Native Libraries
Lecture 10

Operating Systems Practical

7 December 2016

Bionic

SQLite

SSL

WebKit

Surface Manager

Audio Manager

Media Framework

OpenGL ES

Bionic

SQLite

SSL

WebKit

Surface Manager

Audio Manager

Media Framework

OpenGL ES

- Custom C library
- Replacement for glibc
- Not POSIX compliant
- Goals
    - BSD license
        - Avoid GPL and LGPL in userspace
    - Small size
        - glibc is very large
    - High speed
        - Designed for low CPU frequencies

- ► Bionic routines do not throw, pass and handle C++ exceptions
  - ► Support for exceptions adds a great overhead
  - ► Exceptions can be used internally in C++ code, if they do not cross the libc routine
- ► No C++ Standard Template Library
  - ► Native code developers can use gnustl or stlport
- ► New pthread implementation
- ► No wchar_t and no support for LOCALE

- A shared memory region is used for configuration
  - Also used by the applications through property_get() and property_set()
- No openlog() or syslog()
  - __libc_android_log_print() used for logging
- Specific malloc implementation
  - A hash table used for tracking allocations to discover leaks
- No pty support

- No AIO (`aio_read()`, `aio_write()`)
- No `crypt()`
  - Includes MD5 and SHA1
  - Developers should use OpenSSL for crypto
- Android has its own account management
  - Does not use /etc/passwd, no `getpwent()`
  - `getpwnam()` and `getpwuid()` wrappers that use Android ID service
- `getprotobyname()` only prints "FIX ME! implement getprotobyname()"
  - Bionic not finished

- Mutexes, rwlocks, condvars implemented using kernel futexes
- No `pthread_cancel()`
  - Threads cannot kill other threads
- No `pthread_atfork()`
- Thread local storage (TLS) implemented
  - Max 64 keys
  - Id and errno reserved
  - Mapped at 0xffff0ff0 in the process (on ARM)
- `pthread_attr_{set,get}inheritsched` and `pthread_attr_{set,get}scope` not implemented

ӘSP
logcat crunch

- Developed for embedded systems
- Used by Android, Apple iOS, Blackberry
- Low memory consumption
  - Starts from 50 KB, reaches few hundred KB
- Ease of use
  - No configuration files or complex commands
- Free
  - Released under public domain
  - Supported by Google, Adobe, Mozilla
  - Active maintenance

OSP

- ▶ Is serverless
  - ▶ No process
  - ▶ Access database through library
  - ▶ No port configuration, no user adding, no access levels
- ▶ All data in a single file
  - ▶ Indices, triggers, metadata
  - ▶ Journal file added
- ▶ Fewer datatypes
  - ▶ `INTEGER, REAL, TEXT, BLOB`
  - ▶ No datetime -> string
  - ▶ No boolean -> 0 and 1
  - ▶ Blob not recommended on Android -> files in the file system

- ▶ Does not use static typing
  - ▶ Type depends on the inserted value
  - ▶ Not on the definition in CREATE TABLE
  - ▶ Create INTEGER column and insert TEXT
  - ▶ Manifest typing -> between static and dynamic typing
- ▶ No fixed column length
  - ▶ No limit to the inserted value
  - ▶ Constraints in the code
- ▶ Database files are cross-platform
  - ▶ Pull file from device, use it on laptop
  - ▶ Open file with GUI tools
  - ▶ Populate file on laptop, push on device

ƏSP
logcat crunch

- ▶ Generated from OpenSSL external project
- ▶ Includes *libcrypto* and *libssl* libraries
- ▶ Libcrypto implements cryptographic algorithms
  - ▶ Symmetric ciphers
  - ▶ Public key crypto
  - ▶ Certificates
  - ▶ Authentication codes, hashes
- ▶ Libssl implements SSL/TLS for secure communications
- ▶ Cryptographic routines from libcrypto are used by libssl

OSP
logcat crunch

- ▶ Open source web browser engine
- ▶ BSD license but WebCore and JavaScriptCore under LGPL
- ▶ From 2013, WebKit is a trademark of Apple
- ▶ Contributors: Apple, Nokia, Google, Samsung, Adobe, Intel, etc.
- ▶ Ports: Safari, Chrome, Android, Blackberry, Tizen, etc.
- ▶ Originally forked from KHTML and KJS libraries from KDE
- ▶ Google forked WebCore (WebKit component) into Blink

- ▶ Speed, power and versatility
- ▶ Support for new features in CSS3, HTML5 and JavaScript
- ▶ WebCore
  - ▶ LGPL license
  - ▶ C++ based
  - ▶ Layout, rendering, DOM library for HTML and SVG
  - ▶ Creates HTML visualizations based on DOM markup and CSS
- ▶ JavaScriptCore
  - ▶ JavaScript engine
  - ▶ Derived from KJS library in KDE and PCRE regular expression library
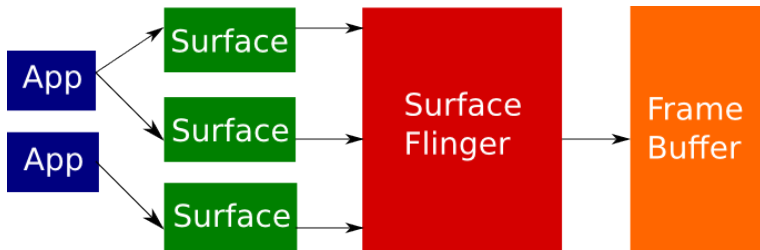 - ▶ High performance interaction engine

OSP
logcat crunch

- a.k.a. Surface Flinger
- Part of System Server
- System-wide surface composer
- Handles surface rendering to frame buffer device
- Combines 2D and 3D surfaces, surfaces from multiple apps
  - Each app generates 1 or more surfaces
  - The Surface Flinger composes these surfaces
  - The result is stored in the Frame Buffer

- ▶ Surfaces passed as buffers using Binder IPC calls
  - ▶ Surface instances written to or restored from Parcels
- ▶ Use OpenGL ES or 2D hardware accelerator for the composition
- ▶ Double buffering using page flip
  - ▶ Front buffer (primary surface) and back buffer
  - ▶ Change primary surface pointer and back buffer pointer
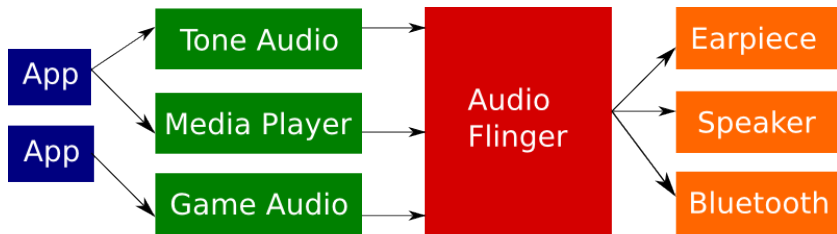 - ▶ Screen never flickers or displays artifacts

OSP
logcat crunch

- ▶ a.k.a. Audio Flinger
- ▶ Part of Media Server
- ▶ Manages the audio output devices
- ▶ Receives and combines multiple audio streams (tone, media player, games)
- ▶ Directs audio to various outputs (Headphones, Speaker)

ƏSP
logcat crunch

- ▶ Uses StageFright engine for audio/video recording and playback
- ▶ Default software codecs
- ▶ Implement your own hardware codec
  - ▶ Using OpenMAX Integration Layer (IL)
- ▶ OpenMAX IL
  - ▶ Standardized way for Stagefright to recognize and use hardware codecs
  - ▶ OpenMAX plugin - shared library
  - ▶ Implemented according to the standard
- ▶ Accessed through android.media API

OSP

- Manage 2D and 3D graphics on embedded systems
- Interface between software and graphics acceleration hardware
- Low-level, lightweight API
- Subset of OpenGL
- Display complex 3D graphics on mobile
- Easy porting

- Industry standard
    - Open, vendor-neutral, multi-platform
    - Anyone can implement OpenGL ES based on the specification
- Small footprint, low power consumption
    - Minimum data storage requirements
    - Small binary
- Works with both hardware and software rendering
    - Calls to hardware, to software routines or combination of both

- ▶ NDK includes both OpenGL ES 1.x and 2.0 libraries (eventually 3.0)
- ▶ Differ significantly
    - ▶ Different Graphics Rendering Pipelines
        - ▶ Processing stages taken by the graphics hardware to produce graphics
        - ▶ Accepts object description (vertices, primitives, color values)
    - ▶ 1.x fixed function pipeline
        - ▶ Input primitive, texture data, lightning
    - ▶ 2.0 programmable pipeline
        - ▶ Write vertex and fragment shaders using the Shading Language (GLSL)

	1.x	2.0
Performance	Fast 2D and 3D graphics	Faster 2D and 3D graphics
Compatibility	Almost all Android devices	Many Android devices
Ease of coding	Fixed pipeline with convenient functions, easy to use for simple 3D apps	No built-in basic functions, more effort to use for simple 3D apps
Graphics control	Difficult or impossible to create some effects (fixed pipeline)	More direct control of the graphics processing pipeline (programmable pipeline)

- ▶ Use OpenGL ES directly from Android app
- ▶ Create a view container
- ▶ Implement a `GLSurfaceView` and `GLSurfaceView.Renderer`
- ▶ `GLSurfaceView`
  - ▶ View container for the graphics
  - ▶ Draw and manipulate objects using OpenGL API
- ▶ `GLSurfaceView.Renderer`
  - ▶ Controls what is drawn in the view
  - ▶ Defines methods required for drawing
  - ▶ Attach it to the `GLSurfaceView` instance
  - ▶ `GLSurfaceView.setRenderer()`

- ► Call OpenGL API
- ► OpenGL ES 1.0/1.1 API
  - ► `android.opengl` (static interface, better performance)
  - ► `javax.microedition.khronos.opengles` (standard implementation)
- ► OpenGL ES 2.0 API
  - ► `android.opengl.GLES20` (starting with Android 2.2, API level 8)
- ► OpenGL ES 3.0 API
  - ► `android.opengl.GLES30` (starting with Android 4.3, API level 18)
  - ► Requires an implementation of the pipeline provided by the manufacturer
  - ► Not all 4.3 devices support 3.0 API
- ► Declare version requirement in the manifest
  - ► `<uses-feature android:glEsVersion="0x00020000" android:required="true" />`

- Karim Yaghmour, Embedded Android: Porting, Extending, and Customizing, Chapter 1
- `http://codingrelic.geekhold.com/2008/11/six-million-dollar-libc.html`
- `https://www.grokkingandroid.com/sqlite-in-android/`
- `http://trac.webkit.org/wiki`
- `https://androidteam.googlecode.com/files/Anatomy-Physiology-of-an-Android.pdf`
- `http://developer.android.com/guide/topics/graphics/opengl.html`

- Bionic
- SQLite
- SSL
- WebKit

- Surface Manager
- Audio Manager
- Media Framework
- OpenGL ES