Matthew Lonis

Jeff Whitmer

CSCI-A 290

January 12th, 2017

***MINI 1***

**Topic 1: What are the basic features of the Android Operating System? What is it based upon? How is it different and/or alike other Operating Systems? This needs to be a very, very basic discussion, so you are looking for information about the “key features,” not the deep, deep, technical details.**

**3 URLs**

1. <https://www.gazelle.com/thehorn/2014/02/10/the-android-operating-system-10-unique-features/>
2. <http://www.howtogeek.com/189036/android-is-based-on-linux-but-what-does-that-mean/>
3. <http://www.infoworld.com/article/2907884/why-is-android-so-different-from-linux-distros.html>

**Precis of Sources**

URL 1 describes 10 unique features for the Android operating system. The first is NFC which allows android operating systems to communicate or interact across short distances. The next, is alternate keyboards which is simply the ability to exchange the stock android keyboard with a keyboard of your choice. The article offers several other ones like, infrared transmission to allow your phone to act like a remote, no-touch control using gestures instead of touches to control your phone, and storage and battery swap which allows the battery and internal memory storage to be changed out without needing any operating system updates or drivers.

URL 2 & 3 talk about what it means to say “Android is based on Linux” and how Android is different than Linux. Android is based on the Linux kernel, which is why it is often said it is based on Linux, but it can’t truly be qualified as a true Linux distribution of the likes of Ubuntu or any other major Linux distribution. Android, unlike a true Linux distribution, is missing all or some of the “GNU libraries” nor does it include an “X server like Xorg” which is the key to running standard graphical Linux applications. In order to run Android applications, Android uses the Dalvik virtual machine or the new Android RunTime to run applications written in Java. The same reasons Linux apps can’t run on Android is inversely the same reason as to why you can’t run Android apps on native Linux. Desktop Linux doesn’t include the Android APIs or Android runtimes necessary to running a standard .apk file package. Since Linux is open source, Google has been able to tailor their own custom version of the Linux kernel to fit the needs of Android and thus make Android “based on Linux” but yet, different.

**Summary of Topic**

Android is an implementation of the Linux kernel. Google has modified the open-source kernel for their own uses in order to optimize performance on a mobile device. Android functions a lot like many other mobile operating systems. However, even though Android is based on Linux, this doesn’t mean Android can run Linux desktop software native on the operating system. Android is missing many core libraries that standard linux OS uses and this prevents it from being like a true Linux operating system.

**JIT 1**

Is Android similar to other Linux distributions tailored to mobile devices? For example, Ubuntu, a common Linux distro, has a mobile operating system.

**Topic 2: What is the relationship between Java and XML in the Android Development environment? Why are both present/used/important? Are there important “gotchas” everyone should be aware of?**

**3 URLs**

1. <http://stackoverflow.com/questions/36957636/what-is-the-relationship-between-xml-and-java-in-android>
2. <http://softwareengineering.stackexchange.com/questions/307768/why-use-xml-in-android>
3. <http://stackoverflow.com/questions/5645468/why-is-xml-used-for-the-creation-of-ui-layouts-in-android>

**Precis of Sources**

URL 1 contains a post by a user that breaks down what XML and Java are and their relationships to one another. The post describes XML as being used for the layout of an app. This means how the UI looks, text placement, buttons, list views, etc. This can be thought of similar to HTML where these UI elements can also be described and defined via attributes such as text color, visibility, and font. The user also explains how XML can be used for animations and to make shapes to be used as a background or app icon.

The second URL is a forum discussion about why Android uses XML since XML files in Android are converted into Java. The explanation for this is because XML is simpler and easier to understand, be modified, and implemented within an app. The post even explains how other services such as WSDL and Microsoft (WinForms) implement a technique like this. It allows programmers to focus on a specific element at a time rather than juggling design/layout with logic. In principle, everything could be written with Java, but XML makes app layout easier with the only major con being that a developer must learn another language on top of Java.

The third URL discusses why Android uses XML. XML is a file type that is meant to be in a human readable format and isn’t truly an efficient file for a mobile device. The thing about android is, it uses XML and converts them into a compressed binary format which allows it to be highly efficient. The reason the developers of Android chose XML is due to its native support in many familiar tools and IDEs.

**Summary of Topic**

In Android, XML and Java both are necessary for an app. XML handles the layout of the app and its views allowing developers to focus on solely the design when working with XML. Java on the other hand handles the logic of the app and, since it is separate from XML, allows the developers to only focus on that. This technique is implemented elsewhere in the programming world and is very efficient and developer friendly/

**JIT 1**

How much XML do we need to know to develop our apps for this class?

**Topic 3: What makes for good Android application design? We will be talking about design for several meetings, but I want you to go out and see if you can find any sort of consensus whatsoever on what makes a good Android application. Are there acknowledged “best practices?” If you need to qualify your answer based on “what kind of application,” you can do so, but I would then like you to discuss several different “kinds” of applications and the best designs practices suggested for each.**

**3 URLs**

1. <https://developer.android.com/design/get-started/principles.html>
2. <http://thenextweb.com/apps/2013/11/04/30-beautifully-designed-android-apps/33/>
3. <http://www.computerworld.com/article/2909897/material-design-apps-android.html>

**Precis of Sources**

The Android Developer page list several broad ideas for good Android design principles. The principles are Enchant me (personalization/predictive), Simplify My Life, and Make Me Amazing (good performance). My last two URLs include a list of 30 apps that the writers deemed to be of good design. Most of these designs are shown in pictures but the consensus across most apps are these keys things:

1. Clear navigation whether it be via a menu style bar or a slide out menu. This means that when you change between pages, there is still a clear navigation as to where you are either with a page title in the menu bar or with a back button to take you back to the previous page
2. Using pictures instead of words wherever appropriate. The most common way I’ve seen this implemented is in news apps where there is a large picture that takes up the entire clickable square with a title underneath. This way the user is clicking on a large area instead of a line of text and the pictures and some “life” to the app.
3. Commonly used actions were in easy reach. Whether this means using the three-dot menu in the top right on all your pages, or having a refresh button in the top right for a news app, or having a search mutton in the top right for any app, these common and necessary features were always within reach and always easy to inherently understand allowing for a very easy user experience to pick up on.

**Summary of Topic**

In general, when thinking of good design principles for Android, I often think back to what I learned when learning how to write essays well. In essays, I was always told to break up paragraphs into smaller chunks because when readers see a paragraph spanning more than half the page, the reader often gets intimidated by the length and may not even want to read. I think about this with regards to Android where instead of making apps with lots of words instead use pictures when possible. Instead of having an app with only a few pages, break up things into multiple views to enhance the simplicity of a page (the key with this is to have a clear and effective navigation within the app). Instead of having an app that has no color, enhance it with a pleasing color scheme.

**JIT 1**

Will we get to implement a slide-out menu in this class?