

# Android on x86

An Introduction to Optimizing for  
Intel® Architecture



Iggy Krajci

Darren Cummings



Apress  
**open**

# Android on x86: An Introduction to Optimizing for Intel® Architecture

Iggy Krajci and Darren Cummings

Copyright © 2013 by Apress Media, LLC, all rights reserved

**ApressOpen Rights:** You have the right to copy, use and distribute this Work in its entirety, electronically without modification, for non-commercial purposes only. However, you have the additional right to use or alter any source code in this Work for any commercial or non-commercial purpose which must be accompanied by the licenses in (2) and (3) below to distribute the source code for instances of greater than 5 lines of code. Licenses (1), (2) and (3) below and the intervening text must be provided in any use of the text of the Work and fully describes the license granted herein to the Work.

(1) **License for Distribution of the Work:** This Work is copyrighted by Apress Media, LLC, all rights reserved. Use of this Work other than as provided for in this license is prohibited. By exercising any of the rights herein, you are accepting the terms of this license. You have the non-exclusive right to copy, use and distribute this English language Work in its entirety, electronically without modification except for those modifications necessary for formatting on specific devices, for all non-commercial purposes, in all media and formats known now or hereafter. While the advice and information in this Work are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

If your distribution is solely Apress source code or uses Apress source code intact, the following licenses (2) and (3) must accompany the source code. If your use is an adaptation of the source code provided by Apress in this Work, then you must use only license (3).

(2) **License for Direct Reproduction of Apress Source Code:** This source code, from *Android on x86, ISBN 978-1-4302-6130-8* is copyrighted by Apress Media, LLC, all rights reserved. Any direct reproduction of this Apress source code is permitted but must contain this license. The following license must be provided for any use of the source code from this product of greater than 5 lines wherein the code is adapted or altered from its original Apress form. This Apress code is presented AS IS and Apress makes no claims to, representations or warranties as to the function, usability, accuracy or usefulness of this code.

(3) **License for Distribution of Adaptation of Apress Source Code:** Portions of the source code provided are used or adapted from *Android on x86, ISBN 978-1-4302-6130-8* copyright Apress Media LLC. Any use or reuse of this Apress source code must contain this License. This Apress code is made available at [Apress.com/9781430261308](http://Apress.com/9781430261308) as is and Apress makes no claims to, representations or warranties as to the function, usability, accuracy or usefulness of this code.

ISBN-13 (pbk): 978-1-4302-6130-8

ISBN-13 (electronic): 978-1-4302-6131-5

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning  
Lead Editors: Jeffrey Pepper (Apress); Stuart Douglas (Intel)  
Contributing Editor: Sarah Yost (Intel)  
Coordinating Editor: Mark Powers  
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

# About ApressOpen

## What Is ApressOpen?

- ApressOpen is an open access book program that publishes high-quality technical and business information.
- ApressOpen eBooks are available for global, free, noncommercial use.
- ApressOpen eBooks are available in PDF, ePub, and Mobi formats.
- The user friendly ApressOpen free eBook license is presented on the copyright page of this book.

*I'd like to dedicate this book to my mother Julie. I can never thank you enough for making me into the man that I am today. Thank you.*

—Iggy

*I dedicate this book to my SAIFE team - you guys are awesome!*

—Darren

# Contents at a Glance

<b>About the Authors.....</b>	<b>xxi</b>
<b>Acknowledgments.....</b>	<b>xxiii</b>
<b>Introduction.....</b>	<b>xxv</b>
<b>■ Chapter 1: History and Evolution of the Android OS.....</b>	<b>1</b>
<b>■ Chapter 2: The Mobile Device and Operating System Landscape.....</b>	<b>9</b>
<b>■ Chapter 3: Beyond the Mobile App—A Technology Foundation ...</b>	<b>17</b>
<b>■ Chapter 4: Android Development—Business Overview and Considerations.....</b>	<b>25</b>
<b>■ Chapter 5: The Intel Mobile Processor.....</b>	<b>33</b>
<b>■ Chapter 6: Installing the Android SDK for Intel Application Development.....</b>	<b>47</b>
<b>■ Chapter 7: Creating and Porting NDK-Based Android Applications.....</b>	<b>75</b>
<b>■ Chapter 8: Debugging Android.....</b>	<b>131</b>
<b>■ Chapter 9: Performance Optimizations for Android Applications on x86.....</b>	<b>185</b>
<b>■ Chapter 10: x86 NDK and C/C++ Optimizations.....</b>	<b>259</b>
<b>■ Chapter 11: Using Intel Hardware Accelerated Execution Manager on Windows, Mac OS, and Linux to Speed Up Android on x86 Emulation.....</b>	<b>285</b>

■ <b>Chapter 12: Performance Testing and Profiling Apps with Platform Tuning</b> .....	<b>303</b>
■ <b>Appendix A: References</b> .....	<b>331</b>
<b>Index</b> .....	<b>343</b>

# Contents

- About the Authors..... xxi**
- Acknowledgments ..... xxiii**
- Introduction .....xxv**
- Chapter 1: History and Evolution of the Android OS ..... 1**
  - Origins ..... 1
    - The First Distribution of Android ..... 1
    - Open Source Apache License ..... 2
  - What Is Android? ..... 2
    - Applications ..... 3
    - Application Frameworks ..... 4
    - Native Libraries ..... 4
    - Android Runtime ..... 4
    - Linux Kernel ..... 5
  - The Open Handset Alliance..... 5
  - Android Open Source Project ..... 5
    - Astro (1.0) ..... 6
    - Cupcake (1.5)..... 6
    - Donut (1.6) ..... 6
    - Éclair (2.0/2.1) ..... 6
    - Froyo (2.2.x)..... 7
    - Gingerbread (2.3.x) ..... 7
    - Honeycomb (3.x)..... 7

■ CONTENTS

Ice Cream Sandwich (4.0.x).....	7
Jelly Bean (4.1.x).....	8
KitKat (4.4.x).....	8
Overview .....	8
<b>■ Chapter 2: The Mobile Device and Operating System Landscape .....</b>	<b>9</b>
Competition in the Mobile Space .....	9
iOS .....	10
BlackBerry.....	11
Windows Phone .....	11
Symbian.....	12
MeeGo.....	12
Before Android.....	12
Smartphone History.....	12
The Mobile Market: Success and Failure .....	13
Motorola i1 .....	13
Droid X.....	13
BlackBerry Torch.....	14
iPhone.....	14
The Mobile Market: Trends .....	14
Location.....	14
Current Mobile Uses .....	14
Commerce .....	15
Overview .....	15
<b>■ Chapter 3: Beyond the Mobile App—A Technology Foundation ...</b>	<b>17</b>
Connected Devices.....	17
Home Computing .....	17
Automotive .....	18
Digital Entertainment.....	18



Special Requirements .....	18
Ruggedization .....	19
Medical .....	19
Virtualized .....	19
Secure Communications .....	20
The Cyber Fiber of Our Connected World .....	20
Cellular Networks .....	20
Wireless Communications .....	21
Mobile Interfaces .....	22
Touch Screens .....	22
Vibration Motors .....	23
LED Lights .....	23
Accelerometer .....	23
Tilt Sensor .....	23
Hardware Buttons .....	23
Overview .....	23
<b>■ Chapter 4: Android Development—Business Overview and Considerations .....</b>	<b>25</b>
The Android Market Share .....	25
How Android Makes Money .....	26
Why Android Is Successful .....	26
Free .....	27
Open Source .....	27
Customization .....	27
Application Base .....	27
Hardware Choices .....	27
Device Price .....	27

<b>Legacy and Future Platform Support.....</b>	<b>28</b>
Legacy Support.....	28
Future Support.....	28
Why x86 and Android Are Right for You .....	28
Cross Compatibility.....	29
Barrier to Entry .....	29
<b>Security of Android .....</b>	<b>29</b>
Application Security.....	29
Platform Security.....	30
<b>Licensing .....</b>	<b>30</b>
Android Licensing Cost.....	31
Application Licensing Cost .....	31
<b>Physical Development Costs .....</b>	<b>31</b>
Software Development Systems.....	31
Android Testing Systems .....	32
<b>Overview .....</b>	<b>32</b>
<b>■ Chapter 5: The Intel Mobile Processor .....</b>	<b>33</b>
<b>Intel's x86 Line .....</b>	<b>33</b>
History .....	34
Strengths and Weaknesses .....	34
Business Model .....	35
<b>Clash of the Mobile Titans: ARM versus Intel .....</b>	<b>35</b>
ARM .....	35
<b>Intel's Atom Line of Microprocessors .....</b>	<b>38</b>
Intel Atom Evolution.....	38
Intel Atom Security .....	39
Intel Atom Features .....	39
Android and the Atom.....	39

Inside the Medfield System-on-Chip .....	40
Zooming In on the Saltwell CPU Architecture .....	41
<b>Architecture Differences between Intel’s Saltwell and ARM’s Cortex A15 .....</b>	<b>41</b>
Architecture .....	42
Integer Pipelines .....	42
Instruction Sets .....	42
Multi-Core/Thread Support .....	43
Security Technology .....	43
<b>Intel Hyper-Threading Technology .....</b>	<b>43</b>
<b>Application Compatibility: Native Development Kit and Binary Translator .....</b>	<b>44</b>
Overview .....	46
<b>■ Chapter 6: Installing the Android SDK for Intel Application Development .....</b>	<b>47</b>
Preparing for the SDK Installation .....	47
Supported Operating Systems .....	47
Hardware Requirements .....	48
Installing the JDK .....	48
Installing Eclipse .....	48
Installing Apache Ant (Optional) .....	48
Downloading the SDK Starter Package and Adding SDK Components .....	48
Setting Up Eclipse to work with the SDK .....	50
Overview of Android Virtual Device Emulation .....	53
Which Emulator Should You Use .....	54
Why Use the Emulator .....	54

Building an Emulator Image .....	55
Setting Up the SDK to Use x86 Emulator Images .....	57
<b>Key Gingerbread Features.....</b>	<b>62</b>
Battery Usage Stats.....	63
Task Manager .....	63
Cut and Paste Text .....	64
<b>Ice Cream Sandwich Emulation .....</b>	<b>65</b>
Prerequisites.....	65
Downloading Through the Android SDK Manager.....	65
Using the System Image.....	66
Downloading Manually .....	67
CPU Acceleration .....	70
GPU Acceleration .....	70
<b>Overview .....</b>	<b>74</b>
<b>■ Chapter 7: Creating and Porting NDK-Based Android Applications .....</b>	<b>75</b>
<b>JNI and NDK Introduction .....</b>	<b>75</b>
JNI Introduction .....	75
Java Methods and Their Corresponding Relationship with the C Function Prototype Java.....	78
Java and C Data Type Mapping.....	79
NDK Introduction .....	81
<b>NDK Installation.....</b>	<b>85</b>
Android NDK Installation.....	86
Install Cygwin .....	87
Install CDT .....	100

NDK Examples .....	105
Using the Command-Line Method to Generate a Library File .....	105
Generating a Library File in the IDE .....	114
Workflow Analysis for NDK Application Development .....	120
NDK Compiler Optimization .....	122
Machine-Independent Compiler Switch Options .....	123
Intel Processor-Related Compiler Switch Options .....	125
Overview .....	130
<b>■ Chapter 8: Debugging Android .....</b>	<b>131</b>
Prerequisites .....	131
Intel USB Driver for Android Devices .....	131
Installing the Intel Atom x86 System Image for Android Emulator .....	133
Application Debugging Using the Android Debug Bridge .....	139
Setting Up ADB .....	140
ADB on Windows .....	141
ADB Host-Client Communication .....	141
Starting ADB .....	142
Key ADB Device Commands .....	142
Using the Android Debug Tools Plug-in for Eclipse .....	143
Intel Hardware Accelerated Execution Manager .....	146
KVM Installation .....	147
Using a 64-Bit Kernel .....	148
Install KVM .....	149
Starting the Android Virtual Device .....	150
Using AVD Manager in Eclipse to Launch a Virtual Device .....	151
Running Android Within Oracle VirtualBox .....	151
Google x86 VirtualBox Build Targets for Android 4.x .....	152
Building a Custom Kernel with Mouse Support .....	153

Build the VirtualBox Disk and Android Installer .....	156
Using an Android Installer Disk to Create a Large Virtual Partition .....	157
Serial Port.....	158
Ethernet.....	159
Debugging with GDB, the GNU Project Debugger .....	160
The Intel Graphics Performance Analyzer (Intel GPA).....	170
System Debug of Android OS Running on an Intel Atom Processor ....	173
JTAG Debugging .....	174
Android OS Debugging,.....	175
Device Driver Debugging.....	176
Hardware Breakpoints.....	177
Cross-Debug: Intel Atom Processor and ARM Architecture .....	178
Variable Length Instructions.....	178
Hardware Interrupts .....	179
Single Step .....	180
Virtual Memory Mapping .....	181
Considerations for Intel Hyper-Threading Technology.....	182
SoC and Interaction of Heterogeneous Multi-Core.....	183
SVEN (System Visible Event Nexus).....	183
Signal Encode/Decode Debug .....	184
SVEN Benefits.....	184
Overview .....	184
<b>■ Chapter 9: Performance Optimizations for Android Applications on x86.....</b>	<b>185</b>
Basic Concepts of Performance Optimization.....	186
Selection of a Faster Instruction.....	186
Improve the Degree of Parallelism .....	187
Effective Use of the Register Cache .....	187

Methodology of Performance Optimizations .....	188
Performance Optimization Approaches .....	188
Performance Optimizations Automatically Done by a Compiler .....	189
Performance Optimizations Assisted by Development Tools .....	189
Performance Optimizations Done Manually .....	190
Performance Tuning with Intel VTune .....	192
System Tuning .....	198
Tuning Based on the Microarchitecture of the Processor.....	198
Intel Graphics Performance Analyzers .....	200
Introduction .....	200
Installation .....	203
Sample Usage of Intel GPA on Android .....	208
Android Multithreaded Design.....	219
Android Framework or a Thread .....	220
Java Thread Programming Interface .....	220
Threaded Programming Extensions and Support.....	223
Thread Example.....	226
Thread Synchronization.....	233
Thread Communication .....	236
Principles of Multithreaded Optimization for Intel Atom Processors .....	238
Case Study: Intel GPA–Assisted Multithreaded Optimization for an Android Application .....	239
Original Application and Intel GPA Analysis .....	240
Optimized Application and Intel GPA Analysis.....	249
Overview .....	257
<b>■ Chapter 10: x86 NDK and C/C++ Optimizations .....</b>	<b>259</b>
Vectorization.....	259
Vectorization Report .....	260
Pragmas .....	262

Auto-Vectorization and Limits .....	263
Interprocedural Optimizations .....	264
Optimization with Intel IPP .....	265
NDK Integrated Optimization Examples.....	266
C/C++: The Original Application Acceleration.....	267
Step 1: Create a New Android Application Project.....	268
Step 2: Write the C Implementation Code of the cCodeTask Function.....	273
Compiler Optimization Extension Application .....	275
Step 1: Modify the Android Part of the Application .....	276
Step 2: Modify the Makefile File of mycomputetask.c and Rebuild the Library Files.....	279
Step 2: Write the C Implementation Code for the anotherCCodeTask Function .....	279
Multiple Situations Comparison of Compiler Optimization Extensions.....	281
Example: Compare the Optimization Results by Using SSE Instructions.....	281
Overview .....	283
<b>■ Chapter 11: Using Intel Hardware Accelerated Execution Manager on Windows, Mac OS, and Linux to Speed Up Android on x86 Emulation .....</b>	<b>285</b>
Introduction .....	285
Downloading Intel HAXM.....	286
Downloading Through Android SDK Manager .....	286
Downloading Manually .....	288
Installing Intel HAXM on Windows.....	288
Adjusting Intel HAXM Memory Allocation .....	292
Intel Virtualization Technology (Intel VT-x) Capability .....	292
Intel VT-x Is Not Supported .....	292
Intel VT-x Is Not Enabled.....	292



Tips and Tricks.....	292
Mac OS.....	293
Adjusting Intel HAXM Memory Allocation .....	296
Removing Intel HAXM.....	296
Troubleshooting.....	297
Intel Execute Disable (XD) Bit Capability Error .....	297
Intel XD Is Not Supported.....	297
Intel XD Is Not Enabled .....	297
Intel Virtualization Technology (VT-x) Capability .....	297
Intel VT-x Is Not Supported .....	298
Intel VT-x Is Not Enabled.....	298
Tips and Tricks.....	298
Linux.....	299
KVM Installation .....	299
Install KVM.....	300
Start the AVD from Android SDK Directly from Terminal.....	301
Start the AVD by AVD Manager in Eclipse.....	302
Overview .....	302
<b>■ Chapter 12: Performance Testing and Profiling Apps with Platform Tuning .....</b>	<b>303</b>
Start with Your First x86 Full Format Video Player .....	303
Compile x86 FFmpeg: Cross-Compile.....	304
Compile x86 FFmpeg: Android.mk.....	305
How to Determine CPU Usage and Find Hotspots .....	306
Show CPU Usage Dynamically Onscreen.....	306
Get Function Running Time.....	307

<b>Use Yasm to Get the Best-Performing x86 Library .....</b>	<b>307</b>
How to Use Yasm .....	308
The Result of Using Yasm .....	308
<b>Use SSE (Intel’s Streaming SIMD Extensions) to Optimize Color Space Transformation .....</b>	<b>309</b>
What Is SIMD? .....	310
How SIMD Works .....	310
Implement NV21-RGB SSE Code.....	312
<b>How to Display an Image Using the Android 4.0 NDK .....</b>	<b>314</b>
<b>The Common Cross-Compile Script .....</b>	<b>314</b>
<b>Testing and Profiling with Hardware Acceleration .....</b>	<b>315</b>
<b>Using the Integration Layer (IL) for Hardware Encoding.....</b>	<b>316</b>
How to Get the OMX-IL Interface on Android for Intel Architecture .....	317
How Does the OMX-IL Work? .....	317
Demo: Special Effects Video Recorder.....	319
Packaging a Hardware Video Encoder Library.....	320
Implement Camera Preview .....	320
Profiling Java Code with Traceview .....	321
Start a Recording Thread.....	322
Adding Special Effects.....	323
<b>Use OpenMAX AL on Android 4.0.....</b>	<b>325</b>
The Advantage of Using Native Multimedia API (OpenMAX AL) .....	325
Demo: Streaming Media Player .....	326
<b>Use a Powerful Media API: MediaCodec on Android 4.1.....</b>	<b>326</b>
Sample Code: Audio Decoder .....	327
Use MediaCodec in NDK.....	328
<b>Overview .....</b>	<b>328</b>

**■ Appendix A: References..... 331**

**Chapter 1: History and Evolution of the Android OS ..... 331**

    Origins ..... 331

    First Android Distribution in 2007 ..... 331

    What is Android? ..... 331

    The Open Handset Alliance (OHA)..... 332

    The Android Open Source Project (AOSP) ..... 332

    Android Versions..... 332

**Chapter 2: The Mobile Device and Operating System Landscape ..... 332**

    Competition in Mobile Space..... 332

    The Mobile Market..... 333

    The Mobile Market: Trends ..... 335

**Chapter 3: Beyond the Mobile App—A Technology Foundation ..... 335**

    Connected Devices ..... 335

    Special Requirements..... 336

    The Cyber-Fiber of our Connected World..... 336

    Mobile Interfaces..... 337

**Chapter 4: Android Development—Business Overview and Considerations ..... 337**

    Market Share ..... 337

    Security ..... 338

    Licensing ..... 338

**Chapter 5: The Intel Mobile Processor..... 339**

    Clash of the Mobile Titans: ARM versus Intel..... 339

    Intel..... 339

    Android Atom Platforms..... 339

**Chapter 6: Installing the Android SDK for Intel Application Development ..... 340**  
    Installation and Setup..... 340  
    Emulation..... 340

**Chapter 7: Creating and Porting NDK-Based Android Applications ..... 341**

**Chapter 8: Debugging Android..... 341**

**Chapter 9: Performance Optimizations for Android Applications on x86 ..... 341**

**Chapter 10: x86 NDK and C/C++ Optimizations ..... 342**

**Chapter 11: Using Intel Hardware Accelerated Execution Manager on Windows, Mac OS, and Linux to Speed Up Android on x86 Emulation ..... 342**

**Chapter 12: Performance Testing and Profiling Apps with Platform Tuning ..... 342**

**Index..... 343**

# About the Authors

**Iggy Krajci** earned his Bachelor of Science degree in computer science from the University of Advancing Technology in Tempe, Arizona. He currently is employed as a software engineer at Go Daddy in North Scottsdale, AZ, where he works on Java development. Previously, Krajci worked with SAIFE Inc. on the Android platform and its incorporation in the x86 ecosystem. In his free time, Krajci supports the open source and information technology communities by attending related conferences and releasing open source projects.

**Darren Cummings** is the CEO and founder of Cummings Engineering and SAIFE® Inc., with a mission to seamlessly and pervasively protect and connect the world's data, bringing security to any device, across any network, for any mission. His companies leverage their Secure Agile Interoperable Framework for the Enterprise (SAIFE) security-as-a-service to provide holistic solutions for secure communications, software and systems architecture, defense projects, secure wireless, and real-time embedded systems. Cummings holds undergraduate degrees in electrical engineering and math from Iowa State University and a Master's degree in software engineering from Walden University. Before founding SAIFE Inc., Cummings served in senior systems and software engineering roles with ViaSat, General Dynamics and the Motorola Government Group where he gained substantial telecommunications and real-time embedded project experience.



# Acknowledgments

I would like to thank Sarah Yost, Jeffrey Pepper, and Javier Leija for their help with this book! I'd also like to thank Sarah's team at SAIFE Inc.: Jen Haldaman, Tayler "Pickles" Nichols, and John Curtis.

—Iggy

Thank you to my team at SAIFE Inc. Your support during the writing process has been invaluable - this book would not be a reality without you! Many thanks to Sarah Yost, for being the most dedicated editor, champion, and author-wrangler ever.

We would both like to acknowledge the Intel® Industry Education division for assisting with content architecture for this book project. Some material was provided by Intel and is used with permission. We would like to thank Ryan Cohen, Stephen Farmer, Eng Kean Lee, Robert Mueller, Dmitry Shkurko, Tao Wang, Keng Lai Yap, and Li Yuming in particular for their contributions!

—Darren





# Introduction

We wrote *Android on x86: an Introduction to Optimizing for Intel® Architecture* to provide a one-stop, detailed resource for the topic's best practices and procedures. The book encompasses the installation issues, hardware optimization issues, software requirements, programming tasks, and performance optimizations that emerge when you consider programming for x86-based Android devices. Having worked on related projects ourselves, we committed to collecting our experience and information into one book which could be used as a guide through any project's specific requirements. We dove into fine-tuned optimizations, native code adjustments, hardware acceleration, and advanced profiling of multimedia applications.

The book is not dedicated solely to code, although you'll find plenty of code samples and case studies inside. Instead, we've filled *Android on x86* with the information you need in order to take advantage of the x86 architectures. We will guide you through installing the Android Software Development Kit for Intel Architectures, help you understand the differences and similarities between the processors available for commercial Android devices, teach you to create and port applications, debug existing x86 applications, offer solutions for NDK and C++ optimizations, and introduce the Intel Hardware Accelerated Execution Manager. The information we've pulled together provides the most useful help for getting your development job done quickly and well.

## Why Android on x86?

In 2011, we experienced a paradigm shift in how we communicate. Smart device sales outpaced personal computer sales for the first time. This changing of the guard emerged from three sources:

- Our increasing professional and social need for open, constant communication
- The lower cost and compelling new features of smartphones and tablets
- The increased ease of use and availability of mobile apps

In the next few years, mobile access to the Internet is likely to exceed access via laptops and desktops; the hardware we use to communicate may change, but our passion for connectivity anytime, anywhere is sure to continue.

Holding more than 80% of the market share for smartphone shipments worldwide, Google's Android operating system has proven to be the leader of this mobile revolution. The key reasons for the success of Android are its open platform and flexible partnerships. The wealth of open-source resources available for Android developers spurs the creation of more apps, giving consumers more choices. In addition, the open platform supports a competitive and diverse hardware environment.

As the market for high-performing mobile devices widens, Google has teamed up with Intel to envision the next frontier for Android: getting the OS to run on devices with Intel architectures inside. The journey towards Android on Intel architectures began unofficially in 2009, when a group of developers started the open source Android-x86 initiative in order to port Android onto devices running on Intel x86 processors. Soon after, with the official Android on Intel architecture project, Intel started contributing code and resources to the Android Open Source Project (AOSP). In 2012, the first Android smartphones featuring Intel processors were released to market worldwide; by late 2013, Android smartphones and tablets with unprecedented processing power were entering United States' markets. Most recently, the two groups committed to getting Android to run on 64-bit devices, including netbooks, laptops, and traditional desktop PCs, meaning that in 2014, Android will break into a market historically dominated by Microsoft Windows and Apple OSX. Android will bring its enormous, thriving community of application developers forward to a wide range of devices and hardware architectures.

The collaboration brings a number of benefits from both groups. Intel's x86 architecture comes with 35 years of well-documented processing excellence, a mature developer ecosystem, and a sophisticated set of development tools. In terms of performance, Intel's latest chips strike a balance between high performance and low power consumption that is ideal for smartphones, tablets, and netbooks. Native x86 emulator support is a key feature of the latest Android SDK versions, and Intel is dedicated to providing developers with a host of tools for optimizing Android application performance for their chips.

By expanding onto both 32-bit and 64-bit architectures, the Android landscape is opening wide. More Android-equipped mobile devices with Intel processors are hitting shelves and our fingertips every day, and the upcoming addition of Intel-powered netbooks and laptops will shape the environment into something amazing. A new Android experience will take shape, one that remains diverse and becomes optimized for larger screens, robust multi-windowing, and ever-faster processor speeds. It's an exciting time, and we hope that developers will seize this new opportunity to expand Android's horizons.

## Who Is This Book For?

This book is aimed at two general categories of people: developers and those interested in choosing Android x86 as a platform for their applications. With this in mind, the beginning chapters focus on much more high-level, nontechnical questions, so that people from all technical backgrounds can make informed choices. The later chapters focus heavily on the developers' side of the world, starting with a basic foundation of microprocessor architectures and Android development environments and then building

to very advanced, performance-focused content. Our goal is to reach the entire spectrum of people who are interested in Android on x86, and to do our best at getting you the answers you need.

We really hope you enjoy the book. We certainly have enjoyed exploring this topic, and look forward to seeing what will happen in this rapidly-expanding field in the upcoming years. We would also like to note that while we may know a thing or two about Android, we recognize that we are certainly not the most knowledgeable about everything. Feel free to challenge any information that you find in this book - we encourage you to use outside resources and really involve yourself in the communities that surround this technology!