**1. Особенности мобильных операционных систем.**

Учет более жестких ограничений по памяти и быстродействию процессора мобильных устройств.

Учет особенностей экранов и устройств ввода

Учет батарейного питания

Совместимость с основными форматами файлов

Мультимедийные возможности: рисунки, видео, аудио, обмен мультимедийными сообщениями.

Поддержка коммуникационных и сетевых технологий

**4. Виды виртуальных машин в Android, их отличия.**

ART vs Dalvik

Ahead-of-time (AOT) компиляция.

Улучшенная сборка мусора.

Встроенная трассировка приложения.



**5. Компоненты Android-приложения.**

Activity представляет собой один экран с пользовательским интерфейсом. Как правило, приложение содержит как минимум одну Activity.

Служба, Сервис (Service) представляет собой компонент, который работает в фоновом режиме и выполняет длительные операции, связанные с работой удаленных процессов.

Поставщик контента (Content provider) управляет общим набором данных приложения.

Приемник широковещательных сообщений (Broadcast receiver) представляет собой компонент, который реагирует на объявления распространяемые по всей системе.

Intent представляет собой объект обмена сообщениями, с помощью которого можно запросить выполнение действия у компонента другого приложения.

Widget (Виджет) представляет собой мини-приложения, которые могут быть встроены в другие приложения (например, домашний экран), и получать периодические обновления. При нажатии на виджет может запускаться основное приложение.

**6. Языки программирования для создания Android-приложений. Среда разработки.**

Java, C#, Python, Kotlin, C/C++.

Android Studio, XCode, Xamarin Studio, IntelliJ Idea …

**7. Языки программирования для создания iOS-приложений. Среда разработки.**

Swift, JavaScript, C#, Objective-C …

Для того чтобы разрабатывать приложения под iOS, раньше обязательно был необходим компьютер Mac или соответствующая виртуальная машина, запущенная на компьютере Windows. Однако на конференции Build 2017 Microsoft представила инструмен для создания приложений под iPhone и iPad на Windows: Xamarin Live Player.

**8. Файл манифеста в Android.**

Файл манифеста (AndroidManifest.xml)

Для запуска компонента приложения системе Android необходимо знать, что компонент существует. Для этого она читает файл AndroidManifest.xml приложения. В этом файле, который должен находиться в корневой папке приложения, должны быть объявлены все компоненты приложения.

Помимо объявления компонентов приложения, манифест служит и для других целей, среди которых:

указание всех полномочий пользователя, которые требуются приложению, например разрешения на доступ в Интернет или на чтение контактов пользователя;

объявление минимального уровня API, требуемого приложению, с учетом того, какие API-интерфейсы оно использует;

объявление аппаратных и программных функций, которые нужны приложению или используются им, например камеры, службы Bluetooth или сенсорного экрана;

указание библиотек API, с которыми необходимо связать приложение (отличные от API-интерфейсов платформы Android), например библиотеки Google Maps ;

**9. Ресурсы Android-приложений.**

Иконки; Изображения; Звуки;

Лейауты; Константы; …

**10. Процесс создания кода для Android.**

кар

**11. Жизненный цикл Activity.**

кар

**12. Отличия ядра Android от ядра Linux. Андроидизмы.**

кар

Андроидизмы

Anonymous Shared Memory (ASHMem)

Binder

Logger

The ION Memory Allocator

Low Memory Killer

Sync driver

Timed Output and GPIO

Wakelocks

**13. Фреймворки в Android.**

кар

**14. Основные разделы диска в Android.**

|  |  |  |
| --- | --- | --- |
| **Раздел** | **Формат** | **Описание** |
| **boot** | bootimg | Содержит ядро и RAM диск для загрузки по умолчанию |
| **cache** | Ext4 | Служит для обновления и восстановления. В Андроиде /caсhe |
| **recovery** | bootimg | Загрузка-для-восстановления. Содержит ядро и альтернативный RAM диск для загрузке в режиме восстановления |
| **system** | Ext4 | В Андроиде /system. Содержит файлы системы и фреймворков |
| **userdata** | Ext4 или F2FS | В Андроиде /data. Пользовательские данные и конфигурация |

**15. Процесс загрузки ОС Android. Необходимые образы.**

Учет более жестких ограничений по памяти и быстродействию процессора мобильных устройств.

Учет особенностей экранов и устройств ввода

Учет батарейного питания

Совместимость с основными форматами файлов

Мультимедийные возможности: рисунки, видео, аудио, обмен мультимедийными сообщениями.

Поддержка коммуникационных и сетевых технологий

кар

Необходимые образы

Boot Loader

Boot Image. Содержит ядро и RAM диск   
Recovery image. Содержит ядро и другой RAM диск, предназначенный для загрузки в режиме восстановления   
System Partition , Data Partition

Boot Loader

Хотя производители могут использовать свой загрузчик, многие используют Little Kernel.

Функции:

- поддержка базовых функций аппаратного обеспечения; - поиск и загрузка ядра;

- минимальный UI (как правило, текстовый);

- поддержка USB, позволяющая общаться с компьютером по протоколу fastboot;

- поддержка файловой системы;

- поддержка цифровых подписей.

**16. Сервисы уровня ядра (демоны) в Android.**

Daemons – сервисы уровня ядра;

Services – сервисы уровня фреймворков

**Adbd** Сервис поддержки Android Debugger Bridge

**Servicemanager** Ключевой компонент механизма IPC.

**Healthd** Отвечает за контроль «здоровья» устройства (в настоящий момент только заряд батареи)

**Lmkd** Low Memory Killer

**Logd** Поддерживает ведение журналов

**Vold** Поддерживает регулировку громкости

Начиная с Android HoneyComb, поддерживает шифрование системы

**Netd** Контроль сетевых интерфейсов

**Mdnsd M**ulticast DNS сервис. Позволяет устройствам находить друг друга, отправляя запрос по адресу 224.0.0.253

(UDP порт 5353). Пример применения – WiFi direct.

**Raccoon** Сторонний проект, адаптированный Андроид. Служит для поддержки VPN.

**Rild** Radio Interface Layer Daemon. Поддержка возможностей связи. Поставляется производителем аппаратной части

**Surfaceflinger** Основа графического стека Андроид. Объединяет объекты *android.View Surface* для их рендеринга.

**Bootanimation** Служит заменой **surfaceflinger** во время загрузки

**Mediaserver** Ключевой компонент работы с мультимедиа, обеспечивающий как запись, так и воспроизведение. Служит контейнером для сервисов AudioFlinger, AudioPolicyService, CameraService, MediaPlayerService

**Drmserver** Digital Rights Management. Управление правами для защищенного контента.

**Installd** Установка и удаление программ

**Keystore** Сервис хранения ключей

**debuggerd[64]** Сервис получения дампа при крахе системы

**zygote[64]** Один из важнейших сервисов, поддерживающий базовую поддержку всех Android Framework Runtime сервисов

**17. Сервисы в Android.**

**18. Способы обеспечения безопасности в Android.**

Стандартные средства Linux

SELinux

Application Sandboxing

Permissions

Code Signing and Platform Keys

Verified Boot

**19. Разрешения (Permissions) и их виды.**

Виды в зависимости от способа создания:

Встроенные, пользовательские.

Виды в зависимости от уровня защиты:

Normal, Dangerous, Signature, signatureOrSystem

**20. SELinux: Архитектура.**

**кар**

**21. SELinux: Режимы.**

Disabled

Permissive

Enforcing

**22. SELinux: Контекст безопасности.**

Security context (или security label, label) – это строка с 4 полями, разделенными двоеточием:

username:role:type:MLS security range

На сегодняшний день, Android использует u для user и s0 для security range.

Role может быть r для процессов или встроеннная object\_r роль для объектов.

**23. SELinux: Инструменты командной строки.**

**кар**

**24. Структура ядра в ОС iOS. Mach, XNU, Darwin.**

Виды ядер: Монолитное, Микроядро, Гибридное

**Mach:** Абстракция процессов и потоков

Работа с памятью

Диспетчер задач

Межпроцессное взаимодействие

**XNU:**

Darwin =Libs+ XNU

XNU =

The Mach microkernel +

The BSD layer +

libKern +

I/O Kit

**Darwin:**

**кар**

**25. Поддерживаемые файловые системы в iOS.**

**Начиная с iOS 10.3 (27 марта 2017) файловой системой по умолчанию стала APFS (Apple File System).**

кар