Высоцкий Андрей 342к

**Android**

**5.1**

Потоки позволяют выполнять несколько задач одновременно, не мешая друг другу, что даёт возможность эффективно использовать системные ресурсы. Потоки используются в тех случаях, когда одно долгоиграющее действие не должно мешать другим действиям. Например, у нас есть музыкальный проигрыватель с кнопками воспроизведения и паузы. Если вы нажимаете кнопку воспроизведения и у вас запускается музыкальный файл в отдельном потоке, то вы не можете нажать на кнопку паузы, пока файл не воспроизведётся полностью. С помощью потоков вы можете обойти данное ограничение.

Когда запускается компонент приложения и приложение не имеет других запущенных компонентов, Android создает новый процесс для приложения с одним потоком исполнения. По умолчанию все компоненты одного приложения запускаются в одном процессе, в потоке называемом «главный». Если компонент приложения запускается и уже существует процесс для данного приложения(какой-то компонент из приложения существует), тогда компонент запущен в этом процессе и использует его поток выполнения. Вы можете изменить данное поведение, задав разные процессы для разных компонентов вашего приложения. Кроме того вы можете добавить потоки в любой процесс.

Задать отдельный процесс для компонента можно с помощью файла манифеста. Каждый тег компонента(activity, service, receiver и provider) поддерживает атрибут android:process. Данный атрибут позволяет задать процесс, в котором будет выполняться компонент. Также вы можете задать процесс в котором будут выполняться компоненты разных приложений. Также данный атрибут поддерживается тегом application, что позволяет задать определенный процесс для всех компонентов приложения.

Android пытается поддерживать процесс приложения как можно дольше, но когда потребуются ресурсы старые процессы будут вытеснены по иерархии важности.

Существует 5 уровней иерархии важности: (процессы первого уровня из списка будут удалены последними)

1.Процесс с которым взаимодействует пользователь(Foreground process)

К таким процессам относится например: активити с которым взаимодействует пользовать; сервис(экземпляр Service), с которым взаимодействует пользователь; сервис запущенный методом startForeground(); сервис, который выполняет один из методов своего жизненного цикла; BroadcastReceiver который выполняет метод onReceive().

2.Видимый процесс

Процесс, в котором не выполнены условия из пункта №1, но который влияет на то, что пользователь видит на экране. К примеру, вызван метод onPause() активити.

3.Сервисный процесс

Служба запущенная методом startService()

4.Фоновый процесс

Процесс выполняемый в фоновом режиме, который невиден пользователю.

5.Пустой процесс

Отмечу, что в компонентах приложения существует метод onLowMemory(), но полагаться на то, что данный метод будет вызван нельзя, также как нельзя на 100% полагаться на метод onDestroy(), поэтому логику сохранения данных или каких-либо настроек можно осуществить в методе onStop(), который(как уверяют) точно вызывается.

Когда запускается приложение, система создает «главный» поток выполнения для данного приложения, который также называется UI-потоком. Этот поток очень важен, так как именно в нем происходит отрисовка виджетов(кнопочек, списков), обработка событий вашего приложения. Система не создает отдельный поток для каждого экземпляра компонента. Все компоненты, которые запущенны в одном процессе будут созданы в потоке UI. Библиотека пользовательского интерфейса Android не является потоково-безопасной, поэтому необходимо соблюдать два важных правила:

1) Не блокировать поток UI

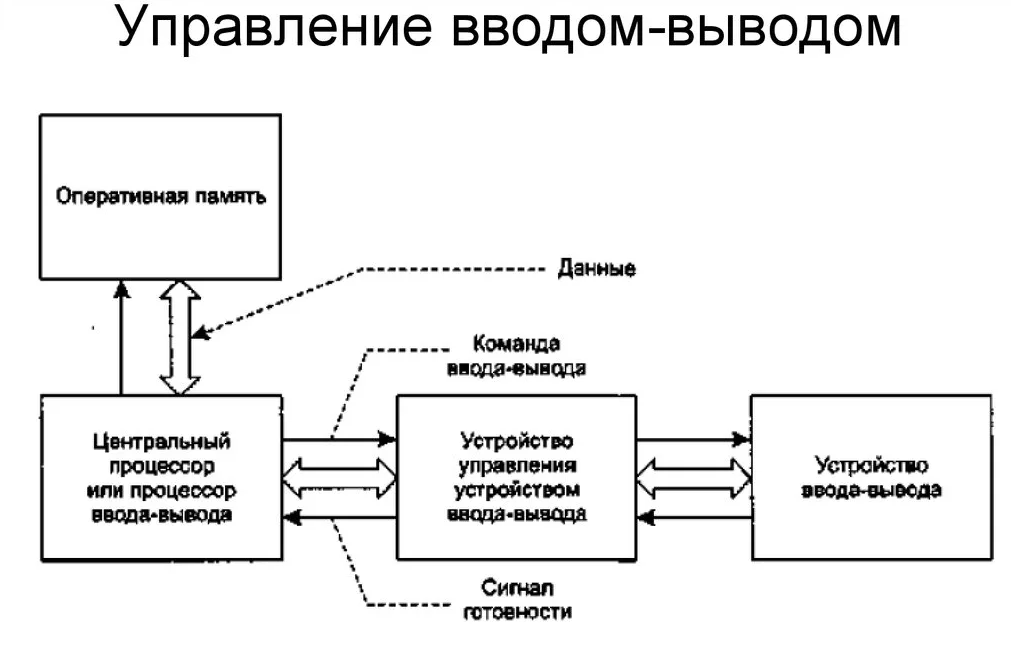
2) Не обращаться к компонентам пользовательского интерфейса не из UI-потока

**5.2**

Среди всех способов, которые помогают понять, как синхронизировать «Андроид» с компьютером, можно выделить подключение через учетную запись Google, используя для этого соединение типа Wi-Fi или Bluetooth, непосредственное соединение через USB-интерфейс, а также использование сторонних программ для управления смартфоном или планшетом.

Раньше невозможно было представить полноценное использование телефона без его регулярного подключения к ПК. Только таким образом на него загружались новые мелодии и картинки. Лишь через USB-кабель передавались фотографии, созданные при помощи встроенной камеры. Также подключение к компьютеру требовалось для обновления прошивки — это касалось не только смартфонов, но и некоторых кнопочных аппаратов. А ещё весьма полезной функцией была синхронизация контактов — только при помощи компьютера можно было перенести телефонную книгу со старого устройства на новое. Что же изменилось сейчас? Пожалуй, практически всё. Аккаунт Google позволяет забыть о компьютере. Контакты, SMS и многие другие данные сохраняются в «облаке», в связи с чем их перенос на ПК рекомендован только тем людям, которые пекутся о безопасности. Равно как давно компьютер не требуется для того, чтобы закачать на Android музыку и изображения. Сейчас это можно сделать при помощи браузера и каких-то сторонних приложений. А обновление операционной системы происходит по воздуху.

**6.1**



**6.2**

Организация ввода-вывода требует наличия информации о ресурсах устройствах и их состоянии. Эта информация обычно хранится в системных таблицах, основными из которых являются: - таблица оборудования ET — eguipment table, список УВВ, подключенных к системе

- таблица виртуальных логических устройств DRT — device reference table

- таблица прерываний IT — interrupt table. Отдельный элемент первой таблицы называется блоком управления соответствующим устройством обмена UCB — unit control block и содержит следующую информацию об устройстве: - тип, модель, имя и характеристики устройства

- характеристики подключения тип интерфейса, порт, линия запроса прерывания

- указание на драйвер управления и адреса его секций

- наличие и адрес буфера обмена для устройства

- уставка тайм-аута и адрес ячейки хранения счетчика тайм-аута

- состояние устройства

- дескриптор задачи, использующий устройство в данный момент времени и т.п.

Вторая таблица DRT избавляет программиста от необходимости знания конкретных параметров устройства. Обращаясь к логическому имени устройства, программист посредством таблицы описания виртуальных логических устройств устанавливает связь с конкретным устройством. Т.о., эта таблица перенаправляет запрос ввода-вывода на те модули и структуры данных, которые хранятся в соответствующем элементе первой таблицы.

Третья таблица IT для каждого запроса на прерывание указывает UCB, сопоставленный этому запросу.

Взаимосвязи названных системных таблиц показаны на рис. 24.

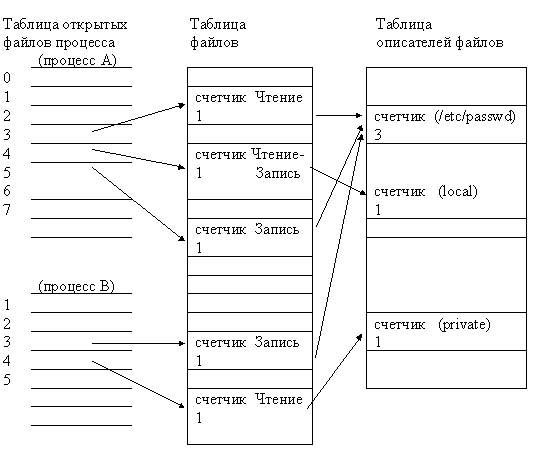


Рис. 24. Взаимосвязь системных таблиц ввода-вывода

Запрос на ввод-вывод от выполняющейся программы поступает на супервизор программ действие 1. Он проверяет вызов на соответствие принятым спецификациям. В случае корректности запроса он перенаправляется в супервизор ввода-вывода действие 2, иначе задаче возвращается соответствующее сообщение действие 1-1.

Супервизор ввода-вывода по логическому виртуальному имени таблицы DRT находит соответствующий элемент UCB в таблице оборудования. Если устройство занято, то описатель задачи помещается в очередь к устройству, иначе запускается препроцессор ввода-вывода. Он формирует последовательность управляемых кодов и данных для устройства ввода-вывода. Когда последовательность сформирована, супервизор ввода-вывода передает управление соответствующему драйверу в секции запуска действие 3. Драйвер инициирует операцию управления, обнуляет счетчик тайм-аута и возвращает управление диспетчеру задач для постановки задачи на процессор действие 4.

По окончании обработки команды ввода-вывода устройство ввода-вывода выставляет сигнал запроса прерывания, по которому управление передается секции продолжения действие 5.

Получив новую команду, УВВ вновь ее обрабатывает, а управление процессором вновь передаётся диспетчеру задач, и процессор продолжает вычисления. Таким образом реализуется параллельная обработка задач, на фоне которой происходит управление вводом-выводом.

**6.3**

Файловая система представляет собой своего рода структуру того или иного накопителя. Она отвечает за то, какой максимальный объем файла можно сохранить в памяти, длину их названий, а также за организацию циклов записи или чтения. Файловые системы выступают в роли посредников между программным кодом и физическим вместилищем.

Для разных операционных систем и типов устройств существуют отдельные ФС и гаджеты на ОС Android, не являются исключением.

1. Yaffs и Yaffs2

Создателем файловой системы Yaffs (Yet Another Flash File System) является Чарльз Мэннинг, родом из Новой Зеландии. Данная ФС предназначается для организации работы флеш-накопителей и оперативной памяти. Основным ее преимуществом является повышение сроков эксплуатации модулей памяти, так как система автоматически пропускает ячейки, предназначенный для однократной записи. Yaffs2 использовалась для организации работы внутренней памяти гаджетов Android версии 2.2 и 2.3.

2. VFAT

Vfat не является полноценной самостоятельной файловой системой, а представляет собой расширение FAT. Доработанная версия позволяет сохранять файлы с длинными именами, но по всем остальным характеристикам является морально устаревшей. Vfat может использоваться на картах памяти, а операционные системы на базе ОС Андроид полностью их поддерживают. В основном, это файловая система флешки android.

3. F2FS

F2FS (Flash Friendly File System) – файловая система, которая предназначена в первую очередь на работу с флеш-памятью и SSD-накопителями. Разработал ее сотрудник компании Samsung, Ким Чэ Гык, а после публикации исходного кода она была доработана другими инженерами компании. F2FS может использоваться на картах памяти SD/MMC, а также со множеством других типов памяти. Для полноценной кастомизации существует целый набор утилит. Из преимуществ можно отметить хорошую гибкость в настройке, высокие показатели сохранения жизненного цикла блоков памяти, а также хранение данных в виде журнала. Хорошую скорость работы обеспечивает то, что индексы данных сохраняются в оперативную память, а поддержка F2FS включена в ядро Linux, начиная с версии 3.8.

4. Ext2-Ext4

Ext2-Ext4 – основные файловые системы Android. Именно они используются для организации работы внутреннего хранилища на большинстве современных гаджетов, и если первые устройства работали под версиями Ext2, то начиная с версии Android 4, основными стали Ext3, а потом и Ext4. Основное отличие между вариациями заключается в наличие журналирования. То есть, если в процессе записи или чтения данных происходит системный сбой, например, неожиданное отключение питания, не произойдет потери или повреждения данных. Несмотря на то, что в основном ФС формата Ext используется в основном в блочных накопителях, пользователи могут установить данный тип и для карт памяти, но без сторонних утилит получить к ним доступ из операционных систем, помимо Linux, будет невозможно. Файловая система флэшки Android обычно форматируются в FAT (VFAT) или NTFS, а флеш-память — в Ext3 или Ext4.

5. UBIFS

mSATA SSD 16 GB Sandisk — SDSA3DD-016G

UBIFS – файловая система, предназначенная исключительно для памяти по типу NAND (флеш-накопители, применяемые на мобильных устройствах). Ее основное преимущество — это снижение износа носителей данных. Состоит такая ФС из двух слоев — UBI (отвечает за работу и связь с физическим носителем) и UBIFS (сама файловая система). Разработчиком UBIFS является компания Nokia, но встретить подобную файловую систему можно не только на оригинальных устройствах от данного производителя, но и на других гаджетах, например, китайского изготовления.

6. Samsung RFS

Samsung RFS – разработанная корейской компанией Samsung файловая система для устройств на базе ОС Linux, а одной из разновидностей последнего является Android. Предназначается фирменная ФС для флеш-памяти NAND и используется во многих гаджетах собственного производства. Для облегчения работы с файлами используется таблица формата FAT, что позволяет максимально просто произвести запись файлов на флэш-память и их чтение. Специфика RFS учитывает особенности NAND накопителей, что позволяет увеличить длительность их эксплуатации, а также снизить вероятность потери данных при системных сбоях и случайных отключениях питания.

7. SDCardFS

Компания Google долгое время использовала в своих устройствах файловую систему FUSE, но она имела ряд недостатков, среди которых необходимо отметить малое количество памяти, выделяемой под приложения под приложения, а также большое потребление хранилища при кэшировнии данных. На замену FUSE пришла SDCardFS, которая решала эти проблемы, но представляла собой своего рода оболочку, эмулирующую FAT32. Она отправляет данные файловым системам нижнего уровня, что позволяет оптимизировать работу смартфонов, а также исправляет ошибки, связанные с передачей данных на компьютер и обратно. SDCardFS по своей сути не является классической файловой системой, но значительно сказывается на скорости работы смартфонов или планшетов на Android.

**7.1**

В области чипсетов ARM, что является общим фактором, весь стек Android, начиная с почти идентичного ядра, основанного на Linux, на самом деле является 32-битным, кросс-компилируется обычно либо из 32-битной/64-битной хост-среды, либо из хост-среды. обычно является одним из дистрибутивов Linux. Рекомендуемый Google дистрибутив для сборки и кросс-компиляции Android — Ubuntu .

Библиотека времени выполнения Android (мультимедиа, графика, файловая система и многие другие) также являются 32-битными, но по мере того, как мы достигаем уровня dalvikvm, количество битов становится неактуальным, поскольку на данный момент загружаются apks. из Google Play Store являются собственным байт-кодом («побочный продукт» сгенерированного кода Java, скомпилированного в переносимый байт-код), который предназначен для DalvikVM (виртуальная машина), которая, в свою очередь, интерпретирует и переводит байт-код, предназначенный для необработанного набора инструкций ARM.

Froyo был последним Android, который позволял компилировать в 32-битной размещенной среде, в которой он был кросс-компилирован для чипсета ARM.

Gingerbread был первым «будущим» Android, тогда около трех лет назад, который ввел требование использовать 64-битную размещенную среду, в которой он был построен. Было много хаков, чтобы заставить Gingerbread быть собранным в 32-битной среде хостинга.

Для ICS и JB и выше теперь определенно требуется 64-битная среда для ускорения компиляции и сокращения времени на сборку.

Системные регистры поддерживают РВА процессоров 286/386+ и включают:

GDTR – содержит адрес таблицы глобальных дескрипторов;

IDTR – содержит адрес таблицы дескрипторов прерываний;

LDTR – содержит селектор таблицы локальных дескрипторов;

TR – содержит селектор сегмента состояния задачи TSS.

Регистры GDTR и IDTR являются общими для всех задач системы. До перевода процессора в РВА необходимо создать в ПАМЯТИ таблицы GDT и IDT и соответственно инициализировать регистры GDTR и IDTR.

Регистры LDTR и TR являются специфичными для каждой задачи. Таблица LDT адресуется селектором (записанным в регистр LDTR), который выбирает дескриптор в таблице GDT, и загружает его в соответствующий теневой регистр. Сама таблица LDT может находится на диске и загружается в память компьютера после записи селектора в регистр LDTR. Во время переключения задач можно заменить все локальное адресное пространство простой перезагрузкой селектора в регистр LDTR.

**7.2**

1.«Режим полёта»

С этим режимом знаком практически каждый владелец современного телефона. Кто хоть раз пользовался услугами авиакомпаний, знает, что при взлёте самолёта пассажиров просят отключить все мобильные устройства и пристегнуть ремни. Однако активация этого режима отключает все существующие способы передачи данных, такие как Wi-Fi, GPS, GSM, GPRS, Bluetooth и прочие, которые могут создавать помехи на радарах и других приборах самолёта, и позволяет пользоваться устройством на протяжении всего полёта

Кроме этого предназначения, у режима есть ещё одно, мало кому известное. Если Вам требуется быстро зарядить устройство, включите «Режим полёта» и зарядка ускорится практически в 2 раза

Активировать режим можно в настройках устройства, или зажав кнопку отключения питания

2.«Беззвучный режим» и «Режим вибрации» на Android

С предназначением этих двух режимов всё понятно и без слов. Активация первого отключает полностью звук на устройстве, активация второго отключает звук, но оставляет вибрацию

Применить эти режимы можно зажав кнопку отключения питания.

3. «Режим энергосбережения»

Этот режим помогает существенно сэкономить заряд батареи. Его активация снижает яркость экрана, закрывает все неиспользуемые процессы и службы, убирает анимацию при переходах между рабочими столами, что снижает нагрузку на процессор и соответственно замедляет расход заряда аккумулятора. Перед длительной поездкой рекомендуется включать этот режим

Активировать режим можно в настройках устройства, в разделе «Батарея»

4.«Спящий режим»

Это стандартный режим, который автоматически активируется, если устройство какое-то время не используется. Происходит подобное с целью экономии заряда аккумулятора, однако в отличие от режима энергосбережения, спящий режим не закрывает открытые в фоне процессы и после «пробуждения» устройства Вы сможете продолжить работу в приложении, которое было запущено

Если зайти в настройки устройства, потом в раздел «Экран» и нажать на пункт «Спящий режим», то Вы сможете задать параметр времени, по истечении которого телефон будет автоматически переведён в спящий режим. Стандартом является 1-2 минуты, но по желанию Вы можете увеличить время до 1 часа или вовсе отключить режим сна

5.«Безопасный режим»

С этим режимом многие уже успели познакомиться ещё при пользовании персональным компьютером с операционной системой Windows. Однако до сих пор многие люди не знают, что такой режим доступен не только на ПК, но и на мобильных устройствах с ОС Android

Данный режим на мобильных устройствах, как и на компьютере, служит для устранения разного рода неполадок в операционной системе. Продвинутые пользователи Android знают, что разработчики постарались на славу и предоставили им возможность полностью перестраивать систему под себя, менять её внешний вид и функционал, что зачастую приводит к множественным программным ошибкам

При загрузке телефона в безопасном режиме активируются только системные службы, а все сторонние приложения и изменения в системе блокируются. Благодаря этому пользователь может удалить неисправную программу, которая мешает корректной работе устройства, перезагрузить телефон в нормальный режим и пользоваться им как раньше

Чтобы перейти в безопасный режим, нужно зажать кнопку питания и дождаться появления на экране меню выключения. После появления меню на пункте «Отключить питание» нужно удерживать палец до тех пор, пока не появится сообщение о переходе в безопасный режим

Для выхода из безопасного режима достаточно просто перезагрузить устройство

6.«Режим TalkBack»

TalkBack — отдельная служба, созданная для людей с плохим зрением, однако её вполне можно отнести к режимам телефона. При её активации телефон переключается с точечного управления на жесты и каждое действие, совершённое пользователем, будет озвучиваться встроенным голосовым роботом. Рядовому пользователю эта служба ни к чему, но если Вы имеете проблемы со зрением, с помощью этого режима Вы сможете управлять телефоном, опираясь только на свой слух

Активировать режим можно в настройках устройства. Для этого зайдите в меню «Специальные возможности» и передвиньте ползунок на пункте «TalkBack»

**7.3**

Операционная система

Минимальная версия ОС

Android 12.0

Минимальные требования к устройствам

ОЗУ

3 ГБ

Хранилище

32 ГБ

Частота процессора

1,4 ГГц

Архитектура

64-битная

Пользовательский интерфейс

Стандартные системные приложения – устройства с рабочими профилями

Контакты

Загрузки

Google Play

Приложение EMM (если включено в системный образ)

Стандартные системные приложения – полностью управляемые устройства

Камера

Контакты

Загрузки

Google (если включено в образ системы)

Сообщения

Телефон

Google Play

Настройки

Приложение EMM (если включено в системный образ)

Другие приложения

Контакты

Телефон

Настройки и т. д.

Разрешения на установку приложений

Приложения, не связанные с Google Play и контроллером политик EMM, можно устанавливать в рабочий профиль или на полностью управляемое устройство только с согласия Google.

Одобренный брендинг

Стандартные элементы оформления и строки

Требования к инициализации

Устройства должны предусматривать стандартную процедуру инициализации в AOSP

Сервисы

Поддержка обновлений системы безопасности

Производители устройств должны опубликовать на своем сайте информацию об обновлении системы безопасности и указать ссылку на нее в бюллетенях по безопасности Google. В частности, нужно указать:

Дату (месяц и год), до которой устройство гарантированно будет получать обновления системы безопасности.

Исправления, содержащиеся в обновлении системы безопасности, включая любые исправления, относящиеся к конкретному производителю.

Поддержка крупных обновлений

Поддержка текущей версии ОС и одного крупного обновления

Информация о том, до какой версии операционной системы Android будет гарантированно обновлено устройство

Повторная проверка

Чтобы устройство продолжило участвовать в программе Android Enterprise Recommended, любое крупное обновление операционной системы должно соответствовать всем требованиям программы.

OEMConfig

Производители устройств могут осуществлять поддержку корпоративных API, разработанных специально для собственных продуктов, например правила для устройств. Они должны подчиняться перечню условий совместимости и требованиям мобильных сервисов Google. Такими API необходимо управлять в приложении OEMConfig.