ABSTRACT
In June of 2007 Apple Inc. released the iPhone mobile device which would prove to be immensely popular both as a phone and as a tool for simplifying many tasks that had previously been difficult for a device of its type. In 2005 Google Inc. acquired Android Inc. and in October of 2008 the first device running the operating system, the HTC Dream, was released. Both the iPhone and Android operating systems have gained sizeable market shares since their introductions and this paper aims to detail the workings of both as well as explore the similarities and differences between the two.

1. INTRODUCTION
In the recent years both the iPhone and Android devices have become increasingly popular. Apple currently holds just over a quarter of the American smartphone market, while the relatively new Android devices already hold almost as much as Palm. (Ritchie, 2010) As these numbers grow it becomes increasingly important to understand the devices and the software that makes them work, to understand the strengths and vulnerabilities of each. This paper will begin in section 2 with a brief history of the continued development of each OS since their respective releases. Section 3 will describe the hardware of the newest iPhone, the iPhone 3GS, and one of the newest phones running the Android OS, the Nexus One. Section 4 examines details of the inner workings of each OS. Section 5 compares and contrasts the two operating systems on topics such as ease of development and security leading to some concluding statements in Section 6.

2. Histories of Development
In the years since their respective releases, both operating systems have seen continuous and fruitful development. Many features have been added and they have become more robust. This section will outline the key moments in the development of both the iPhone and Android operating systems.

2.1 Development of the iPhone OS
While the nature of the relationship between the iPhone OS and the iPhone device makes it difficult to describe the development of one without also describing that of the other, this section will focus on the software advances made in the years since iPhone 1.0 was released in June of 2007.

Currently in version 3.1.3 (with version 3.2 available for the iPad) and with 4.0 announced, the iPhone OS has grown to support many features desired by users but unavailable at the original release. Much of the 1.x release cycle, as should be expected when releasing a new OS and device to the market, focused on fixing bugs and making small aesthetic changes such. The aesthetic changes included items such as changing the icons of official Apple Inc. applications, allowing a double-tap of the spacebar to insert a period, enabling landscape mode for the several applications, and relabeling buttons to clarify their uses. These changes, while small, combined to add Apple's well-known shine and polish to the user's experience when using the device.

The first major update to the device came in July of 2008, more than a year after the initial release of the device. The release of iPhone OS version 2.0 corresponded with the release of the updated iPhone 3G device. In addition to the obvious support for the new network capabilities of the iPhone 3G, the 2.0 version added many other important features. These features included support for WPA2¹, IPSec VPN,² Scalable Vector Graphics, and many new supported languages. There were also many aesthetic changes similar to those made throughout the 1.x release cycle such as the ability to take screenshots, saving pictures from the Safari web browser application, parental controls, and the ability to view many Microsoft Office and iWork documents on the device. Other notable releases in the 2.x cycle include 2.0.1, which focused on increasing the performance and reliability of many core applications on the device; 2.1, which delivered improved battery life, numerous bug fixes in core applications, and the usual aesthetic and usability changes.

The most recent major update, iPhone OS version 3.0, was released with the updated iPhone 3GS in June , 2009. The iPhone 3GS added new hardware functionality to the device which will be discussed later in section 3.1. On all devices, however, version 3.0 brought an impressive list of features and additions including the much requested ability to cut, copy, and paste. Other additions included turn-by-turn navigation for the Maps application, major changes to the Photos application (to accommodate the ability to copy and paste photos), the ability to send multimedia messages (MMS), better standards coherence in Safari as well as some support for features of HTML5, and 3G tethering. Several more releases have been made up to version 3.1.2 this February; these releases have mostly been small maintenance updates leading up to the release of 4.0 which has been announced for a June 2010 release.

2.2 Development of the Android OS
The Android OS is developed for use on multiple devices, unlike Apple's iPhone OS, so there is not as strong a correlation between version releases and new hardware. As a result, the major releases have focused less on aesthetic changes and more on increasing the capabilities of the core OS and often correspond with an updated Linux kernel. The Android release cycle also seems to focus more on swift small updates rather than Apple's approach of yearly major updates with minor maintenance releases in between.

¹ Wi-Fi Protected Access
² Internet Protocol Security Virtual Private Network
Version 1.5 of the Android software stack was released in April of 2009. With it came support for accelerometer based application rotation, improved performance for many core applications, an onscreen keyboard, video recording and playback, many additions to the API, and perhaps most importantly an upgrade to the 2.6.27 Linux kernel. (Android 1.5 Platform Highlights, 2009)

A few months later in September, 2009, version 1.6 was released. This version added a quick search box similar to the iPhone's Spotlight, support for several types of VPNs, a new text-to-speech engine, gesture support, multiple API additions, and an update of the Linux kernel to version 2.6.29. (Android 1.6 Platform Highlights, 2009)

Version 2.0 was released in October. Notably, it did not include an update to the kernel as a new version had not yet been released. This release was also quite small with the major additions focusing on managing contacts, improved camera performance, HTML5 support in the browser, and improvements to the virtual keyboard. (Android 2.0 Platform Highlights, 2009)

While the feature list of each successive update does not seem as daunting as those of each major iPhone OS release, it should be noted that all the changes listed were made within roughly a year while those listed for the iPhone span nearly 3. In the relatively short time that it has been available, the Android OS has grown into a competent and accessible operating system and will continue to benefit from its relatively responsive release cycle.

3. Hardware Overview

While choosing a device to study using the iPhone OS is fairly simple, the task is not so simple for the Android OS. Google claims 20 mobile phones are available running the OS. In order to make as fair a comparison as possible, I have chosen to use Nexus One and the iPhone 3GS. The Nexus One is one of the most recent and most open devices to utilize the Android OS and the iPhone 3GS is the most recent iteration of the device.

3.1 The Hardware of the iPhone 3GS

The iPhone 3GS is an integration of the iPhone focused on one thing: speed. Previous versions proved to be popular and usable, but with time users demanded more from the device, often more than the device could comfortably satisfy. With this in mind, Apple upgraded many of the core components as well as added new ones, in order to allow new features.

3.1.1 Processor

The iPhone 3GS contains an ARM A8 Samsung S5PC100. While the processor is rated to run at 833 MHz, in the iPhone 3GS it has been underclocked to 600 MHz. (Shimpi, The iPhone 3GS Hardware Exposed and Analyzed, 2009) Underclocking is often in cases requiring cool, quiet, low-power operation, all of which are imperative in a mobile device. The ARM A8 is a dual-issue (but in-order executing) 13-stage pipelined processor. It also provides a SIMD\(^1\), bringing great performance on vector operations (such as video decompression). The 64KB L1 cache is divided equally between instructions and data. There is also a 256KB L2 cache. (Shimpi, The iPhone 3GS Hardware Exposed and Analyzed, 2009)

---

\(^1\) Single Instruction Multiple Data
Figure 3. Dataflow of the PowerVR SGX GPU

3.1.3 Memory
The iPhone 3GS is available with a choice of 16 or 32 GB of flash storage. It also has 256 MB of DRAM main memory. This is twice as much as the previous iteration.

3.1.4 Display
Every iteration of the iPhone has shared the same display specifications:
- 3.5” diagonally, 3:2 aspect ratio
- 262,144 color 480x320px LCD with 163 ppi [iPhone Technical Specifications]

3.1.5 Camera
The iPhone 3GS features an upgraded 3 megapixel camera capable of autofocus, recording 30 FPS video, white balance and geotagging. [iPhone Technical Specifications]

3.1.6 Connectivity
The following is a list of connectivity features available in the iPhone 3GS:
- Wi-Fi (802.11 b/g)
- USB 2.0
- Quad band GSM/GPRS/EDGE, Triband UMTS/HSDPA
- Bluetooth 2.0, 2.1
- A-GPS
- Magnetometer, accelerometer, proximity and ambient light sensors
[Dion, 2007], [iPhone Technical Specifications]

3.1.7 Miscellaneous
The iPhone 3GS’s battery is a rechargeable lithium-ion battery listed as 1219 mAh and claimed to deliver 12 hours of talk time on a 2G network, 5 on 3G and up to 30 hours of audio playback.

Its audio system has a frequency response between 20 and 20,000 Hz (the general range of human hearing). Playback is supported for AAC, protected AAC, HE-AAC, MP3 (both constant and variable bitrates), Apple Lossless, AIFF, and WAV.
[iPhone Technical Specifications]

3.2 The Hardware of the Nexus One
The Nexus One is built on the QSD 8250 Snapdragon architecture which is a family of chipsets constructed around an ARM CPU by Qualcomm. The chipsets are made specifically for mobile devices.

As it is regarded as a set, it was difficult to find many details on the individual components.

The set includes a 1 GHz ARM-based CPU, a vast improvement in clock-rate over the iPhone 3GS. However, this particular chip does not include a dedicated GPU. This means that the gains in CPU performance must be leveraged against the need to utilize it in rendering complex graphics for games and videos.

The Nexus One comes with 512 MB of flash memory, but can also use microSD cards up to 32 GB. It has 512 MB of DRAM.

The display is 3.7” diagonally with an aspect ratio of 3:5, 800x480 px resolution and 252.15 ppi. It has a 100,000:1 contrast ratio and a 1ms response rate.

It boasts a 5 megapixel camera with autofocus capability, 2X digital zoom, and LED flash. It can capture video at 720x480 at 20 FPS or higher.

It can connect to 3 UMTS, HDSMA, HSUPA, and GSM/EDGE cellular networks as well as Wi-Fi and devices using Bluetooth 2.1. It has a digital compass, A-GPS receiver, cell tower and Wi-Fi positioning and an accelerometer.

(Feature Overview & Technical Specifications)

4. OS Features

4.1 Features and Capabilities of the iPhone OS

While the iPhone OS itself is proprietary and not available for viewing, it is known to at the very least be strongly based on the Mac OS X kernel, XNU, which is open source as part of the Darwin OS. This exploration will proceed under the assumption that for the purposes of memory and process management the two are largely similar.

4.1.1 Process Management, Inter-Process Communication, and Memory Management
The Mach microkernel is the portion of the OS responsible for handling the processor and memory resources presented to the OS. It handles process and thread scheduling as well as memory protection and management. Mach uses tasks to describe a collection of system resources which can be referenced and shared with other tasks through ports. A task can contain multiple threads. A Mach thread owns no resources, but can access those belonging to the parent thread. The Mach scheduler assigns runnable threads to processors. Tasks can be scheduled based on roles and priority. (Kernel Programming Guide)

[4] Pixels Per Inch
IPC is accomplished by the use of ports. A port is a unidirectional communication channel between a client requesting a service and a server providing it. The kernel allocates ports, and every entity in Mach but virtual memory ranges is given a port name at creation. Communication is limited by port rights. A port right can be given to receive, send, or send-once. Messages sent between ports are not entities. Mach also provides systems for message queues, semaphores, notifications, locks, and remote procedure call objects to accommodate IPC.

The Mach address space defines the set of valid virtual addresses. Mach is responsible for translating a given virtual address to a corresponding physical location.

(Kernel Programming Guide)

### 4.1.2 Power Management

In the Darwin OS, power is managed within I/O Kit. I/O Kit is an object-oriented framework for developing device drivers. I/O Kit aims to minimize the power consumed by a system, a goal that is increasingly important on mobile devices. Three power events are recognized: sleep, wake, and shutdown/restart. Every driver must recognize these events. Sleep can be further specified by the cause (idle, system, etc.). Power dependencies are monitored by a tree structure called the power plane. A device must at least have an on and off state, intermediate states of limited functionality and power consumption are also allowed. I/O Kit aims to use these attributes of these states to switch between them for optimal power usage without negatively impacting performance.

(I/O Kit Fundamentals: Power Management)

---

4.2 Features and Capabilities of the Android OS on the Nexus One

The Android OS is built on a highly customized Linux kernel, however it is not a distribution of Linux. It has no native windowing system, does not support glibc and does not include the full set of standard utilities. The kernel was chosen as a basis for the project because of its already tested and proven memory and process management as well as its security and driver models.

(Bradly, 2008)

#### 4.2.1 Process and Interrupt Management

Process and interrupt management is handled by the kernel. Every process is allocated a `struct task_struct`, a doubly linked list of which represents all tasks active in the system. Each task can be in 1 of 6 states: TASK_RUNNING, TASK_INTERRUPTIBLE, TASK_UNINTERUPTIBLE, TASK_ZOMBIE, TASK_STOPPED, or TASK_EXCLUSIVE. These states are used by the process scheduler. The process scheduler arbitrates access to the CPU between multiple processes.

When a userspace application makes a system call, the application executes an instruction, passing the arguments in registers and jumps to the system call entry point. This causes a trap into kernel mode. Once activated the system saves the registers, executes the requested system call from the system call table, returns the result and returns the processor to the point in the program where the call was made. (Linux Kernel Internals: Process and Interrupt Management)

#### 4.2.2 Power Management and Inter-Process Communication

Development of Android has added many enhancements to kernel, especially in the area of power management and the Binder Driver to facilitate more secure inter-process communication.
Android's power management policies are more aggressive than those standard to the Linux kernel. This is done with its use on mobile devices in mind: battery life is a high priority. The system implemented uses "wake locks." Each application and service must use a wake lock when requesting CPU resources. Open wake locks in the system prevent the OS from cycling into low power states. Requests received in the middle of switching to a low power state cancel the transition. The process of acquiring a wake lock is demonstrated below.

```java
PowerManager pm = (PowerManager)mContext.getSystemService(Context.POWER_SERVICE);
PowerManager.WakeLock wl = pm.newWakeLock(
        PowerManager.SCREEN_DIM_WAKE_LOCK |
        PowerManager.ON_AFTER_RELEASE, TAG);
wl.acquire();
// ...
wl.release();
```

Instead of glibc, Android uses Bionic libc. Bionic libc is released under the BSD license, is very small and fast. It supports Android specific features for properties and logging, and is not compatible with glibc. All Android applications must be compiled against bionic.

### 4.2.3 Virtualization and Sand-Boxing

Applications on Android devices run in a virtual machine called Dalvik. Dalvik was designed specifically with the limitations of working on mobile devices in mind. These include constraints on CPU speed, available RAM, swap space, and power consumption. The VM runs dex files which are specially compiled jar files. dex files save on memory by minimal repetition, per-type pools and implicit labeling. These savings can be as high as 50%.

Dalvik is a register based machine in order remove unnecessary memory access and increase the semantic density of instructions.

### 5. Comparison of the iPhone and Android Operating Systems

#### 5.1 The Android and iPhone Development Environments

Both Apple and Google provide developers with convenient packages to begin creating their applications. Apple's SDK is made to work with its IDE, Xcode. Xcode provides tools to manage projects, build projects for running on the iPhone, performance monitoring, and trouble shooting. Xcode contains an advanced text editor with the expected features such as code completion and syntax highlighting, inline errors and notes. An iPhone simulator is included in the SDK for running and testing applications once they have been built with Xcode. Also included is an interface builder which contains prebuilt interface elements allowing applications to be constructed which conform to the expected look and feel of an iPhone application.

![Figure 8. The Xcode project window. (Tools for iPhone OS Development)](image)

Android developers have a bit more choice in their development platform and IDE choice. They are not tied down to a single IDE, however, there is a preference for Eclipse as there is an official plug-in for the program to tailor the development environment for Android. Eclipse is already a popular and feature rich IDE. Furthermore, it is available on the three major platforms: Windows, Mac, and Linux. The provided plug-in provides tools to create and manage Android projects, create interfaces, add and configure interfaces, and build and debug applications.
6. Conclusion

As has been shown, both devices have their merits and faults. The iPhone provides a consistent platform for development, but requires developers to work on Apple's computers and subject their applications to its App Store approval process. Android follows a more open approach to development, providing SDKs for the three major platforms and encouraging the use of the equally available Eclipse IDE. However, this openness in choice of applications available for a user to run can backfire if malicious applications are proliferated.

Both platforms have had vulnerabilities exposed, but both have also been quick to repair and patch them. Neither is perfect, and I believe that the mobile device market is surely large enough for both to thrive.

7. Bibliography


