Analysis and Comparison with Android and iPhone Operating System

Lu Cheng

1 System Development History

1.1 History of Android

Android is not only a mobile operating system that uses a modified version of the Linux kernel [1][2][3], but also a software stack for mobile devices that includes an operating system, middleware and key applications.

In July 2005, Google bought Android, Inc., a tiny startup company based in Palo Alto, California, USA. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel and their main marketing targets were hardware component and software partners with an easy and flexible operating system. On the November 5th in 2007, several companies including Google, HTC, Motorola, Intel and so on came together to form the Open Handset Alliance and stated the goal for developing open standards for mobile devices, meanwhile unveiled the new product: Android, a mobile device platform built on the Linux kernel version 2.6[4]. Android has been available as open-source software since October 2008. Under the Apache License, private companies could add on their own applications and extensions and sell them, without having to submit them to the open-source community [5]. Since its original release, Android has seen a number of updates with fixed bugs and new features. Cupcake (the official 1.5 update), which based on Linux kernel 2.6.27 was released on 30 April 2009. Cupcake improves the ability to record and watch videos with the camcorder mode and enable uploading videos to YouTube and pictures to Picasa directly from the phone. The Cupcake also supports Bluetooth A2DP and can connect to a Bluetooth headset within a certain distance. On 15 September 2009, Donut (the 1.6 SDK) was released. Donut improved Android Market experience and provides an integrated camera, camcorder and gallery interface. It also updated Voice Search to have a faster response time deeper integration with native applications. Search experience and technology support are also updated. The updated version 2.0, Eclair was released on 26 October 2009. It optimized the hardware speed and support for more screen sizes and resolutions. Eclair introduced new browser UI and new contact lists, and provides better white/black ratio for backgrounds. Besides, it built in flash support for Camera and enables the digital zoom. The subsequent version 2.1 is named FroYo and the next update after FroYo will be called Ginerbread.

1.2 History of iPhone

iPhone operating system is a mobile operating system developed and marketed by Apple Inc. It is the default operating system for the iPhone, the iTouch and the iPad. The iPhone OS was derived from Mac OX X and the version history of iPhone OS began at June 29, 2007 with the release of the iPhone [6]. iPhone OS had no official name until the first beta version of the iPhone SDK released in March 6, 2008. Before then, Apple marketing literature simply stated the iPhone uses OS X. The initial version of iPhone was released on June 29, 2007. The version 1.0.2 was initially released on iPod Touch on September 14, 2007. Version 1.1.1 updated the interface for the calculator application, supported the TV out and provided adjustable volume and louder speakerphone. Later the version 1.1.2 was released on November 12, 2007. It provides the battery charge level indicator and international language & keyboard support. Version 1.1.3 provides more feature then previous versions. It added the mail, maps, stocks, weather and notes applications for iPod touch, enable lyrics support for music and chapter support for movies. Besides, it increased the SMS storage capacity from 1,000 to 75,000 and gains the new feature on Google Maps, the 'Locate Me' can determine the phone's approximate location. Version 1.1.4 and 1.1.5 was released on February 26, 2008 and July 15, 2008 respectively. They fixed bugs and improve the speed of interface. The version 2.0 was available with the release of the iPhone 3G on July 11, 2008. It enhanced the operating system, making it able to turn Wi-Fi back on in Airplane mode and support SVG and Cisco IPsec VPN. 2.0 also enhance the language by providing extra keyboard languages and support the traditional and simplified Chinese handwriting recognition. On September 9, 2008, Apple Inc. released the version 2.1. It fixes the bugs and enhances the system performance. It provides new OS features like changed 3G, EDGE and GPRS indicators, turn off the camera within the restrictions page. Besides, version 2.1 provides the new application features like more information within iPod music lists and audiobooks. The version 2.2 which released on November 21, 2008 enhances the performance on maps and mail, improved the stability and performance of Safari. It also improved the performance on many aspects like the sound quality of visual voicemail message and so on. On June 17, 2009, Apple Inc. released the version 3.0 with the iPhone 3GS. Current version is 3.2 which published on January 27, 2010. It supports the newly announced iPad and the landscape springboards, added terrain view in maps application.

2 System Architecture

2.1 Android Architecture

The Android architecture has several layers as showed in Figure 1.

Kernel: Linux is support for core system services such as security, memory management, process management, network stack, and driver model [7]. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

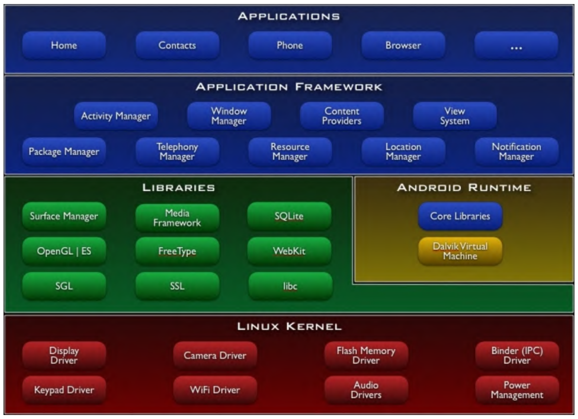

Figure 1. Android Architecture

Runtime: Runtime includes core libraries and Dalvik virtual machine. Core libraries have a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Libraries: Android has a set of C/C++ libraries used by various components of the Android system. These libraries are exposed to developers. The system C library is a BSD-derived implementation of the standard C system library. The media libraries are based on PacketVideo's openCORE which support playback and recording of many popular audio formats. The surface manager manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.

Application Framework: All Android applications are written with Java programming language, it ships with a set of core applications including email client, SMS program, calendar, maps, browser, contacts and others. Android offers developers the ability to build various applications with an open development. Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities. This same mechanism allows components to be replaced by the user. The applications including views that can be used to build an applications, including lists, grids, text boxes, buttons, and even an embeddable web browser. Content providers enable applications to access data from other applications or to share data with others. The resource manager providing access to non-code resources such as localized strings, graphics and layout files. The notification manager enable all applications to display custom alter in the status bar. The activity manager manages the lifecycle of applications and provides a common navigation backstack.

2.2 iPhone Architecture

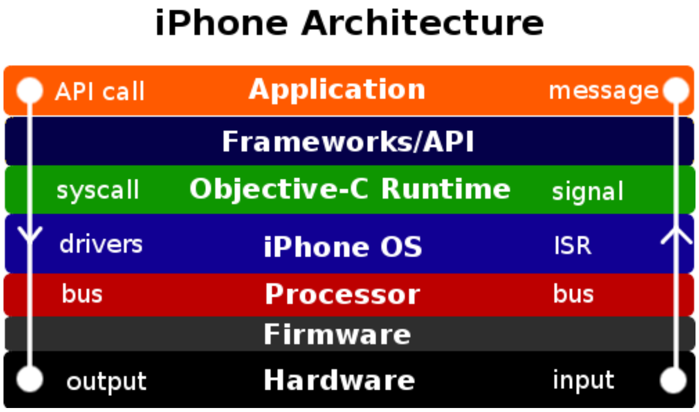

Figure 2. iPhone Architecture

Hardware: In iPhone, Hardware refers to the physical chips soldered to the iPhone's circuitry. The actual processor falls under this layer, but the instruction set and in-memory descriptor tables are contained within the "processor" layer.

Firmware: Firmware refers the chip-specific code that is either contained with memory in/around the peripheral itself, or within the drive for said peripheral.

Processor: Processor is refers to the ARM instruction set and the interrupt descriptor table as set up by the iPhone OS during boot and driver initialization.

iPhone OS: iPhone OS is the kernel, drivers, and services that comprise of the iPhone Operating System. It sits between the user space and hardware.

Objective-C Runtimes: Objective-C runtime is comprised of both the Objective-C dynamically-linked runtime libraries, as well as the underlying C libraries.

Frameworks/API: Frameworks/API layer has API calls which are Apple-distributed headers with the iPhone SDK, with some dynamic linking occurring at runtime. These reside on top of the Objective-C runtime, as many of these are written in Objective-C.

Application: The application stored in iPhone has to be purchased through the application store. This application was compiled to native code by the Apple-distributed iPhone compiler, and linked with the Objective-C runtime and C library by the linker. The application also runs entirely within the user space environment set up by the iPhone OS.

3 Hardware

3.1 Android

Android is used as mobile operating system for many mobile phones, like Motorola CLIQ, Samsung Behold and so on. DROID by Motorola adopt Android 2.0 as the operating system.

Processor: process in DROID is Arm® Cortex™ A8 which based on the ARMv7 architecture with the speed of 550 mHz. The high-performance of Arm® Cortex™ A8 processor is proven in end devices today as it can meet the requirements for power-optimized mobile devices needing operation in less than 300mW.

Memory and secondary storage: The Motorola Droid has only 256MB of memory on which it can store applications and it cannot run applications off the SD Card. The secondary memory for music, image and general storage installed in DROID is 16 GB microSD, a removable flash memory cards. It also can support up to 32 GB microSD.

Camera: The camera with automatic focus and dual led has 5 MP megapixels and 4x digital zoom, it has the features like image stabilization, real-time color effects, scene modes and location tagging.

Sensor: The sensor in DROID has proximity, ambient light and eCompass.

Screen: The screen is 3.7 inch with 16:9 widescreen with TFT display type.

Battery: The DROID uses 1400 Ah Li Ion battery with up to 270 hours standby time, the continuous usage time is up to 358 minutes.

Media and Network: The DROID has the audio and video streaming media and supports multiple video formats like AAC, MP3, WAV, MIDI and so on. It supports Wi-Fi and Bluetooth, the GPS provides the service like aGPS, sGPS, Google maps, eCompass.

3.2 iPhone

The iPhone operating system is used on iPhon3, iTouch and iPad.

Processor: Samsung S5PC100 (ARM Cortex A8 @ 600 MHz. The processor in the 3GS at 600 MHz is 1.5x faster in terms of raw MHz, but it's based on the Cortex A8 which uses the more advanced ARM v7 architecture versus the 3G model at ARM v6[10].

Memory and secondary storage: The memory in iPhone is only 128MB, but it supports 8GB second storage for general storage.

Screen: The screen in iPhone is a 9 cm liquid crystal display with scratch-resistant glass and 18-bit color. The touch screen has multi-touch sensing and its touch and gesture features are based on technology developed by FingerWorks [9].

Sensor: The iPhone has three sensors, proximity sensor deactivates the display and touch screen, it save the battery power and prevent inadvertent inputs. The ambient light sensor adjusts the display brightness and hence also saves the battery power. The 3-axis accelerometer sensor senses the orientation of the phone and changes the screen accordingly. Plus, moisture sensor installed in iPhone is capable to detect whether the device has been affected or damaged by water.

Media and network: The audio processing plug-ins, known as audio units, can be dynamically accessed from the user's iPhone OS application. It allows user to add a variety of useful, prepackaged audio features and take advantage of the low latency that audio units offer. iPhone OS ships with audio units that support mixing, equalization, format conversion, and I/O for recording, playback, or live chat. The built-in Bluetooth 2.x + EDR support the wireless earpieces and headphones. Stereo audio was added in the 3.0 update for hardware that supports A2DP. With these profiles, iPhone can exchange multimedia files like pictures, music and video, with other Bluetooth enabled cell phones

Battery: iPhone has an internal rechargeable battery which is not user-replaceable. The battery can be charged when connect the iPhone to a computer via USB to dock connector cable. The official report from Apple's website says that the battery life is designed to retain up to 80% of its original capacity after 400 full charge and discharge cycles. The iPhone's battery is capable of providing up to 250 hours standby, 24 hours of music, seven hours of video, six hours of web browsing and eight hours of talk time.

Camera: iPhone has a built-in fixed focus 2.0 megapixel camera located on the back for digital photos. The camera has no optical zoom, flash or auto-focus, and does not supporting video recording neither. iPhone 3GS has a 3.2 megapixel camera which supports auto focus, auto white balance and video recording. The video can be uploaded to YouTube or other services directly.

4 System management

4.1 Thread management

Android

When the first of an application's components needs to be run, Android starts UI thread, a Linux process for it with a single thread of execution. The UI thread, which also know as main thread is automatically created when an Android application is started. UI thread is very important because it takes in charge of dispatching the event to the appropriate widgets. UI thread also the thread people interact with the Android widgets. When people touches a button, the UI thread will dispatches the touch event to the widget which in turn sets its pressed state and posts an invalidate request to the event queue. It dequeues the request and notifies the widget to redraw itself. By default, all components of the application run in that process and thread. Each component is run by a process, and the component elements have a process attribute which specify the process where the component should run. By setting bthe attributes in different way, it is easy to know whether the component run in its own process or share a process with other components. No component should perform long operation when called by the system in order not to block other components which also in the process. Besides, when the memory is low and other processes have more immediately requirement, the Android will decide to shut down a process. The application components running in that process will be destroyed consequently. Therefore, android weights the importance of each thread to the user to decide the victim.

Android provides Handler and Looper to manage threads and let them communicate with each other. Looper is used for running a message loop within a thread, headler is processing the messages and HandlerThread is setting up a thread with a message loop. After a child thread create an image from the web, it notifies the UI thread by sending a message using the handler that's bound to UI thread's message queue. The data produced by the child process can also send via the message. Therefore, the UI thread can update with the data produced by the child thread. When a Handler is created, it is bound to the message queue of the thread that created it.

iPhone

In iPhone OS, each application is made up of one or more threads, which represents a single path of execution. Every applications starts with a single thread, which runs the application's main function. The applications can have additional threads with executes a specific function. When application generates a new thread, it becomes an independent entity inside the process space. Each thread has its own execution stack and the kernel schedules its run time separately. As threads are in the same process space, they can communicate with other threads and processes. All threads in a single application share the same virtual memory space and have the same access rights as the process itself. Each thread requires the memory allocation in both the kernel memory space and the program memory space. The core structures needed to manage the thread and coordinate its scheduling. The stack space of thread and the per-thread data is stored in program's memory space. When first create the thread, the most of structures are created and initialized. In order to creating the low-level thread, a function or method to act as the main entry point for thread is needed, then one of the available thread routines has to be used to start the thread. The thread can be created by using the different methods as using NSThread, POSIX Threads, or using NSObjec to spawn a thread. After a thread is created, different thread environment needed to be configured, like configuring the stack size of a thread and the thread-local storage, setting the detached state of a thread and the thread priority.

4.2 Interrupt

An interrupt is an indication to a thread that it should stop what it is doing and do something else. It's up to the programmer to decide exactly how a thread responds to an interrupt, but it is very common for the thread to terminate.

a. Android

In Android, signal is a blocking I/O operation has been interrupted. The number of bytes that were transferred successfully before the interruption took place is stored in a field of the exception. Lock implementations provide more extensive locking operations than can be obtained using synchronized methods and statements. They allow more flexible structuring, may have quite different properties, and may support multiple associated Condition objects. As interruption generally implies cancellation, and checks for interruption are often infrequent, an implementation can favor responding to an interrupt over normal method return. This is true even if it can be shown that the interrupt occurred after another action may have unblocked the thread. An implementation should document this behavior.

b.iPhone

Most device drivers never need to take primary interrupts because their interrupt lines don't cascade into the system's interrupt controller. For example, FireWire and USB devices have a notion of interrupts, but are really just messages on a serial bus. These are commonly referred to as software interrupts because apart from the interrupt caused by the message itself, the interrupt is entirely simulated in software.

4.3 Inter-process communication

a. ***Android***

In Abdroid's computing, the Inter-process communication (IPC) is a set of techniques for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC techniques are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC). The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated.

b. ***iPhone***

For iPhone, any time the sequence in which two operations are completed affects the result, there is the potential for a race condition. For example, if two processes (in a single program or different programs) share the same global variable, then there is the potential for one process to interfere with the other or for an attacker to alter the variable after one process sets it but before the other reads it. The solution to race conditions of this type is to use some locking mechanism to prevent one process from changing a variable until another is finished with it. There are problems and hazards associated with such mechanisms, and they must be implemented carefully.

4.4 System calls

A system call is a request made by any program to the operating system for performing tasks—picked from a predefined set—which the said program does not have required permissions to execute in its own flow of execution. System calls provide the interface between a process and the operating system.

a. Android

Developer can trace the system calls invoked by Android for specific tasks by Android Task. The developer have to figure out which process is actually managing the data items, and then use strace <app name> after start to strace an application. It can also attach to an existing process by using the command 'strace -p <pid>'. Use 'ps' to find the process which is managing the contacts and messaging.

b.iPhone

The iPhone OS provides tools for system profiling. The Shark application lets user selectively find out what file system-related function calls the application makes. When user set up the session configuration, he can tell Shark exactly which function calls you want it to watch. A list of system calls includes all the file I/O calls plus fcntl, flock, fstat, fsync, link, lstat, lstatv, and stat. The sc_usage tool displays an ongoing sample of system statistics for a given process, including the number of system calls and page faults. The tool adds new system calls to the list as they are generated by the application being watched. The counts displayed are both the cumulative totals since sc_usage was launched and the delta changes for this sample period.

4.5 Memory management

Memory management is the programming discipline of managing the life cycles of objects and freeing them when they are no longer needed. Managing object memory is a matter of performance; if an application doesn't free unneeded objects, its memory footprint grows and performance suffers.

a. Android

Android applications usually are limited to 16 MB of heap. Though it is very little for some developers, it is a lot of memory for a phone. Because the more applications Android can keep in memory, the faster it will be when user switch between its applications. Therefore, the applications should use as little memory as possible to guarantee multiple applications run without getting killed. The memory leak phenomenon affects the memory usage, and hence affects the application switch efficiency. All Android apps are written in Java. Java, unlike other programming languages, does not require any coding to allocate or deallocate memory and handles all memory allocation/deallocation through a "feature" called automatic garbage collection. The garbage collector kicks in whenever an application no longer has any active threads pointing to it. By design, the garbage collector will not remove applications that are actively being used; but there may be a bit of system degradation while things are being cleaned up (usually a very fast process). Android rely on automatic memory management which handled by garbage collector. However, the garbage collector can sometimes cause performance issues if memory allocation is not handled carefully. The Android SDK provides allocation tracker, a tool to avoid the frequent garbage collection***.***

b.iPhone

iPhone has no garbage collection, developer has to clean up the variables after use them, otherwise the program will leak memory. Though NSObject class has accounting stuff help to keeps the track of how many other objects are currently using the object, but it is not automatic and developers have to adjust that by themselves. The rule for managing memory is to make sure the number of ownership methods called on an object will equal the number of loss-of-ownership methods by the time the program has finished executing. When create or copy an object, its retain count is 1. Thereafter other objects may express an ownership interest in your object, which increments it's retain count. The owners of an object may also relinquish their ownership interest in it, which decrements the retain count. When the retain count becomes zero, the object is deallocated (destroyed)***.***

4.5 Networking support

Wi-Fi: Wi-Fi is the IEEE 802.11 standard, is a two-way, short range protocol and operates in two bands. Initially, all Wi-Fi used 2.412-2.472 GHz. The 802.11n protocol added the ability to use 5.15 to 5.25 GHz (indoor use), but the iPhone is restricted to 802.11b/g. The networking stack in iPhone OS includes several interfaces over the radio hardware of iPhone and iPod touch devices. The main programming interface is the CFNetwork framework, which builds on top of BSD sockets and opaque types in the Core Foundation framework to communicate with network entities. Android comes with a complete support for the Wi-Fi connectivity. The main component is the system-provided WiFiManager. As usual, user obtain it via getSystemServices() call to the current context. Once they have the WiFiManager, they can ask it for the current Wi-Fi connection in form of WiFiInfo object. Also they can ask for all the currently available networks via getConfiguredNetworks(). That gives the list of WifiConfigurations.The WiFiScanReceiver is registered by WiFiDemo as a broadcast receiver to be invoked by the system when new Wi-Fi scan results are available. WiFiScanReceiver gets the callback via onReceive(). It gets the new scan result from the intent that activated it and compares it to the best known signal provider. It then outputs the new best network via a Toast.

Bluetooth: Bluetooth is a two-way, ultra short range protocol that works in the 2.40-2.485 GHz band. It avoids interference with other systems in the same band by being very low power, about a milliwatt, and has a maximum range of about 10 meters. The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

3G: iPhone 3G(S) is a "quad-band phone" which means it can also operate on the GSM 900 and GSM 1800 frequencies used in Europe. 3G technology gives iPhone fast access to the Internet and email over cellular networks around the world. With support for 7.2Mbps HSDPA, iPhone 3GS also makes it possible to do more in more places: Surf the web, download email, get directions, and watch video — even while user is on a call. Since iPhone seamlessly switches between EDGE, faster 3G, and even faster Wi-Fi, user can always get the fastest connection available. Users can even share their Internet connection with laptop via Internet tethering.

4G: 4G refers to the fourth generation of cellular wireless standards. It is a successor to 3G and 2G standards, with the aim to provide a wide range of data rates up to ultra-broadband (gigabit-speed) Internet access to mobile as well as stationary users. Sprint openly confirsm plans to sell a 4G Android smartphone next year. 4G aptly called MAGIC (Mobile multimedia, Anytime/any-where, Global mobility support, Integrated wireless and Customized personal service) provides for data rates of up to more than 100Mbps for the mobile users. This WiMax-connected device will run three or five times faster than the 3G-connected iPhone (on a good day), and will feature a bunch of apps that fully exploit that speed. Verizon Wireless said earlier this year that they see the Apple iPhone on their 4G network in 2010. A rumor about Verizon and Apple already testing a new iPhone on the Verizon Wireless 4G LTE network sneaked its way through the Motorola Droid coverage over the weekend***.***

4.6 Power Management

a.Android

Android supports its own Power Management (on top of the standard Linux Power Management) designed with the premise that the CPU shouldn't consume power if no applications or services require power. Android requires that applications and services request CPU resources with "wake locks" through the Android application framework and native Linux libraries. If there are no active wake locks, Android will shut down the CPU. The Android Framework exposes power management to services and applications through the PowerManager class. User space native libraries should never call into Android Power

Management directly. Bypassing the power management policy in the Android runtime will destabilize the system. All calls into Power Management should go through the Android runtime PowerManager APIs.

b.iPhone

iPhone do not have the power management toolkit as Mac OS does. Instead, this function is embedded into the core layer, which intelligently powers up planes of devices as the system goes into standby or to sleep. The most power hungry systems in iPhone from most to least are 3G radio system, Wi-Fi, 2G radio system, Bluetooth and GPS. Reduce the number of power consuming applications helps to save the energy. When put an iPhone into sleep, it will disconnect from network, turn off the Wi-Fi and screen light.

4.7 Development kit

a.Android

The Android Software development kit includes a comprehensive set of development tools which include debugger, libraries, documentation, sample codes and tutorials. The SDK support the development platforms with x86-architecture computers running Linux, Mac OS X, Windows XP and Vista. The SDK also requires to development environment have Java Development Kit, Apache Ant and Python. Eclipse with Android Development Tools Plug-in is the official integrated development environment (IDE) for Android SDK.

The Android 0.9 SDK beta was released on 18 August 2008. It provides an updated and extended API, improved development tools and an updated design for the home screen. The version 1.0 Android SDK was released on 23 September, it had fixed the main bugs and made several API changes from the 0.9 version. Google released the Android SDK version 1.1 on 9 March 2009. Version 1.1 includes the update for supporting search by voice, priced applications, alarm clock fixes, fixes mail notification and so on. It also made the update that the Dev phones can access paid applications and developers can see them on the Android Market. The SDK versions 1.5 and 1.6 were released on May 2009 and September 2009 respectively. They support many new features like video recording, voice recognition, battery usage indicator and VPN control applet.

b.iPhone

With this software the programmer can develop applications that run on iPhone and iPod touch, Including the Xcode IDE, Instruments, iPhone simulator, frameworks and samples, compilers, Shark analysis tool, and more. With over 1,000 new APIs, iPhone SDK provides developers with a range of new possibilities to enhance the functionality of their applications. New APIs also provide support for applications to communicate with hardware accessories attached to iPhone or iPod touch. The SDK requires an Intel Mac running Mac OS S Leopard or later, other operating systems like Microsoft Windows and older version of Mac OS X are not supported.

5 Comparison

5.1 Development Environments

The Android use java as develop language. With Android's support for multiple processes and component reuse, the platform itself provides support for Intents and Activities (an intent is just a variant of a command) and provides a way of declaring user preferences in XML which is extensible allowing custom UI components to be integrated. Android development leverages the excellent JDT tools, everything Java is indexed, the IDE has a rich model of the source code, and refactoring is seamless. JDT's incremental compiler provides immediate feedback with errors and warnings as you type. In Android, UI builder can't display UIs how they'll actually appear.

iPhone uses Objective-C as its development language. Eclipse can be used as iPhone OS development IDE. Beside, Aptana, a cross-platform IDE has released their iPhone Development Plug-in for their Ajax IDE. It can preview pages in both horizontal and vertical mode, and includes a wizard for importing the most common Ajax libraries. iPhone application developers are given a good UI builder; It's flexible and can model some sophisticated UIs, ease to port third party applications**.**

5.2 Reliability and Security

Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications. Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform, and per-URI permissions for granting ad-hoc access to specific pieces of data. As an open platform, Android allows users to load software from any developer onto a device. As with a home PC, the user must be aware of who is providing the software they are downloading and must decide whether they want to grant the application the capabilities it requests. This decision can be informed by the user's judgment of the software developer's trustworthiness, and where the software came from.

The iPhone has no security software and Apple doesn't let people load third-party programs on the device, which could reduce the risk of infection from malicious software. When the iPhone is connected to the

Web, dangerous possibilities emerge. The iPhone Auto-Lock disables the device's screen after a preset time period of non-use, and the Passcode Lock feature takes that a step further. Whenever the device's display locks, whether due to Auto-Lock or because you've hit the iPhone Sleep button--found on the top right of the device--Passcode Lock requires a four-digit code to be entered before the device can be employed again. With the password failure actions, it’s possible to configure an iPhone to perform a soft reset, lock the device, or wipe all the data after a pre-configured number of failed login attempts. If a user forgets his or her password, the device can be unlocked using a remote unlock service. The iPhone OS security APIs are located in the Core Services layer of the operating system and they are based on services in the Core OS (kernel) layer of the operating system. Applications on the iPhone call the security services APIs directly rather than going through the Cocoa Touch or Media layers. Networking applications can also access secure networking functions through the CFNetwork API, which is also located in the Core Services layer.

The iPhone OS security implementation includes a daemon called the Security Server that implements several security protocols, such as access to keychain items and root certificate trust management. The Security Server has no public API. Instead, applications use the Keychain Services API and the Certificate, Key, and Trust services API, which in turn communicate with the Security Server. Because, unlike the Mac OS X security services, the iPhone OS security services do not provide an authentication interface, there is no need for the Security Server to have a user interface. Although Mac OS X includes a low-level command-line interface to the OpenSSL open-source cryptography toolkit, this interface is not available on the iPhone OS. For iPhone OS development, use the CFNetwork API for secure networking and the Certificate, Key, and Trust Services API for cryptographic services**.**

5.3 Virtualization

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management. Unlike most virtual machines and true Jave VMs which are stack machine, the Dalvik VM is a register-based architecture. Like the CISC vs. RISC debate, the relative merits of stack machine vs. register-based approaches is a subject of continual argument. Generally, stack-based machines must use instructions to load data on the stack and manipulate that data, thus, require more instructions than register machines to implement the same high level code, but the instructions in a register machine must encode the source and destination registers and, therefore, tend to be larger. This difference is primarily of importance to VM interpreters for whom opcode dispatch tends to be expensive and other factors are relevant for JIT compliation.

A tool called dx is used to convert Java .clasee files into the .dex format. Multiple classes are included in a single .dex file. Duplicate strings and other constants used in multiple class files are included only once in the .dex output to conserve space. Java bytecode is also converted into an alternate instruction set used by the Dalvik VM. An uncompressed .dex file is typically a few percent smaller in size than a compressed .jar (Java Archive) derived from the same .class files.

The Dalvik executables may be modified again when they get installed onto a mobile device. In order to gain further optimization, byte order may be swapped in certain data, simple data structures and function libraries may be linked inline, and empty class objects may be short-circuited, for example.

Being optimized for low memory requirements, Dalvik has some specific characteristics that differentiate it from other standard VMs: The VM was slimmed down to use less space. Dalvik currently has no just-in-time compiler, but Android 2.0 includes experimental source for one (disabled by default). The constant pool has been modified to use only 32-bit indexes to simplify the interpreter. It uses its own bytecode, not Java bytecode. Moreover, Dalvik has been designed so that a device can run multiple instances of the VM efficiently.

For iPhone, Sun Microsystems plans to release a Java Virtual Machine (JVM) for iPhone OS, based on the Java Platform, Micro Edition version of Java. This would enable Java applications to run on iPhone and iPod Touch**.**

6 Conclusion

Table 1 compares features between iPhone 3.0 and Android.

Table 1 iPhone 3.0 vs. Android

	iPhone 3.0	Android
MMS Support	Yes	Yes
Copy/Cut and Paste	Yes	Yes
Application Notification	Push Notifications	Background Processes
Phone-wide Search	Yes	No
Email Search	Yes	Yes(Gmail search box)
Calendar Search	Yes	No
3G Tethering	Yes	Yes(3rd-party app)
Voice Over IP	Yes	No
File Sharing	Yes	Yes(3rd-party app)
Peer to Peer Bluetooth Connectivity	Yes	Yes

Compare with iPhone, Android provides developer more freedom. As Android is an open source, developer can create and deploy anything on the device. Compare with iPhone OS, Android is more widespread and applied in different model. Besides, Google provides many services like Google Search, gTalk, Google Doc, Google Map and so on. Once integrate these services with Android, it can provide customer more benefits.

iPhone OS, on the contrary, is designed exclusively for Apple's products, like iPhone, iPod and iPad. This helps to avoid reduce the functionality to suit the hardware needs for different devices. Also this focus can help developers make more useful and powerful applications based on customers requirement.

Reference

1. Greg Kroah-Hartman. "Android and the Linux kernel community" http://www.kroah.com/log/linux/android-kernel-problems.html.
2. "Linux developer explains Android kernel code removal". http://news.zdnet.com/2100-9595_22-389733.html.
3. "What is Android". Android Developers. http://developer.android.com/guide/basics/what-is-android.html.
4. Open Handset Alliance. "Industry Leaders Announce Open Platform for Mobile Devices". http://www.openhandsetalliance.com/press_110507.html.
5. Bort, Dave. "Android is now available as open source". Android Open Source Project. http://source.android.com/posts/opensource.
6. Honan, Matthew. "Apple unveils iPhone". Macworld. http://www.macworld.com/article/54769/2007/01/iphone.html.
7. Android architecture. http://www.slideshare.net/deepakshare/android-arch-presentation
8. An overview of iPhone OS architecture. http://thecoffeedesk.com/news/index.php/2009/05/17/iphone-architecture/
9. Fingerworks. http://www.fingerworks.com/
10. Into Mobile. http://www.intomobile.com/2009/06/18/iphone-3g-s-torn-apart-samsung-s5pc100-arm-cortex-a8-600-mhz-inside-and-double-the-ram.html