попробую с этим разобраться

Что получится в итоге?



05

Заключение

Итог

- Fuchsia достаточно интересный проект, с которым можно повозиться в свободное время
- Практического применения в ближайшие пару лет этому точно не будет
- Писать UI на Fuchsia можно не только на Flutter, но и на остальных поддерживаемых языках. Правда, на Flutter гораздо удобнее
- Вокруг неё появляются всякие прикольные проекты типа dahliaOS и Pangolin

Яндекс

Гена Евстратов