# 안드로이드 소개

## 안드로이드란?

안드로이드(Android)는 휴대전화를 비롯한 휴대용 장치를 위한 운영체제이자 소프트웨어 스택(software stack) 이다.

안드로이드에는 다음 항목들이 포함된다.

- 운영체제

- 그래픽 사용자 인터페이스

- 표준 앱

## 안드로이드 구현 언어

안드로이드 운영체제는 리눅스 커널을 기반으로 C 언어로 구현되었다.

그래픽 라이브러리를 비롯한 다양한 안드로이드 구성 요소는 C/C++로 구현되었다.

안드로이드 앱은 Java로 구현된다. 안드로이드에서 Java는 표준 Java 기계와는 다른, 달빅(Dalvik) 기계에서 실행된다.

## 컴파일과 실행

### C/C++

C/C++ 컴파일러는 C/C++ 소스 코드를 기계어 코드로 변환한다.

C/C++ 소스 파일을 컴파일하여 생성된 \*.exe 실행 파일에는 기계어 코드가 들어있다.

\*.exe 파일은 직접 CPU에서 실행된다.

### 표준 Java

표준 Java 컴파일러는 Java 소스 코드를 바이트 코드(bytecode)로 컴파일한다.

\*.java 소스 파일을 컴파일하여 생성된 \*.class 파일에는 바이트 코드가 들어있다.

바이트 코드는 CPU에서 직접 실행될 수 없다.

Java 가상 기계(Virtual Machine)가 \*.class 파일을 읽어서 기계어 코드로 변환해서 실행한다.

표준 Java 컴파일러 실행 파일은 javac.exe 이다.

표준 Java 가상 기계 실행 파일은 java.exe 이다.

### 안드로이드 Java

먼저 표준 Java 컴파일러를 사용하여 Java 소스 코드를 표준 바이트 코드(bytecode)로 컴파일한다.

그 다음 달빅(dalvik) 컴파일러를 사용하여 표준 바이트 코드를 달빅 바이트 코드로 다시 컴파일한다.

달빅 바이트 코드는 달빅 기계에서 실행된다.

Java로 개발된 앱이 메모리가 작은 휴대용 기기에서 잘 실행될 수 있도록,

휴대용 기기에 특화된 달빅 컴파일러와 달빅 가상 기계를 구글이 개발하였다.

## bytecode로 컴파일하는 이유

기계어 코드로 컴파일 하지 않고 bytecode로 컴파일하는 방식의 장단점은 다음과 같다.

단점: 느리다.

CPU가 기계어 코드를 직접 읽어서 실행하는 방식이 가장 빠르다.

장점: CPU에 독립적이고, 운영체제에 독립적이다.

실행파일에 들어있는 기계어 코드는 CPU마다 다르다.

실행파일 구조는 운영체제마다 다르다.

그래서 Windows 실행파일을 맥에서 실행할 수 없다.

Java 소스파일을 컴파일하여 생성된 \*.class 파일의 구조는 운영체제에 무관하다.

그리고 bytecode도 CPU에 무관하다.

따라서 Java로 개발한 앱은, 특정 운영체제나 CPU에 무관하게 배포될 수 있고,

Java virtual machine만 있으면 실행될 수 있다.

## JIT & AOT 컴파일

JIT 컴파일: Just In Time 컴파일

AOT 컴파일: Ahead of Time 컴파일

Java는 JIT 컴파일 방식을 사용한다.

JIT 컴파일이란, bytecode를 실행하기 직전에 기계어 코드로 컴파일 하는 방식을 말한다.

Java virtual machine은 실행하기 직전에, bytecode를 기계어 코드로 변환(컴파일)해서 실행한다.

bytecode를 해석해서 실행하는 것보다(인터프리터 방식), 기계어 코드를 실행하는 것이 훨씬 더 빠르기 때문이다. 변환된 기계어 코드를 따로 저장하지는 않기 때문에, 실행할 때 마다 매번 다시 JIT 컴파일 해야 한다.

AOT 컴파일 방식은, 바이트 코드를 처음 실행할 때, 기계어 코드로 변환(컴파일)하고, 변환된 기계어 코드를 캐시(임시 저장소)에 저장하고, 다음에 실행할 때는 캐시에서 찾아서 실행하는 방식을 말한다.

Android는 AOT 컴파일 방식이다.

## Android SDK와 Android Studio IDE

Android SDK는 컴파일러 + 표준 라이브러리.

Android Studio는 통합 개발 도구 GUI

(Java SDK = JDK, Java IDE = eclipse)

# 안드로이드 역사

2003년 창업

2005.07 : 안드로이드 사를 구글에서 인수

2007.11.05

하드웨어, 소프트웨어, 통신 회사가 모여 만든 OHA(Open Handset Alliance) 결성

OHA에서 모바일 기기용 공개 표준 개발

2007.11.12 : 휴대전화용 안드로이드 플랫폼 발표 (무료로 공개)

안드로이드 SDK, 동작 샘플, 안드로이드 시스템 구조, 안드로이드 내장 SW 개발 구조 공개

2008.09 : Android SDK 1.0 발표

2008.10.21 : 안드로이드를 오픈소스(아파치 라이선스)로 공개

2008.10.22 : 미국 T-Mobile에서 최초의 Android 폰인 G1 출시

2008.10.22 : Android Market 정식 오픈

|  |  |
| --- | --- |
| http://cfs15.tistory.com/image/21/tistory/2009/02/11/06/22/4991f0214382c  Android 1.0, 1.1  2009.02 | http://www.blogcdn.com/kr.engadget.com/media/2009/04/android-cupcake-ofc-logo.jpg  Android 1.5 (Cupcake)  2009.04 (한글 지원) |
| http://1.bp.blogspot.com/_kNUIczJH1OM/Sqmkj5D-jwI/AAAAAAAAAAM/Tr6FGp6-6sI/s320/donut.png  Android 1.6 (Donut)  2009.09 (UI 지원 강화) | http://3.bp.blogspot.com/_kNUIczJH1OM/SuYtG1H7bkI/AAAAAAAAAA0/DfNn2agKf_U/s320/eclair.png  Android 2.1 (Eclair)  2010.01 (홈스크린, 음성인식, 멀티터치) |
| http://1.bp.blogspot.com/_kNUIczJH1OM/S_SfteV5BgI/AAAAAAAAABM/fi0j5ssUrjE/s320/froyo.png  Android 2.2 (Froyo)  2010.05 (성능향상, 테더링) | http://3.bp.blogspot.com/_kNUIczJH1OM/TPWnTpPpv-I/AAAAAAAAABc/KS-JDa4RcUI/s1600/gingerdroid.png  Android 2.3 (Gingerbread)  2010.12 (mVoIP, NFC) |
| http://4.bp.blogspot.com/-9Ez2y71RLWY/TVmisYUffnI/AAAAAAAAABs/iN2LC8JQTNw/s400/android-bumblebee-holo.png  Android 3.0 (Honeycomb)  2011.02 (태블릿 정식 지원) | http://www.clubcity.kr/news/photo/201105/78625_33787_3049.jpg  Android 4.0 (Ice cream Sandwich)  2011.12 (Face Unlock, People, 안드로이드 빔) |
| Android 4.4 (KitKat)  2013.09  웹 브라우저를 구글 크롬으로 변경 | Android 5.0 (Lolipop)  2014.06 |
| Android 6.0 (Marshmallow)  2015.09 | Android 7.0 (Nuga)  2016.08 |
| Android 8.0 (Oreo)  2017.08 |  |

# 안드로이드 구조



Linux 커널 부분은 C 언어로 구현되었다.

Libraries 부분은 C/C++ 언어로 구현되었다.

SurfaceManager, OpenGL/ES, SGL : 3D 그래픽

MediaFramework : 멀티미디어

SQLLite : 데이터베이스,

SSL : 통신

FreeType : 폰트

WebKit : 웹브라우저 엔진

libc : 표준 C 라이브러리

Android Runtime은

Java 컴파일러, Java 가상 기계, Java 표준 라이브러리로 구성된다.

### Application Framework

Application Framework 부분은 안드로이드 앱을 개발하기 위해 사용하는 라이브러리이다.

안드로이드 프로그래밍 수업에서 배울 내용이 Application Framework의 내용이다.

### 라이브러리

프로그래밍 할 때 필요한 소스 코드를 미리 컴파일하여 사용하기 편하도록 묶어 놓은 것을 라이브러리라고 한다.

### 표준 라이브러리

언어와 함께 배포되는 라이브러리를 표준 라이브러리라고 부른다.

### 클래스 라이브러리

라이브러리에 들어있는 소스 코드가 클래스인 경우에 그 라이브러리를 클래스 라이브러리라고 부른다.

### Application Framework Class

애플리케이션의 골격에 해당하는 주요 클래스들을 application framework class 라고 부른다.

이 클래스들이 들어있는 라이브러리를 Application Framework Class Library라고 부른다.

줄여서 Application Framework 이라고 부른다.

# 수업 계획

수업목표:

스마트폰 시장에서 다수를 차지하고 있는 구글이 만든 모바일 운영체제인 안드로이드의 애플리케이션 개발 기술인, Java 안드로이드 프로그래밍 기술을 학습한다. 안드로이드 애플리케이션 개발을 위해 필요한 안드로이드 플랫폼에 대한 이해와 API 클래스 사용법, 그리고 안드로이드 표준 개발 도구인 Android Studio 개발도구 사용법을 학습한다.

수업방법:

수업은 예제 소스 코드 위주로 진행된다. 개발 프로젝트에 필요한 기술들을 예제 소스 코드로 보여주고, 필요한 배경 지식을 설명한다. 학습한 기술을 구현 과제로 실습해 본다.

수업은 다음과 같은 방식으로 진행된다.

(1) 학생들은 강의노트를 따라서 예제 프로젝트를 생성하고 소스 코드를 작성한다.

(2) 예제 프로젝트를 컴파일하고 실행해본다.

(3) 예제 소스 코드를 이해하기 위해 필요한 배경 지식을 교수가 설명한다.

(4) 예제 소스 코드와 그 실행 매커니즘을 교수가 설명한다.

(5) 예제로 학습한 구현 기술을 응용해 보기 위한 구현 과제가 제시된다.

(6) 학생들은 구현 과제를 실습한다.

수업에 필요한 개발환경은 다음과 같다.

JDK 8, Android Studio 2.1.2

평가기준:

구현시험 4회 (4주, 8주, 12주, 16주) = 45%

필기시험 4회 (2주, 6주, 10주, 14주) = 45%

과제 & 출석 = 10%

구현시험은 구현 과제와 유사한 문제가 출제된다.

필기시험은 강의노트의 내용을 위주로 출제된다.

주교재: 강의노트

부교재:

그림으로 쉽게 설명하는 안드로이드 프로그래밍

저자: 천인국

출판사: 생능출판사