Origins

iPhone OS holds much of its origins in NEXTSTEP, an operating system created in the late 1980s by original Apple cofounder Steve Jobs. Developed with a core UNIX base and a mach (micro-) kernel, NEXTSTEP featured several developmental and user interface features that have carried forward into OS X today. NEXT acquired an object oriented, message-passing language built on top of C known as Objective-C for the core of its operating system.

After NEXTSTEP was acquired by Apple in the late 1990s, work began on the next Mac operating system with a large basis on NEXTSTEP. Although NEXTSTEP with Steve Jobs at the helm seemed like the perfect match for the company and the future operating system, NEXTSTEP was only acquired after BeOS declined acquisition by Apple. BeOS was created by several former Mac developers and was based on a UNIX-like environment developed in C++.

Apple based the development of OS X on a smaller sub-environment known as Darwin, which is based on both FreeBSD and the NEXTSTEP Mach kernel and associated libraries and contains a fair amount of its open-source code. The iPhone OS is able to trace its ancestry back to this operating system layer almost directly, since nearly all of the OS-level features (kernel, services, libraries, etc.) are shared between Mac OS X and the iPhone OS, and both are developed in Objective C.

Android can find its origin largely in the Linux kernel. Developed by Linus Torvalds in the early 1990s, the kernel is licensed under the GNU Public License (GPL) and is the basis of many operating systems distributed throughout the world. The use of the GPL requires that nearly all component features which statically or dynamically link with the kernel release source code under the license.

Unlike the microkernel used in OS X and iPhone OS which runs only necessary processes in supervisor mode, the Linux kernel is run entirely in supervisor mode. This requires that much of the code executed in normal operation under Linux is run in supervisor mode, allowing for an increased potential for abuse should a single feature of the kernel be exploited by a malicious user.

The Android OS is an initiative created by Google after acquiring Android, Inc., a software development company, in 2005. Programs written for Android are run under a Java Virtual Machine using Google-created libraries for both functionality and user interface.

Hardware

Apple has held tight control over the hardware which runs iPhone OS. There have been several revisions of two specific hardware lines: the iPhone, and the iPod Touch. Recently, an iPhone OS-based tablet computer was released which uses a modified version of the OS with its own distinct libraries and interface paradigms. Unlike Android devices, the iPhone OS does not allow for storage increase after sale, nor does it allow for its storage to be used as a general purpose USB Mass Storage Device. All iPhones support assisted GPS.

Device	iPhone OS Version (Maximum)	Storage Size	CPU	RAM	Camera
iPhone 2G	3.1.3	4-16 GB	412 MHz	128 MB	2 MP
iPhone 3G	3.1.3	8-16 GB	412 MHz	128 MB	2 MP
iPhone 3GS	3.1.3	16-32 GB	600 MHz	256 MB	3 MP
iPod Touch 1G	3.1.3	8-32 GB	412 MHz	128 MB	No
iPod Touch 2G	3.1.3	8-32 GB	532 MHz	128 MB	No
iPod Touch 3G	3.1.3	32-64 GB	600 MHz	256 MB	No
iPad	3.2 (iPad-only)	16-64 GB	1 GHz	256 MB	No

Unlike the control Apple exerts over its mobile devices, Google's Android devices feature a wide variety of both operating system modifications by the manufacturer and carrier and a wide variety of hardware paradigms, display sizes, and operating system versions. For the most part, Android handset manufacturers do not feel the need, even if possible, to keep their devices up-to-date. All Android devices support assisted GPS.

Device (US only, ordered by release date)	**Android OS Version** (Maximum)	**Storage Size** (Internal MB, all support μSD)	**CPU** (MHz)	**RAM** (MB)	**Camera**
HTC G1	1.6	256 MB	528	192	3.2 MP
Samsung Galaxy	1.5	8 GB	528	?	5 MP
HTC Hero/G2	1.6 (modified)	256 MB	528	288	5 MP
Motorola Cliq	1.5	512 MB	528	256	5 MP
Samsung 5700	2.1	200 MB	800	?	3.2 MP
Motorola Droid	2.1	512 MB	550	256	5 MP
HTC Google N1	2.1	512 MB	1000	512	5 MP
Motorola Backflip	1.5 (modified)	?	?	?	5 MP

Android also features a few in-production tablet devices but none have been widely released. As one can readily tell, the Android OS is fractured and inconsistent across devices. While a manufacturer could readily keep their users up to date on the most recent version, many choose not to and consider the devices complete.

Although all Android devices feature expandability with their μSD card slots, the hardware limit of μSD cards is 32 GB which are currently only available from SanDisk for $200 MSRP. Android applications are only able to be installed to internal storage, eliminating applications such as GPS software which requires multi-GB storage space for offline data. The HTC Google N1 is the only Android device with an 800x480 pixel resolution.

The amount of control carriers and hardware manufacturers have on the system is unlimited by Google. For example, the Motorola Backflip was modified by AT&T to remove application installation outside of the Android marketplace and has largely been poorly received on the network. HTC devices featuring the Sense UI have reports of a significant increase in user interface lag and a significant delay in operating system updates if provided, largely due to the extensive modifications created by HTC.

Thread Management

The use of a microkernel by iPhone OS based on the OS X counterpart produces a few downsides to non-user-land services such as interprocess communication (IPC). Two system calls are necessary to invoke the kernel service which forwards a message to the other process, which adds a significant amount of additional delay in IPC; the Linux kernel only requires a single system call to achieve the same result, and can access the data structures necessary in the same operation. With a set of optimizations, even a user-land driver in a microkernel is able to achieve nearly kernel level performance without potentially compromising the security of the kernel's execution despite a high level of interrupts and context switches. However, no third-party drivers are allowed to be executed on the iPhone OS, relegating any external hardware support to third-party applications which do not remain executing all the time.

For the most part, however, the operational flow of both operating systems follows a fairly standard pattern since both have UNIX-like origins. However, paradigms for execution are wildly different on both operating systems. In Android, a user interface thread separate from program execution is a common and often required technique to allow for power-sensitive multitasking. When relegated to background status, the user interface thread is often suspended under Android. However, in iPhone OS, only system-level processes are able to be suspended when not actively run and third-party processes are killed when the user changes to another application. The libraries featured in iPhone OS are almost entirely non-blocking, largely eliminating the need to separate UI from functionality.

Networking Support

Both iPhone devices and Android devices contain an abstraction layer between the operating system and actual wireless communication. In the iPhone OS, the radio and associated activity is managed by a separate firmware module known as the Baseband. In the Android OS, the firmware associated with the network activity is known as Radio Firmware.

Both iPhone OS and Android devices generally support Bluetooth 2.1, 802.11b/g, and varying support for communication protocols for 2G and 3G. Android devices span both GSM and CDMA networks, so either dual-network or specific radios must be used in the device for either network type. However, the iPhone devices only support GSM connectivity, and inside the United States feature a software lock which doesn't allow operation outside of the AT&T network. iPod touch devices do not feature a GSM or CDMA radio, but do include Bluetooth and Wi-Fi.

Support for 4G is forthcoming for Android devices; a modified version of the HTC Nexus One is planned to be released on Sprint for its 4G support being deployed this year. A 4G iPhone device is anticipated but has not been announced by Apple; given that AT&T's 4G network will not be available until 2011 it is unlikely that support for 4G would be usable in the short-term for iPhone users.

Power Management

Both Android and iPhone OS support reducing screen brightness based on ambient conditions. Additionally, a timer-based screen sleep is used by both to reduce power usage. By far, the display uses a significant portion of all power consumption on both devices. Although iPhone information is unreleased, Android features a “battery use” information screen which shows most users have roughly 50% of power consumption by the display device.

On the Android device, a “widget” (small application on the home screen) allows for quickly disabling certain power-consuming features such as 3G, Wi-Fi, Bluetooth, GPS, etc. On the iPhone, a user must navigate a large amount of menus to achieve the same level of control over power consumption.

Since Android allows for third-party applications to execute in the background, the power consumption of such applications must be considered for overall power management. However, Android does feature a significant level of control over how a process is executed while in the background. The user interface thread is often suspended, and only timed actions are performed by applications for syncing reasons, or for music playback through Android’s system-level media input service which executes audio for media applications.

On the iPhone, only Apple-created applications are capable of continuing execution in the background while a user executes other operations. Applications are notified of low-memory conditions on the iPhone and are able to destroy cached objects to lower memory usage during execution.

Software Development

Development for iPhone OS-based devices is only available through Xcode on Mac OS X. Using the iPhone SDK available for free, developers can create and compile iPhone OS applications on any Intel-based Mac and test execution using the iPhone or iPad simulator included.

Android development is available for all three major platforms: Windows, Mac and Linux. Since Android is Java-based, Eclipse is the preferred IDE for Android development and the SDK includes Eclipse-specific development utilities.

iPhone OS applications can only be written in three languages: C, C++ and Objective-C. The C-level libraries in the iPhone OS are known as Core Foundation and can be used in any other programming language available for the device. The Objective-C libraries are known as Foundation and feature a staggering amount of APIs. Apple uses a shared library system known as frameworks which allow applications to link against the frameworks at runtime. The most common framework on iPhone OS is UIKit, the iPhone equivalent of OS X's AppKit, and handles much of the user interface work.

Android features libraries in standard Java library access methods. Google's Java libraries also include a set of UI libraries. However, unlike iPhone OS, applications on Android can reference and communicate with each other applications and share executable code, such as displaying data or basic tasks and activities.

On both platforms, code must be signed before execution takes place. However, Android only requires the signing for the application to execute. On iPhone OS, distribution must be performed in the App Store which requires approval by Apple. On Android, the Market is an optional route for application release and users can install any application distributed in the .apk format.

Application Porting

Both Android and iPhone OS support openGL, which allows for video game development with limited effort. iPhone OS has always supported C execution of openGL applications, but until recently Android still requires a Java level of execution. Recently the Native Development Kit, or NDK, is a SDK which allows for compiled C or C++ execution on Android devices with 1.5 or later installed.

Neither platform can execute code compiled or designed for the opposite platform. Some intermediary programs exist, such as the Corona SDK, which allow for applications to be written in a common language and then compiled for a specific platform when distribution is necessary. Adobe has also announced support in CS5 for compiling Flash programs as native iPhone applications, but has not discussed Android which will have support for Flash 10.1 when it is released.

Memory Management

Although Objective-C gained garbage collection support in Objective-C 2.0, this did not extend to the iPhone. Memory management on the iPhone must be done at the object level using a release and retain paradigm in which a count of the number of retentions an object has is used to determine if it should continue to exist

For example, consider the following execution:

Object	Action	Retain Count
A	Allocate	1
B	Allocate	1
A	Retain	2
A	Release	1
B	Release	0 (freed)
A	Retain	2
A	Retain	2
A	Autorelease	1 (pending on run loop)
A	Release	0 (freed on next run loop)

Autoreleased objects free themselves automatically on the runloop following their retain count reaching 0. Objects which are not autoreleased and reach 0 are freed when their final retention is released. Often objects from the iPhone libraries return autoreleased Memory allocation using standard malloc/free can also be used, but for Objective-C programs it is uncommon to manage memory directly.

On Android, memory management is handled using the standard Java garbage collection, but is performed on a thread-level basis. For this reason it is general practice to avoid object allocation on the user interface thread, since the execution of garbage collection can cause a noticeable delay in execution to the user.

Virtualization

The major difference in virtualization between iPhone OS and Android involves virtual memory. For speed and efficiency reasons, Apple does not implement a virtual memory system in iPhone OS. This means that no swap space can be used, and applications are always limited to the size of the physical memory. Applications can be alerted to low memory situations to dump any potentially unnecessary data.

On Android, application swapping to SD card, an enormously slow operation, is performed in cases of low memory. Additionally, applications can reduce their memory footprint in low-memory situations with an alert from the operating system.

The ramifications for significant multitasking in iPhone OS by the lack of virtual memory are significant. Any multitasking implementation would have to involve killing the processes and not writing them to disk when memory usage becomes critical. In Android, multitasking is as normal as it is on a full operating system.

Reliability and Security

Reliability is a difficult metric to measure when it comes to the hardware of mobile devices. Although all of Apple's iPhone, iPod touch and iPad devices are manufactured under Apple's direct control, Android devices are created by an array of manufacturers with varying quality and control systems.

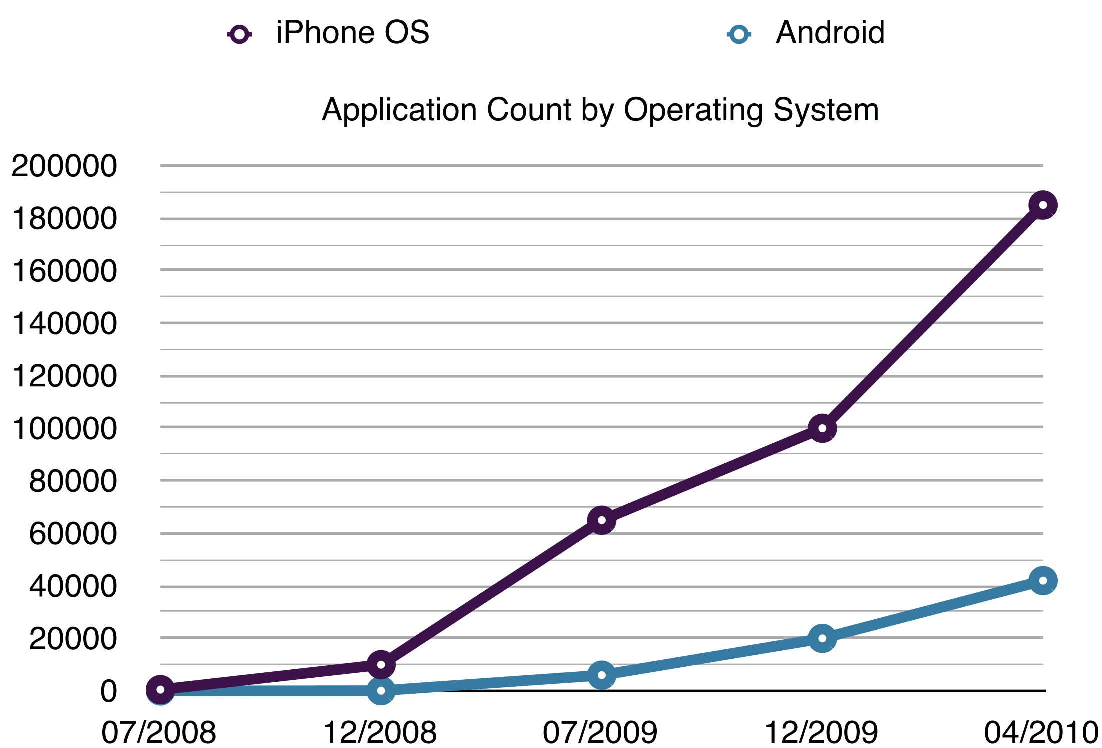
Focusing on the software, the iPhone has a leg up in that it shares much of the operating system in common with OS X, which has had several years of intense development to create a relatively reliable and consistent operating system experience. iPhone OS's microkernel allows for a more modular, smaller-weight kernel which allows for iPhone-specific kernel changes to be minor or modularized.

Android's monolithic Linux kernel, however, requires more extensive changes to support the smaller device. Since it's not based on a modular design, the Linux kernel's interactions with other parts of its codebase is direct. Although it may be a faster mechanic, from a reliability perspective it is less efficient and more prone to problems.

Android's open Market allows for any developer to add an application without an approval process. This also allows for malicious applications to enter the marketplace, or for users to install a malicious application outside the Market with the open installation policy. From this perspective, Apple's closed App Store and limited third-party-code execution helps to lock down the device from potential malicious uses.

The open source nature of Android allows its code to receive a more extensive security audit than may be possible in the closed source environment of the iPhone OS. A significant security vulnerability in iPhone OS would require Apple to update and release a fix, while a recompiled or corrected version of Android can be released potentially without Google interaction at all.

Strengths and Weaknesses



The volume of third-party applications is a major driving force of any platform, and iPhone has had a huge amount of development since the release of third-party support in July of 2008. The Android Market didn't exist until midway through 2009, which drastically reduced the initial adoption rate for Android applications.

Android application development has begun to increase at a faster rate following the release of Android devices on Verizon and Sprint, outside of the smaller T-mobile bubble.

One major advantage of Android is its focus on cloud-based computing. All Android applications which require persistent information sync their data to a server in the internet. For example, contacts on the device are all stored in Gmail's contacts; an Android device could be formatted and restored with only settings and applications and still retain all non-SD-card data.

Android also has a significantly more integrated notification system than iPhone OS. iPhone OS uses a significant amount of modal dialogues in situations where Android places a non-modal entry in the system wide notification list. iPhone operational flow is largely dependent upon immediately responding to actions, while Android allows users to delay actions.

The iPhone OS's major advantages focus mostly upon media playback, the medium that Apple's original portable devices generally aligned with. Its iPod and iTunes integration allow users to keep all of their digital media in one location and generally from one unified source.

Android's development environment is also more freely available than the iPhone OS. Android devices are able to be unlocked to run unofficial firmware through a simple command in the Android development environment, although it does void the warranty if not done on a development phone. iPhones must be exploited, or jail-broken, to allow for any application to be executed or for the firmware to be altered, while Android can execute any application from any source.

In the end, the differences between Android and iPhone OS begin to blur as the competition forces adoption of similar features in the opposite system. iPhone OS 4 is gaining multitasking and a larger array of integration features Android has had from the beginning, and Android adopted a native C SDK to allow for easier and significantly faster game development. The competition between Google and Apple is starting to become severe and only time will tell what direction it takes both companies.