Android Operating System
Presentation

- History and Basics
- OS Performance and Optimization
- Application Lifecycle
Google acquired the startup company Android, Inc. in 2005

Open Handset Alliance formed in 2007

- Sprint Nextel
- T-Mobile
- Motorola
- Samsung
- Sony Ericsson
- Toshiba
- Vodaphone
- Google
- Intel
- Texas Instruments
Open Handset Alliance

- Innovate rapidly and respond better to consumer needs

- Key outcome was the Android Platform
  - Designed to serve mobile operators, handset manufacturers, and application developers
  - Based on Linux kernel version 2.6
  - Source code licensed under Apache License, Version 2.0
Android Software Stack

Hardware Abstraction Layer

- Linux
- Provides Memory Management, Security, Proven Driver Model, Networking

Written in C/C++
Core Power of Andriod Platform

Memory Management, Security, Proven Driver Model, Networking

Display Library
- 3D Graphics
- 2D Graphics

Media Codec
- Render Fonts

Database Storage
- Open Source Browser Engine

Highly Optimized Bytecode Interpreter
Written in Java

Toolkit all applications use

Manages Lifecycle of Applications

Maintains Common Backstack

Tracks Applications installed on the device and their static privileges

Standard Java abstraction

Contains Buttons, Layouts, Lists UI

XMPP Service

Allows any app to send data to any other Android user

Allows Alerts to be propagated throughout the device

Standard Java Window Manager abstraction, utilizes Surface Manager API that the Telephone application is built on

External file storage

Contains Buttons, Layouts, Lists UI

All Location Services

All

XMPP Service

Allows any app to send data to any other Android user

Allows Alerts to be propagated throughout the device

XMPP Service

Allows any app to send data to any other Android user

Allows Alerts to be propagated throughout the device
Applications

- Written in Java, translated to .dex files, and executed in their own VM

- Each application runs as its own process
  - Restricts possible effects of applications on one another
    - Security
    - CPU intensive applications
Presentation

- History and Basics
- OS Performance and Optimization
- Application Lifecycle
Android Platform

- Fully Featured
  - Android SDK supports Java Standard Edition except for AWT and Swing
    - Android provides own modern UI framework

- Extensive features of Android libraries use as much as 10 MB even with an optimized JVM
  - Solution: Dalvik VM
Register-based architecture rather than traditional JVM-stack-based machine

- Up to 30% fewer instructions as a result

Combines Java class files into one or more Dalvik Executable files (.dex)

Dalvik Executables reuse duplicate information from multiple class files

- Shared constant pool of java class files
  - Reducing space requirement by ½ of a traditional .jar file
Example Java Source Code

```java
public interface Zapper {
    public String zap(String s, Object o);
}

public class Blort implements Zapper {
    public String zap(String s, Object o) {
        ...
    }
}

public class ZapUser {
    public void useZap(Zapper z) {
        z.zap(...);
    }
}
```
Java Class Files
Register-based Architecture

- Each DEX method has its own number of virtual registers
- The Dalvik VM interpreter manages method registers with an internal execution state stack
  - Current method’s registers are always on the top stack frame
- All computation occurs on registers
  - Higher semantic density per instruction
Dalvik VM

- Fine-tuned garbage collection
- Android 2.2 included a just-in-time compiler
- Optimized to allow devices to run multiple Dalvik VMs at once
- Runs .dex files (cannot run java class files directly)
Presentation

- History and Basics
- OS Performance and Optimization
- Application Lifecycle
Application Lifecycle

- Provide seamless transition process between applications and views
  - Must deal with limited memory
Application Lifecycle

System Process

Home

Mail

Browser

Mail List
Message
Browser
Map