



หนังสือคอมพิวเตอร์ที่มีผู้อ่านมากที่สุด

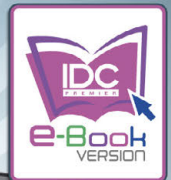


Basic Android App Development

แนะนำ ให้รู้จักระบบปฏิบัติการ Android และสอนการติดตั้งเครื่องมือที่จำเป็นสำหรับการพัฒนาแอปพลิเคชัน สอนการพัฒนาแอปพลิเคชันและเกมอย่างละเอียด รวมไปถึงการพัฒนาแอปพลิเคชันใช้งานฐานข้อมูลและ GPS

เหมาะสำหรับ ผู้เริ่มต้นพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android

Source Code, Android-SDK, Eclipse, Android Development Tool, Java Development Kit, GoogleServicesFramework.apk และ Thai-Keyboard



ดร. จักรชัย โสอินทร์ (CCNP, CCDP, MCP)
พณิศร จันทร้อย

Infopress
Group



10th ANNIVERSARY BEST COMPUTER BOOK

Everything IT, Everything we're expert.

Basic Android App Development

ผู้แต่ง

ดร.จักรชัย โสอินทร์

android.kku@gmail.com

พงษ์ศธร จันทน์ยอย

บรรณาธิการ

กิตินันท์ พลสวัสดิ์

kitinan@infopress.co.th

ออกแบบปก

อนัน วาโษะ

ออกแบบและจัดรูปเล่ม

วุฒิพันธ์ สมพระเมฆ

พิสูจน์อักษร

สุนทรี บรรลือศักดิ์, มนฤดี ศรีอุทโยภาส

ประสานงานการผลิต

สุพัตรา อาจปฐู, ฉัตรชนก แก้วจันทร์

Android เป็นเครื่องหมายการค้าของบริษัท Google และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ

สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอดีซี พรีเมียร์ จำกัด ห้ามลอกเลียนไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใด นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้จัดพิมพ์เท่านั้น

บริษัท ไอดีซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้เทคโนโลยีสารสนเทศทั้งในระดับพื้นฐานและระดับสูง เรายินดีต้อนรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน โดยเฉพาะงานที่เกี่ยวข้องกับสารสนเทศ ท่านผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ editor@infopress.co.th หรือโทรศัพท์หมายเลข 0-2962-1081 (อัตราในมิตี 10 คู่สาย) โทรสาร 0-2962-1084

ข้อมูลทางบรรณานุกรม

ดร.จักรชัย โสอินทร์

Basic Android App Development

นนทบุรี : ไอดีซีฯ, 2554

264 หน้า

1. แอนดรอยด์

2. การเขียนโปรแกรม

I พงษ์ศธร จันทน์ยอย (ผู้แต่งร่วม) II ชื่อเรื่อง

006.76

978-616-200-210-6

มกราคม 2555

ISBN

พิมพ์ครั้งที่ 1

จัดพิมพ์และจัดจำหน่ายโดย

IDC
PREMIER

สำหรับลูกค้า

ลูกค้าสัมพันธ์

โทรศัพท์ 0-2962-1081 ต่อ 121

โทรสาร 0-2962-1084

สำหรับร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2951-1300-2,

0-2591-9446

โทรสาร 0-2591-9447

บริษัท ไอดีซี พรีเมียร์ จำกัด

200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารจัสมินอินเตอร์เนชั่นแนลทาวเวอร์

ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120

โทรศัพท์ 0-2962-1081 (อัตราในมิตี 10 คู่สาย) โทรสาร 0-2962-1084

ราคา 170 บาท

บทบรรณาธิการ

ปัจจุบันโทรศัพท์เคลื่อนที่และแท็บเล็ตบนระบบปฏิบัติการแอนดรอยด์นั้นมีมากมายหลากหลายค่าย หลากหลายแบรนด์ และคาดว่าจะมีการขยายตัวมากขึ้นในอนาคต ซึ่งส่งผลให้มีการพัฒนาแอปพลิเคชันเพื่อตอบสนองความต้องการของผู้ใช้งานมากขึ้นเป็นเงาตามตัว อีกทั้งแอนดรอยด์ยังเป็นระบบปฏิบัติการแบบโอเพ่นซอร์ส ยิ่งส่งผลให้การพัฒนาแอปพลิเคชันแอนดรอยด์ขยายวงกว้างขึ้นเป็นเท่าตัว

การพัฒนาแอปพลิเคชันแอนดรอยด์มีหลักการพัฒนาเหมือนกับการพัฒนาแอปพลิเคชันอื่นๆ เนื่องจากการพัฒนาแอปพลิเคชันแอนดรอยด์จะใช้การเขียนโปรแกรมภาษา Java เป็นภาษาหลักในการพัฒนา ทำให้การเรียนรู้สำหรับพัฒนาแอปพลิเคชันแอนดรอยด์นั้นไม่เป็นเรื่องยากแต่อย่างใด และหนังสือเล่มนี้ก็มีเนื้อหาการพัฒนาแอปพลิเคชันแอนดรอยด์ในระดับเบื้องต้น เพื่อเป็นพื้นฐานที่สำคัญสำหรับผู้สนใจพัฒนาแอปพลิเคชันแอนดรอยด์ โดยมีรูปแบบการนำเสนอที่ทำให้ผู้อ่านเข้าใจง่าย เรียนรู้ได้เร็ว เนื่องจากความรู้ต่างๆ ในหนังสือเล่มนี้เป็นความรู้พื้นฐานที่สำคัญที่ผู้อ่านควรรู้สำหรับนำไปใช้ในการพัฒนาแอปพลิเคชันขั้นสูง ไม่ว่าจะเป็นการพัฒนาแอปพลิเคชันใช้งานระบบ GPS, การพัฒนาแอปพลิเคชันให้ทำงานกับฐานข้อมูล หรือแม้แต่การพัฒนาเกม

ดังนั้น ทางสำนักพิมพ์เชื่อว่าหนังสือเล่มนี้ จะเป็นหนังสือที่เป็นคู่มือเริ่มต้นสำหรับผู้สนใจพัฒนาแอปพลิเคชันแอนดรอยด์ได้อย่างดี อย่างไรก็ตามทางสำนักพิมพ์พร้อมรับคำติชมจากท่านผู้อ่านทุกท่านเพื่อนำมาปรับปรุงแก้ไข และหากหนังสือเล่มนี้มีข้อผิดพลาดประการใด เราต้องขออภัยมา ณ ที่นี้ด้วย

กิตินันท์ พลสวัสดิ์

บรรณาธิการ

สิงหาคม 2554

คำนำ

ทั้งนี้ถึงแม้ว่าแอนดรอยด์จะเป็นระบบปฏิบัติการแบบโอเพ่นซอร์ส แต่ในปัจจุบันจะสังเกตได้ว่ามีหนังสือหรือคู่มือประกอบการพัฒนาแอปพลิเคชันแอนดรอยด์น้อยมาก ซึ่งโดยส่วนใหญ่แล้วหนังสือเหล่านั้นจะอธิบายถึงการนำเสนอแอปพลิเคชันและโทรศัพท์มือถือถึงแอนดรอยด์ในรุ่นต่างๆ จุดประสงค์เพื่อทางการค้าเป็นส่วนใหญ่ โดยที่จะไม่มีการอธิบายถึงขั้นตอนและหลักการพัฒนาแอปพลิเคชันแอนดรอยด์แบบสมบูรณ์ที่ใช้งานได้จริง

ดังนั้น หนังสือเล่มนี้จึงถือได้ว่าเป็นหนังสือภาษาไทยเล่มแรกที่อธิบายถึงหลักการ ทฤษฎี เบื้องต้น และวิธีการพัฒนาแอปพลิเคชัน รวมไปถึงตัวอย่างแอปพลิเคชันต่างๆ ที่จำเป็น ซึ่งผู้อ่านสามารถนำไปใช้พัฒนาแอปพลิเคชันได้จริง อีกทั้งยังสามารถนำไปประยุกต์เพื่อใช้ในการพัฒนาโปรแกรมขั้นสูงได้อีกด้วย

โครงสร้างของเนื้อหาของหนังสือเล่มนี้ จะเริ่มอธิบายตั้งแต่การติดตั้งแอนดรอยด์ การพัฒนาแอปพลิเคชัน รวมไปถึงเทคนิควิธีเพื่อเป็นแนวทางการแก้ไขปัญหา ทั้งนี้โดยภาพรวมสามารถแบ่งโครงสร้างของหนังสือเล่มนี้ออกเป็น 3 ส่วนคือ ส่วนแรกจะเป็นการแนะนำขั้นตอนวิธีการติดตั้งเครื่องมือสำหรับการพัฒนาแอปพลิเคชันแอนดรอยด์ ส่วนที่ 2 จะเป็นการฝึกพัฒนาแอปพลิเคชันในรูปแบบต่างๆ ซึ่งผู้เขียนได้ออกแบบเนื้อหาในบทต่างๆ ให้มีความครอบคลุมกับทักษะต่างๆ ที่จำเป็นสำหรับการนำไปพัฒนาแอปพลิเคชันขั้นสูงได้ และส่วนที่ 3 จะเป็นการแนะนำเทคนิคในการแก้ไขปัญหาสำหรับการพัฒนาแอปพลิเคชัน และการนำเอาโปรแกรมที่ได้พัฒนาไว้บนอีมูเลเตอร์ไปใช้งานหรือติดตั้งบนเครื่องโทรศัพท์เคลื่อนที่จริง

สุดท้ายนี้สำหรับการเขียนหนังสือเล่มนี้ขอขอบพระคุณ คุณพ่อคุณแม่และครอบครัวของผู้เขียนที่คอยเป็นกำลังใจและให้การสนับสนุนมาตลอด คณะบุคคลอีกหลายฝ่ายของภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น ท่านหัวหน้าภาควิชา รศ.สมจิตร อาจอินทร์ ที่เปิดโอกาสให้สอนรายวิชาต่างๆ อีกทั้งท่านอาจารย์ รศ.สุศักดิ์ สงวนพงษ์ และรศ.เย็น ภู่วรรณ ที่เป็นแรงผลักดันสำหรับการเขียนหนังสือต่างๆ รวมไปถึงนายพงษ์ศร จันทัยยอย ที่เป็นส่วนหลักในการเรียบเรียงหนังสือ อีกทั้งนายอนุชิต ขนสี และนางสาวณัฐธิดา วีระมงคลเลิศ ที่ช่วยพิสูจน์อักษรและทดสอบโปรแกรม

ดร.จักรชัย ไสอินทร์

Chakchai So-In, Ph.D.

บทที่ 1 รู้จักแอนดรอยด์ (Android)

แอนดรอยด์คืออะไร.....	1
ประเภทของระบบปฏิบัติการ Android.....	2
สามารถพัฒนาอะไรได้บ้างบน Android	3
สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)	5
ส่วนประกอบของแอปพลิเคชัน (Application Component).....	9
วงจรชีวิตของแอปพลิเคชัน (Application Life Cycle)	10
แบบฝึกหัดท้ายบท.....	12

บทที่ 2 การติดตั้งเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบน Android

การติดตั้ง JDK (Java Development Kit).....	14
การติดตั้งโปรแกรม Eclipse.....	18
เริ่มต้นเขียนโปรแกรมด้วย Eclipse	19
กำหนดให้ Text Editors แสดงหมายเลขบรรทัด	26
การติดตั้ง Android SDK (Android Software Development Kit).....	27
การติดตั้ง ADT (Android Development Tool).....	28
การปรับแต่งค่า ADT และ Android SDK	34
การติดตั้ง SDK Platform	38
การสร้างโทรศัพท์จำลอง (Emulator)	40
แบบฝึกหัดท้ายบท.....	44

บทที่ 3 เริ่มต้นพัฒนาแอปพลิเคชัน

เริ่มต้นสร้างโปรเจกต์.....	46
ออกแบบหน้าจอแอปพลิเคชัน	53
เขียนโค้ดกำหนดการทำงานของแอปพลิเคชัน.....	60
ทดสอบการทำงานของแอปพลิเคชันผ่านโทรศัพท์จำลอง (Emulator).....	62
การส่งออกและนำเข้าโปรเจกต์ใน Eclipse.....	67
การส่งออกโปรเจกต์เป็น Archive File	67
การนำเข้าโปรเจกต์จาก Archive File.....	70
แบบฝึกหัดท้ายบท.....	73

บทที่ 4 พัฒนาแอปพลิเคชันอย่างง่าย

เริ่มต้นสร้างโปรเจกต์.....	76
ออกแบบหน้าจอแอปพลิเคชัน	78
การแก้ไขไฟล์ main.xml.....	78
การแก้ไขไฟล์ string.xml	80
เขียนโค้ดกำหนดการทำงานของแอปพลิเคชัน.....	83



ทดสอบการทำงานของแอปพลิเคชันผ่านโทรศัพท์จำลอง (Emulator).....	86
แบบฝึกหัดท้ายบท.....	91

บทที่ 5 การพัฒนาแอปพลิเคชันรับ/ส่งข้อความ (SMS)

หลักการทำงานของแอปพลิเคชันรับ/ส่งข้อความ	94
เริ่มต้นสร้างโปรเจกต์	95
ออกแบบหน้าจอแอปพลิเคชัน	97
แก้ไขไฟล์ AndroidManifest.xml.....	97
การแก้ไขไฟล์ main.xml	98
เขียนโค้ดกำหนดการทำงานของแอปพลิเคชัน.....	99
ทดสอบการทำงานของแอปพลิเคชันผ่านโทรศัพท์จำลอง (Emulator).....	101
แบบฝึกหัดท้ายบท.....	103

บทที่ 6 การพัฒนาแอปพลิเคชันติดต่อกับ GPS และการอ้างอิงกับระบบแผนที่ (MAP)

ความรู้พื้นฐานเกี่ยวกับ GPS	105
การอ่านตำแหน่งพิกัดของระบบ GPS.....	106
เริ่มต้นสร้างโปรเจกต์.....	107
ออกแบบหน้าจอแอปพลิเคชัน	107
เขียนโค้ดกำหนดการทำงานของแอปพลิเคชัน.....	108
ทดสอบการทำงานของแอปพลิเคชันผ่านโทรศัพท์จำลอง (Emulator).....	111
การแสดงผลค่าพิกัดบน Google Map	114
สร้าง Google API key	114
ออกแบบหน้าจอแอปพลิเคชัน	116
เขียนโค้ดกำหนดการทำงานของแอปพลิเคชัน.....	117
ทดสอบการทำงานของแอปพลิเคชันผ่านโทรศัพท์จำลอง (Emulator).....	119
แบบฝึกหัดท้ายบท.....	119

บทที่ 7 การพัฒนาแอปพลิเคชันติดต่อบนเครือข่าย

การสร้างเซิร์ฟเวอร์อย่างง่าย.....	122
การพัฒนาแอปพลิเคชันฝั่งไคลเอนท์	126
แบบฝึกหัดท้ายบท.....	130

บทที่ 8 การพัฒนาแอปพลิเคชันติดต่อกับฐานข้อมูล

ความรู้เบื้องต้นเกี่ยวกับการทำงานกับฐานข้อมูล	131
การสร้าง Database Adapter.....	132
การใช้งาน Database Adapter	141

ออกแบบหน้าจอแอปพลิเคชัน.....	141
แก้ไขไฟล์ SQLiteDatabase.java	142
การปรับแต่งหน้าจอแอปพลิเคชันด้วย List View	147
แก้ไขไฟล์ main.xml	148
ออกแบบหน้าจอในส่วนแสดงรายการข้อมูล.....	149
การแก้ไขไฟล์ SQLiteDatabase.java.....	152
ออกแบบหน้าจอเพิ่มรายการข้อมูล.....	155
ทดสอบการทำงานของแอปพลิเคชัน.....	165
แบบฝึกหัดท้ายบท.....	166

บทที่ 9 การพัฒนาเกมกราฟิกแบบ 2 มิติ

วิธีการเล่นเกม	168
การสร้างกราฟิกเบื้องต้น.....	169
การปรับแต่งการแสดงผลบนหน้าจอ	172
การจัดการเกี่ยวกับหน้าจอสัมผัส (TouchScreen).....	175
ทดสอบการทำงานของเกม.....	182
แบบฝึกหัดท้ายบท.....	189

บทที่ 10 เทคนิคในการแก้ไขปัญหาสำหรับการพัฒนาแอปพลิเคชัน

การใช้งานเครื่องมือ Emulator Control.....	193
การแสดงผลละเอียดของอุปกรณ์ (Devices).....	195
ส่วนสำคัญของ Emulator Control.....	195
การใช้งาน LogCat.....	198
การตรวจสอบข้อผิดพลาดของแอปพลิเคชันโดยการอ่านค่า Log	200
ค้นหาตำแหน่งของไฟล์ด้วย File Explorer.....	201
แบบฝึกหัดท้ายบท.....	202

บทที่ 11 การติดตั้งแอปพลิเคชัน Android Market uu Emulator

การติดตั้ง Android Market	204
ทดสอบการใช้งาน Android Market บน Emulator	208
แบบฝึกหัดท้ายบท.....	212

บทที่ 12 การติดตั้งคีย์บอร์ดภาษาไทย

แบบฝึกหัดท้ายบท.....	216
----------------------	-----

บทที่ 13 การติดตั้งแอปพลิเคชันบนเครื่องโทรศัพท์จริง

การติดตั้งไดรเวอร์ของเครื่องโทรศัพท์ (Driver Installation).....	218
การส่งออกไฟล์ .apk.....	223



การเตรียมการส่งออกไฟล์ .apk.....	223
การนำไฟล์ .apk ไปติดตั้งบน Emulator เครื่องอื่นๆ.....	227
การนำไฟล์ .apk ไปติดตั้งบนโทรศัพท์เคลื่อนที่จริง	228
การพัฒนาแอปพลิเคชันผ่านทางเครื่องโทรศัพท์เคลื่อนที่จริง (Android Application Development via Mobile Device)	230
แบบฝึกหัดท้ายบท.....	234

ภาคผนวก ความรู้เบื้องต้นการเขียนโปรแกรมภาษาจาวา

ชนิดของข้อมูล (Primitive Data Type)	235
การประกาศตัวแปร (Declaration).....	236
หลักการตั้งชื่อตัวแปร	236
ตัวแปรชนิดข้อความ.....	236
การประกาศค่าคงที่.....	237
ตัวดำเนินการ (Operators)	237
ลำดับความสำคัญของตัวดำเนินการ	239
การแสดงผลอักขระพิเศษ.....	239
การจัดรูปแบบแสดงผลตัวเลข.....	240
การแปลงชนิดข้อมูล.....	240
การแปลงชนิดข้อมูลแบบ Implicit type conversion.....	240
การแปลงชนิดข้อมูลแบบ Explicit type conversion.....	241
การเขียนคำอธิบายโปรแกรม	242
คำสั่งตัดสั้นใจ.....	243
คำสั่งวนลูป.....	244
ตัวแปรอาร์เรย์	245
ตัวแปรอาร์เรย์ 1 มิติ.....	245
ตัวแปรอาร์เรย์หลายมิติ	246
การเขียนโปรแกรมกับตัวแปรอาร์เรย์	247
การประกาศตัวแปรอาร์เรย์.....	247
การกำหนดค่าข้อมูลให้ตัวแปรอาร์เรย์.....	248
การอ้างถึงข้อมูลในตัวแปรอาร์เรย์.....	248
พื้นฐานการเขียนโปรแกรมเชิงวัตถุ.....	249

รู้จักแอนดรอยด์ (Android)

บทนี้เป็นเนื้อหาที่อยู่ในส่วนแรกสำหรับหนังสือเล่มนี้ ซึ่งเป็นเนื้อหาที่อธิบายถึงแอนดรอยด์ในเบื้องต้น และรวมไปถึงการอธิบายถึงสถาปัตยกรรมของแอนดรอยด์ที่สำคัญ ทั้งนี้เพื่อให้ผู้อ่านสามารถนำไปประยุกต์ใช้เพื่อทำความเข้าใจในการพัฒนาแอปพลิเคชัน โดยที่เนื้อหาจะแบ่งออกเป็น 4 ส่วนคือ แอนดรอยด์คืออะไร สถาปัตยกรรมของแอนดรอยด์ ส่วนประกอบของแอปพลิเคชัน และวงจรชีวิตของแอปพลิเคชัน

แอนดรอยด์คืออะไร

แอนดรอยด์ (Android) เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) ซึ่งรวมเอาระบบปฏิบัติการ มิดเดิลแวร์ และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกันเพื่อใช้สำหรับทำงานบนอุปกรณ์พกพาเคลื่อนที่โดยเฉพาะ เช่น โทรศัพท์มือถือ เป็นต้น



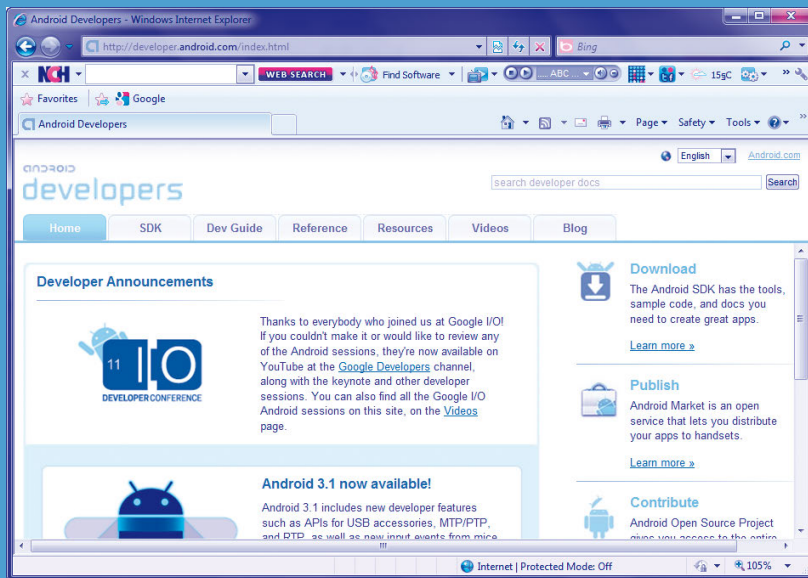
การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลินุกซ์ เคอร์เนล (Linux Kernel) ซึ่งใช้ **Android SDK (Software Development Kit)** เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยใช้ภาษา Java ในการพัฒนา



Android เริ่มพัฒนาโดยบริษัทแอนดรอยด์ และต่อมาได้ผนวกเข้ากับบริษัท Google ซึ่งในเดือนพฤศจิกายน ปี พ.ศ. 2550 ได้มีการร่วมมือกันระหว่างบริษัทชั้นนำมากกว่า 33 บริษัทเพื่อพัฒนาระบบ Android ทั้งบริษัทผู้ให้บริการโทรศัพท์เคลื่อนที่ บริษัทผู้ผลิตอุปกรณ์โทรศัพท์เคลื่อนที่ บริษัทซอฟต์แวร์ และบริษัทเอกชนต่างๆ เช่น HTC, LG, Motorola, Samsung, China Mobile Communications, KDDI, DoCoMo, Sprint/Nextel, T-Mobile, Telecom Italia, Telefonica, Audiance, Broadcom, Intel, Marvel, NVidia, eBay, Packet Video, TAT และ Wind River เป็นต้น โดยใช้ชื่อกลุ่มว่า **OHA (Open Handset Alliance)**

ทั้งนี้ OHA ได้ร่วมมือกันพัฒนามาตรฐานสำหรับการพัฒนาซอฟต์แวร์ระบบเปิด (Open System) หรือ โอเพ่นซอร์ส (Open Source) โดยมีลิขสิทธิ์ตาม Apache Version 2 license

ซึ่งหลักลิขสิทธิ์ของ Apache จะอนุญาตให้ผู้พัฒนาสามารถนำโค้ดที่มีอยู่ไปพัฒนาต่อได้ ทั้งในส่วนของแบบการค้า (Commercial) หรือซอฟต์แวร์กรรมสิทธิ์ (Proprietary) และแบบใช้ฟรีหรือฟรีแวร์ (Freeware) ก็ได้ ผู้อ่านสามารถศึกษาเพิ่มเติมหรือดาวน์โหลดซอฟต์แวร์ได้จากเว็บไซต์ <http://developer.android.com>



▲ เว็บไซต์ Android Developer

ประเภทของระบบปฏิบัติการ Android

เนื่องจากระบบปฏิบัติการ Android เป็นซอฟต์แวร์ระบบเปิด จึงอนุญาตให้นักพัฒนาหรือผู้ที่สนใจสามารถดาวน์โหลด Source Code ได้ ทำให้มีผู้พัฒนาจากหลายๆ ฝ่ายนำ Source Code มาปรับแต่งและพัฒนาสร้างแอปพลิเคชันบนระบบ Android ในแบบฉบับของตนเองมากขึ้น โดยสามารถแบ่งประเภทของระบบ Android ออกเป็นกลุ่มๆ ได้ 3 ประเภท ดังต่อไปนี้

- ◆ **Android Open Source Project (AOSP)** เป็นระบบ Android ประเภทแรกที่ทางบริษัท Google เปิดให้สามารถนำ Source Code ไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ ได้โดยไม่ต้องเสียค่าใช้จ่าย
- ◆ **Open Handset Mobile (OHM)** เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่ม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาระบบ Android ในแบบฉบับของตนเอง โดยมีรูปร่างหน้าตาการแสดงผล และฟังก์ชันการใช้งานที่แตกต่างกัน รวมไปถึงอาจจะมีเอกลักษณ์และรูปแบบการใช้งานเป็นของตัวเองแต่ละบริษัท และโปรแกรมแอนดรอยด์ประเภทนี้ก็จะได้รับสิทธิบริการเสริมต่างๆ จากกูเกิลที่เรียกว่า **GMS (Google Mobile Service)** ซึ่งเป็นบริการเสริมที่ทำให้ระบบ Android มีประสิทธิภาพมากขึ้นนั่นเอง
- ◆ **Cooking หรือ Customize** เป็นระบบ Android ที่นักพัฒนานำเอาซอร์สโค้ดจากแหล่งต่างๆ มาปรับแต่งให้อยู่ในแบบฉบับของตนเอง ซึ่งการพัฒนาจะต้องปลดล็อกสิทธิในการใช้งานอุปกรณ์ (Unlock) เสียก่อนจึงจะสามารถติดตั้งได้ ทั้งนี้ระบบ Android ประเภทนี้ถือได้ว่าเป็นประเภทที่มีความสามารถสูงที่สุด เนื่องจากจะได้รับการปรับแต่งขีดความสามารถต่างๆ ให้มีความเข้ากันได้กับอุปกรณ์นั้นๆ จากผู้ใช้งานจริง

อย่างไรก็ตามการที่ระบบ Android ของแต่ละผู้ผลิตจะได้รับบริการ Google Mobile Service (GMS) นั้นผู้ผลิตจะต้องทดสอบระบบและขออนุญาตกับทางกูเกิลเสียก่อนจึงจะสามารถนำเครื่องออกสู่ตลาดได้

สามารถพัฒนาอะไรได้บ้างบน Android

การพัฒนาแอปพลิเคชันบน Android มีไลบรารี (Library) สำหรับใช้งานมากมายที่อำนวยความสะดวกให้แก่ นักพัฒนา ในหัวข้อนี้จะขอยกตัวอย่างเฉพาะไลบรารีที่น่าสนใจ ตัวอย่างเช่น

Note

ไลบรารี (Library) สำหรับใช้ในการพัฒนาแอปพลิเคชันจะมีโครงสร้างเป็นมาตรฐาน ซึ่งภายในไลบรารีนั้นจะมีโค้ดที่นำกลับมาใช้ได้ (Reusable) อยู่เป็นจำนวนมาก ซึ่งจะทำให้ นักพัฒนาประหยัดเวลาในการพัฒนาแอปพลิเคชันต่างๆ ได้อย่างมาก เนื่องจากไม่มีความจำเป็นต้องเขียนโค้ดเดิมๆ ซ้ำๆ อีก เมื่อมีการพัฒนาแอปพลิเคชันใหม่หรือแอปพลิเคชันที่มีฟังก์ชันการทำงานที่คล้ายคลึงกัน

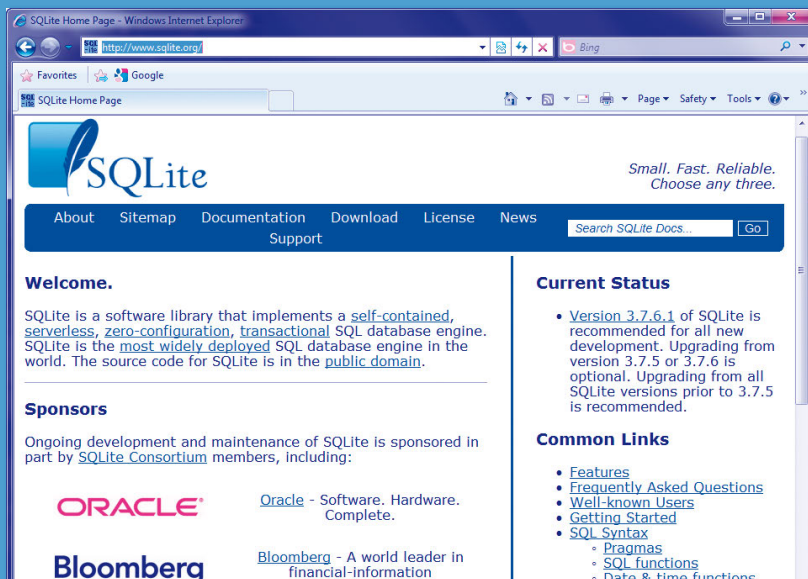
ดังนั้น ไลบรารีนี้จะทำหน้าที่หลักคือ การนำกลับมาใช้หรือพัฒนาต่อยอดการทำงานหลักให้ดีขึ้นนั่นเอง

- ◆ **Dalvik Virtual Machine (VM)** เป็นส่วนของการสร้างเครื่องจำลองแบบเสมือนที่มีการออกแบบให้เหมาะสมกับอุปกรณ์เคลื่อนที่หรืออุปกรณ์มือถือ
- ◆ **Integrated Browser** เป็นการผนวก Web Browser เข้าไว้กับแอนดรอยด์ทั้งนี้พื้นฐานมาจากซอฟต์แวร์เว็บคิต (WebKit)



▲ เว็บไซต์ WebKit

- ◆ **Optimized Graphic** เป็นส่วนสนับสนุนการทำงานแบบกราฟิกทั้งในส่ว 2 มิติและ 3 มิติ โดยใช้เครื่องมือ OpenGL
- ◆ **SQLite** เป็นส่วนสนับสนุนการทำงานสำหรับการจัดเก็บฐานข้อมูล

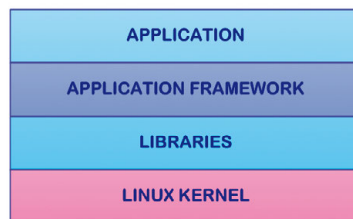


▲ เว็บไซต์ SQLite

- ◆ **Media Support** เป็นส่วนสนับสนุนการทำงานแบบสื่อประสม หรือมัลติมีเดีย เช่น ออดิโอ วิดีโอ และรูปภาพ
- ◆ **GSM Telephony** เป็นส่วนรองรับการทำงานบนระบบโทรศัพท์เคลื่อนที่แบบ GSM (Global System for Mobile Communications)
- ◆ **Bluetooth, EDGE, 3G, WiFi** เป็นส่วนรองรับการทำงานกับ Bluetooth, EDGE (Enhanced Data rates for GSM Evolution) 3G และ WiFi
- ◆ **Camera, GPS, Compass, Accelerometer** เป็นส่วนสนับสนุนการทำงานของระบบกล้องถ่ายรูป, ระบบกำหนดตำแหน่งบนโลก หรือ GPS (Global Positioning System), เข็มทิศ และการวัดอัตราความเร็ว
- ◆ **Rich Development Environment** เป็นส่วนสนับสนุนฟังก์ชันต่างๆ ที่ช่วยในการพัฒนาแอปพลิเคชัน เช่น Emulator, Debugging Tool, Memory and Performance Profiling และ Plug-in สำหรับเครื่องมือ Eclipse

สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)

สถาปัตยกรรมของการออกแบบแอนดรอยด์ (Android Architecture) นั้นถูกแบ่งออกเป็นลำดับชั้น หรือจะเรียกว่า **Layer** โดยที่แต่ละชั้นจะเรียกใช้บริการจากระดับชั้นที่อยู่ด้านล่างของตัวเอง ซึ่งสามารถแบ่งออกเป็น 4 ชั้นหลักคือ ชั้นลินุกซ์เคอร์เนล (Linux Kernel), ชั้นไลบรารี (Library), ชั้นแอปพลิเคชันเฟรมเวิร์ค (Application Framework) และชั้นแอปพลิเคชัน (Application)



▲ สถาปัตยกรรม Android

- ◆ **ชั้นแอปพลิเคชัน (Application)** ชั้นนี้เป็นชั้นบนสุดของโครงสร้าง Android ซึ่งเป็นส่วนของแอปพลิเคชันที่พัฒนาขึ้นมาใช้งาน เช่น แอปพลิเคชันรับ/ส่งอีเมล, แอปพลิเคชันโทรศัพท์ (Phone Dial), แอปพลิเคชันเว็บเบราว์เซอร์ (Web Browser) เป็นต้น ทั้งนี้โปรแกรมในชั้นแอปพลิเคชันนั้นจะอยู่ในรูปแบบของไฟล์ .apk ซึ่งโดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app ของโทรศัพท์เคลื่อนที่



▲ สถาปัตยกรรมชั้นแอปพลิเคชัน



▲ ตัวอย่างแอปพลิเคชัน

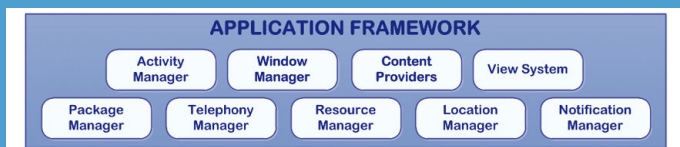
Tips

แอปพลิเคชันและวิดเจ็ต (Widget) มีลักษณะที่เหมือนกันคือ เป็นโปรแกรมที่ถูกพัฒนาขึ้นเหมือนกัน แต่จะต่างกันที่แอปพลิเคชันนั้นจะทำงานโดยใช้พื้นที่เต็มจอ (Full Screen) แต่ในส่วนของวิดเจ็ต หรือบางครั้งเรียกว่า แก็ดเจ็ต (Gadget) จะทำงานบนพื้นที่ขนาดเล็กหรือไม่เต็มหน้าจอเท่านั้น

- ◆ **ชั้นแอปพลิเคชันเฟรมเวิร์ก (Application Framework)** โดยปกติแล้วนักพัฒนาสามารถเรียกใช้งาน Android ผ่าน API (Application Programming Interface) ได้ ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งานซ้ำของ Application Component ซึ่งมีตัวอย่างแอปพลิเคชันเฟรมเวิร์ค ดังนี้
 - **View System** เป็นส่วนควบคุมการทำงานสำหรับการสร้างแอปพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser
 - **Location Manager** เป็นส่วนดูแลตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่
 - **Content Provider** เป็นส่วนควบคุมการเข้าถึงของข้อมูลที่มีการใช้งานร่วมกัน (Share Data) ระหว่างแอปพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ใช้ (Contact)
 - **Resource Manager** เป็นส่วนดูแลการเข้าใช้ข้อมูลต่างๆ ที่ไม่ใช่โค้ด เช่น localized strings, graphics และ layout ซึ่งจะอยู่ในไดเรกทอรี res/

ทั้งนี้ข้อมูลต่างๆ ในส่วนนี้จะถูกคอมไพล์และผนวกเข้ากับโปรแกรมที่เขียนขึ้น ณ เวลาการ Build โดยแอนดรอยด์จะใช้เครื่องมือ `adpt` สำหรับการคอมไพล์ และหลังจากการคอมไพล์แล้วจะสร้างคลาสที่ชื่อว่า `R` ซึ่งเป็นส่วนที่ระบุถึงข้อมูลต่างๆ ที่ใช้อ้างอิงสำหรับแอปพลิเคชันที่ถูกพัฒนาขึ้น

- **Notification Manager** เป็นส่วนควบคุมอีเวนต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความและการแจ้งเตือนต่างๆ
- **Activity Manager** เป็นส่วนควบคุม Life Cycle ของแอปพลิเคชัน



▲ สถาปัตยกรรมชั้นแอปพลิเคชันเฟรมเวิร์ค

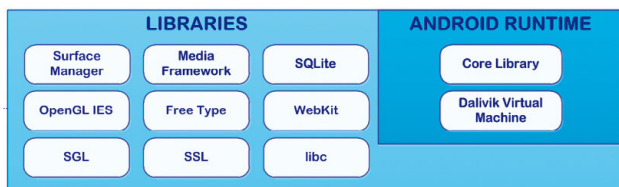
Note

API (Application Program Interface) อ่านว่า เอพีไอ หรือส่วนต่อประสานโปรแกรมประยุกต์คือ วิธีการเฉพาะสำหรับการเรียกใช้งานระบบปฏิบัติการหรือแอปพลิเคชันอื่นๆ ซึ่งทำหน้าที่เชื่อมต่อการทำงานระหว่างแอปพลิเคชันกับระบบปฏิบัติการ ทั้งนี้การที่แอปพลิเคชันจะเชื่อมต่อการทำงานกับระบบปฏิบัติการได้นั้น จำเป็นต้องมี API เป็นตัวเชื่อม ซึ่งหากไม่มีการเปิดเผย API ของระบบปฏิบัติการออกมาแล้ว ผู้ที่พัฒนาแอปพลิเคชันจะมีความลำบากเมื่อต้องการพัฒนาแอปพลิเคชันให้มีความเข้ากันได้กับระบบปฏิบัติการได้อย่างมีประสิทธิภาพ

- ◆ **ชั้นไลบรารี (Library)** แอนดรอยด์ได้รวบรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นต่อการพัฒนาโปรแกรมเอาไว้มากมาย ซึ่งถูกเขียนไว้ด้วยภาษา C และ C++ โดยตัวอย่างของไลบรารีที่สำคัญที่ผู้อ่านควรารู้คือ
 - **System C library** เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc)
 - **Media Libraries** เป็นกลุ่มการทำงานมัลติมีเดีย เช่น ออดิโอ วิดีโอ รวมถึงรูปภาพต่างๆ เช่น ไฟล์สกุล MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG
 - **Surface Manager** เป็นกลุ่มการจัดการรูปแบบของหน้าจอ การวาดหน้าจอ
 - **2D/3D library** เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (Scalable Graphics Library) และแบบ 3 มิติ หรือ OpenGL
 - **FreeType** เป็นกลุ่มของบิตแมป (Bitmap) และเวกเตอร์ (Vector) สำหรับการเรนเดอร์ (Render) ภาพ
 - **SQLite** เป็นกลุ่มของฐานข้อมูล ซึ่งเป็นระบบฐานข้อมูลที่ใช้เช่นเดียวกับซอฟต์แวร์ Firefox และ Apple iPhone ทั้งนี้นักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลของแอปพลิเคชันต่างๆ ได้
 - **Browser Engine** เป็นกลุ่มของการแสดงผลบนเว็บเบราว์เซอร์โดยอยู่บนพื้นฐานของ Webkit ซึ่งจะมีลักษณะคล้ายคลึงกับ Google Chrome, Safari และ Nokia S60

สำหรับการเรียกใช้แอปพลิเคชันต่างๆ ในชั้นไลบรารี จะไม่สามารถเรียกใช้แอปพลิเคชันในระดับเดียวกับตัวเองได้ โดยจะต้องเรียกใช้แอปพลิเคชันในชั้นที่สูงกว่าเท่านั้นจึงจะสามารถเรียกใช้ได้ นอกจากนี้ในชั้นไลบรารีนี้ Android ยังแบ่งเป็นชั้นย่อยที่เรียกว่า **Android Runtime** ซึ่งประกอบไปด้วย 2 ส่วนหลักคือ Dalvik VM และ Core Java Library

- **Dalvik VM (Virtual Machine)** ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานอุปกรณ์พกพาเคลื่อนที่ อย่างไรก็ตามสิ่งที่แตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM ซึ่งจะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์พกพามากกว่า .class โดยเฉพาะอย่างยิ่งประสิทธิภาพในการใช้พลังงานจากแบตเตอรี่
- **Core Java Library** ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)



▲ สถาปัตยกรรมชั้นไลบรารี

Tips

ความแตกต่างระหว่าง Java VM และ Dalvik VM คือ การแปลคำสั่ง หรือมีอินเทอร์พรีเตอร์ (Interpreter) ที่แตกต่างกัน โดยส่วนของ Java VM นั้นจะมีรูปแบบเป็นแบบสแต็ก (Stack) แต่ส่วนของ Dalvik VM นั้นจะเป็นแบบรีจิสเตอร์ (Register)

โดยที่ Java VM นั้นจะเป็นการอ่าน byte code (class) ของจาวา จากนั้นจะแปล byte code เหล่านั้นให้เป็นคำสั่งต่างๆ และโหลดเข้าไปเก็บไว้ในหน่วยความจำแบบสแต็ก

แต่ในส่วน Dalvik VM นั้นจะมีการทำงานที่คล้ายคลึงกันเพียงแต่จะเป็นการโหลดคำสั่งต่างๆ เข้าไปเก็บไว้ในหน่วยความจำแบบรีจิสเตอร์แทน

โดยภาพรวมแล้วทั้ง 2 ประเภทจะมีการทำงานที่คล้ายกัน แต่ Java VM นั้นจะมีการสร้างคำสั่งต่างๆ มากกว่า ซึ่งทำให้ไม่เหมาะสมกับการนำไปประยุกต์ใช้ในอุปกรณ์เคลื่อนที่ ซึ่งโดยปกติแล้วจะมีทรัพยากรที่จำกัด เช่น ชีตจำกัดของ CPU (Central Processing Unit) และ RAM (Random Access Memory)

- ◆ **ชั้นลินุกซ์เคอร์เนล (Linux Kernel)** ระบบ Android อยู่บนพื้นฐานของระบบปฏิบัติการ Linux โดยชั้น Linux Kernel นั้นมีฟังก์ชันการทำงานหลายๆ ส่วน ซึ่งแต่ละส่วนถูกพัฒนาขึ้นด้วยภาษา C เช่น การจัดการหน่วยความจำ (Memory Management) การจัดการโพรเซส (Process Management) การเชื่อมต่อเครือข่าย (Networking) และฟังก์ชันการทำงานส่วนอื่นที่เกี่ยวข้องกับระบบปฏิบัติการ ทั้งนี้ นักพัฒนาจะไม่มีสิทธิเข้าถึงส่วนนี้ได้โดยตรง อย่างไรก็ตาม นักพัฒนาสามารถเข้าถึงระบบปฏิบัติการ Linux ได้จากชุดคำสั่ง Command Prompt เช่น adb shell ซึ่งจะสามารถใช้เครื่องมือต่างๆ ได้ เช่น การเข้าดูระบบไฟล์ (File System), โพรเซสการคัดลอกไฟล์ (Copy File) เป็นต้น

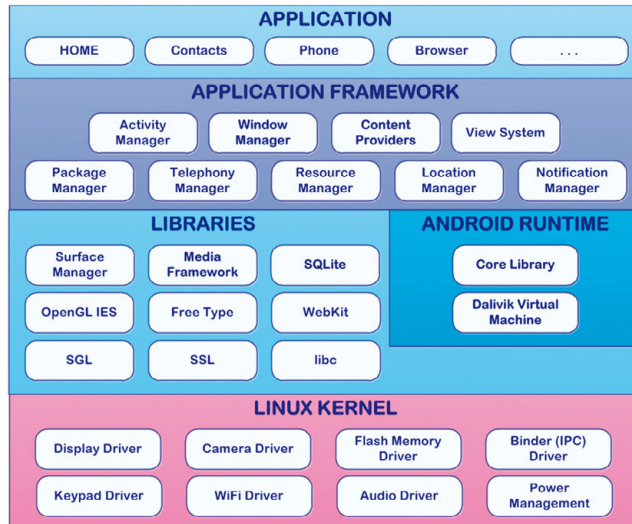


▲ สถาปัตยกรรมชั้น Linux Kernel

Note

Linux Kernel ถูกสร้างขึ้นโดย ลินุส เบนดิคต์ โตร์วัลด์ส (Linus Benedict Torvalds) ในปี ค.ศ. 1991 ซึ่งขณะนั้นศึกษาอยู่ที่มหาวิทยาลัยเฮลซิงกิประเทศฟินแลนด์ (The University of Helsinki)

จากที่กล่าวมาพอสรุปเป็นภาพของสถาปัตยกรรม Android ได้ดังนี้



▲ สถาปัตยกรรมของ Android

ส่วนประกอบของแอปพลิเคชัน (Application Component)

คุณลักษณะอย่างหนึ่งของแอนดรอยด์ก็คือ เป็นแอปพลิเคชันที่สามารถใช้เป็นส่วนประกอบของแอปพลิเคชันอื่นๆ ได้ ตัวอย่างเช่น ถ้าผู้อ่านต้องการสร้างแอปพลิเคชันให้แสดงการเลื่อนของรายการรูปภาพ โดยที่ผู้อ่านอาจมีแอปพลิเคชันส่วนอื่นที่ได้พัฒนาไว้แล้ว ผู้อ่านสามารถเรียกใช้แอปพลิเคชันในส่วนที่มีอยู่มาพัฒนาต่อได้ โดยที่ผู้อ่านไม่จำเป็นต้องพัฒนาขึ้นมาเอง เพื่อความสะดวกและรวดเร็วในการพัฒนาแอปพลิเคชัน ซึ่งเรียกแอปพลิเคชันเหล่านั้นว่า **Application Component**

ซึ่ง Application Component ของแอนดรอยด์สามารถแบ่งออกเป็น 4 ส่วนคือ Activity, Service, Content Provider และ Broadcast Receiver

- ◆ **Activity** คือ หน้าจอที่ติดต่อกับผู้ใช้ ทั้งนี้ในแต่ละแอปพลิเคชันอาจจะมีมากกว่า 1 หน้าจอ หรือ 1 Activity ซึ่งแต่ละ Activity จะทำหน้าที่เก็บสถานะการใช้งานในส่วนต่างๆ ตัวอย่างเช่น
 - ในการแสดงรายการเมนู นักพัฒนาสามารถเลือกให้รายการเมนูที่แสดงออกมามีภาพและคำบรรยายได้ภาพได้
 - สำหรับแอปพลิเคชันส่งข้อความอาจจะมี Activity หนึ่งที่แสดงรายการของส่วนติดต่อในการส่งข้อความ อีก Activity หนึ่งจะเป็นส่วนของการเลือกการติดต่อและ Activity อื่นๆ จะทำหน้าที่ดูข้อความเก่าที่ถูกส่งมาแล้ว เป็นต้น
- ◆ **Service** คือ งานหรือบริการต่างๆ ที่ทำงานอยู่เบื้องหลัง เช่น Service ใช้ที่เปิดดนตรีอยู่ขณะที่ผู้ใช้งานทำงานอื่นๆ หรือใช้แอปพลิเคชันอื่นๆ ไปด้วย



- ◆ **Broadcast and Intent Receiver** คือ การตอบสนองซึ่งโดยปกติแล้ว Broadcast Receiver จะเป็นการตอบสนองต่อการเกิดอีเวนต์ของระบบในวงกว้าง เช่น การประกาศเตือนว่าแบตเตอรี่ใกล้จะหมดแล้ว เป็นต้น นอกจากนี้ Intent Receiver เป็นส่วนทำให้แอปพลิเคชันอื่นๆ เข้าถึงการทำงานของ Activity และ Service ซึ่งในการปฏิบัติงานแต่ละอย่างเป็นการตอบสนองการร้องขอจากข้อมูลหรือบริการของ Activity อื่นๆ
- ◆ **Content Provider** คือ ส่วนของการให้บริการข้อมูลสำหรับแต่ละแอปพลิเคชัน ทั้งนี้ข้อมูลสามารถเก็บอยู่ในรูปแบบของระบบไฟล์ หรือฐานข้อมูลก็ได้ เช่น Google สามารถเข้าใช้งานข้อมูลร่วมกับผู้ใช้งานได้ในแอปพลิเคชันที่ต้องการข้อมูลของผู้ใช้งาน

Note

การอนุญาตให้การเข้าใช้งานในแต่ละส่วนประกอบของแอปพลิเคชัน (Activating Component) หรือ Intent คือ วิธีที่อธิบาย Action ต่างๆ เช่น การเลือกรูปภาพ การเปิดโปรแกรมฟังเพลง ซึ่งทั่วไปแล้ว Intent จะถูกใช้สำหรับการสร้างแอปพลิเคชันใหม่จากแอปพลิเคชันเดิม โดยที่นักพัฒนาไม่จำเป็นต้องสร้างใหม่ทั้งหมด (Mobile Mashup) และ Intent อนุญาตให้ผู้พัฒนาสามารถติดต่อกันระหว่างแอปพลิเคชันและบริการที่แตกต่างกันได้

Tips

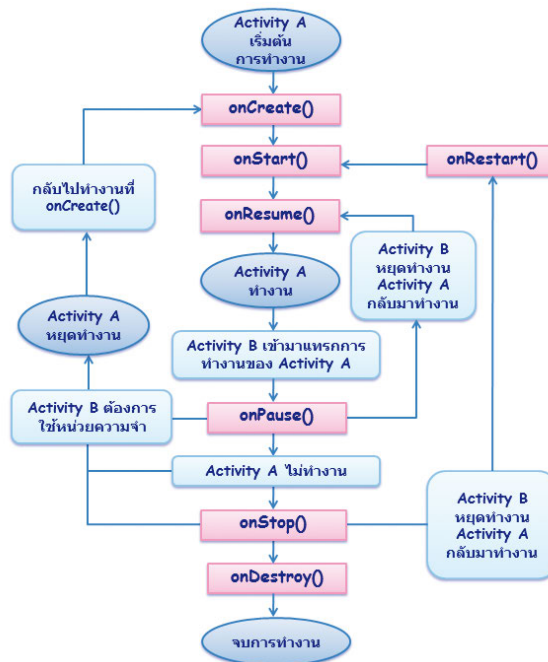
โดยปกติแล้วในแต่ละแอปพลิเคชันจะทำงานแยกกันในแต่ละโพรเซส (Process) ซึ่งมีการทำงานที่แยกจากกันโดยชัดเจน ในกรณีที่นักพัฒนาต้องการเข้าถึงการทำงานหรือการขอสิทธิเข้าใช้นั้น นักพัฒนาจะต้องกำหนดการขอเข้าใช้ที่ไฟล์ AndroidManifest.xml ตัวอย่างเช่น

- ◆ INTERNET เป็นการอนุญาตให้เข้าใช้อินเทอร์เน็ต
- ◆ READ_CONTACTS เป็นการอนุญาตให้อ่านข้อมูลผู้ใช้
- ◆ WRITE_CONTACTS เป็นการอนุญาตให้เขียนข้อมูลผู้ใช้
- ◆ RECEIVE_SMS เป็นการอนุญาตให้ตรวจสอบการรับ SMS (Short Message Service)
- ◆ ACCESS_COARSE_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งแบบคร่าวๆ (Coarse)
- ◆ ACCESS_FINE_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งที่มีความถูกต้องสูง (Fine) เช่น GPS

วงจรชีวิตของแอปพลิเคชัน (Application Life Cycle)

โดยปกติแล้วแอปพลิเคชันจะทำงานแยกกันในแต่ละโพรเซส และในแต่ละโพรเซสอาจจะมี Activity/Service ที่ทำงานอยู่มากกว่า 1 Activity/Service ดังนั้น ในแต่ละแอปพลิเคชันอาจจะมีมากกว่า 1 Activity ซึ่งในการเริ่มทำงานของ Activity จะเริ่มด้วย startActivity() สำหรับ Activity แบบซิงโครนัส (Synchronous) และจะเริ่มด้วย startSubActivity() สำหรับ Activity แบบอะซิงโครนัส (Asynchronous) โดยในแต่ละ Activity จะมีวงจรชีวิต (Life Cycle) ที่แยกจากกันโดยชัดเจน ซึ่งมีสถานะการทำงานหลักดังนี้

- ◆ **onCreate** (Bundle savedInstanceState) ส่วนนี้จะถูกเรียกใช้งานเมื่อเริ่มทำงาน ในกรณีที่มีการเรียกใช้งานเมธอด (Method) นี้ Android Framework จะนำ Bundle object ไปบันทึกไว้ใน Activity ก่อนที่ Activity จะทำงาน ซึ่งจากนั้นจะตามด้วยฟังก์ชัน onStart()
- ◆ **onStart()** ส่วนนี้เป็นการระบุว่า Activity นั้นๆ จะถูกแสดงขึ้นมา จากนั้นสถานะจะถูกย้ายไปเป็นสถานะ onResume แต่ถ้า Activity นั้นไม่สามารถทำงานได้ด้วยเหตุผลบางประการ สถานะจะถูกย้ายไปเป็นสถานะ onStop
- ◆ **onRestart()** ส่วนนี้จะเป็นการระบุว่า Activity นั้นจะถูกแสดงขึ้นมาอีกครั้งหนึ่ง ซึ่งจะตามด้วยสถานะ onStart()
- ◆ **onResume()** ส่วนนี้จะถูกเรียกเมื่อ Activity นั้นๆ มีการติดต่อปฏิสัมพันธ์กับผู้ใช้งาน เช่น นักพัฒนาต้องการเรียก Activity นั้นให้ขึ้นมาทำงานอีกรอบหนึ่ง หลังจากที่ Activity นั้นอยู่ในสถานะ onPause
- ◆ **onPause()** ส่วนนี้จะถูกเรียกใช้เมื่อ Activity นั้นจะถูกเปลี่ยนไปเป็นการทำงานทางเบื้องหลัง (Background)
- ◆ **onStop()** ส่วนนี้จะถูกเรียกใช้เมื่อผู้ใช้ไม่ต้องการใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำงานที่ Activity นั้นอีกครั้งหนึ่ง หรือตามด้วยสถานะ onDestroy() เมื่อต้องการปิด Activity นั้นๆ
- ◆ **onDestroy()** ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละ Activity



▲ Activity Life Cycle

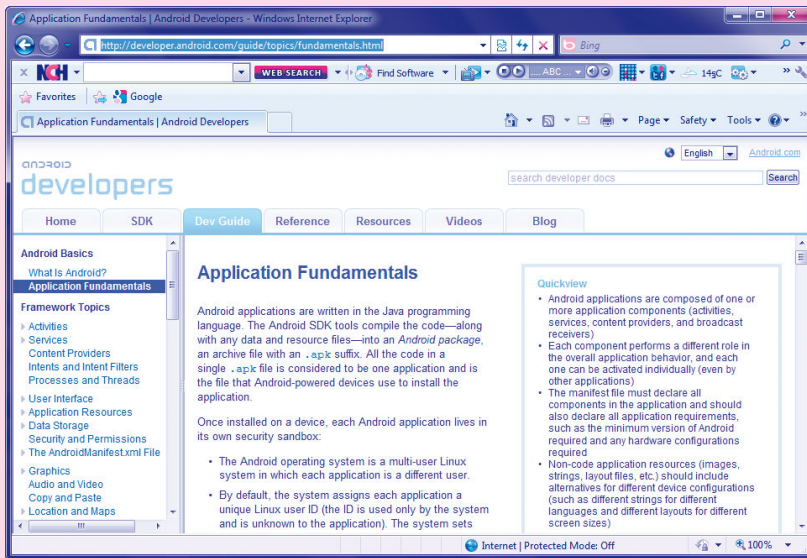


Note

สำหรับ Android นั้นแอปพลิเคชันที่อยู่เบื้องหน้านั้นมีได้เพียง 1 หน้าจอเท่านั้น ซึ่งโดยปกติแล้วจะใช้พื้นที่ทั้งหน้าจอของโทรศัพท์เคลื่อนที่ ยกเว้นส่วนของแถบสถานะภาพ หรือแถบสถานะ (Status bar)

แต่อย่างไรก็ตามผู้ใช้สามารถเลือกหรือเปลี่ยนแอปพลิเคชันขึ้นมาใช้งานได้ ทั้งนี้แอปพลิเคชันทั้งหมดจะถูกเก็บค่าสถานะไว้ที่แอปพลิเคชันสแต็ก (Application Stack) ซึ่งควบคุมโดย Activity Manager

ดังนั้น เมื่อผู้ใช้คลิกปุ่มกลับ หรือ Back ในส่วนของ Activity Manager ก็จะได้เอาแอปพลิเคชันบนสแต็กขึ้นมาทำงาน ทั้งนี้ผู้อ่านสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์ <http://developer.android.com/guide/topics/fundamentals.html>



แบบฝึกหัดท้ายบท

- ◆ จงอธิบายความแตกต่างระหว่าง Content Provider และ Resource Manager
- ◆ จงอธิบายความสัมพันธ์ระหว่างแอปพลิเคชัน, โพรเซส และแอสกิติวิตี้