WIKIPEDIA

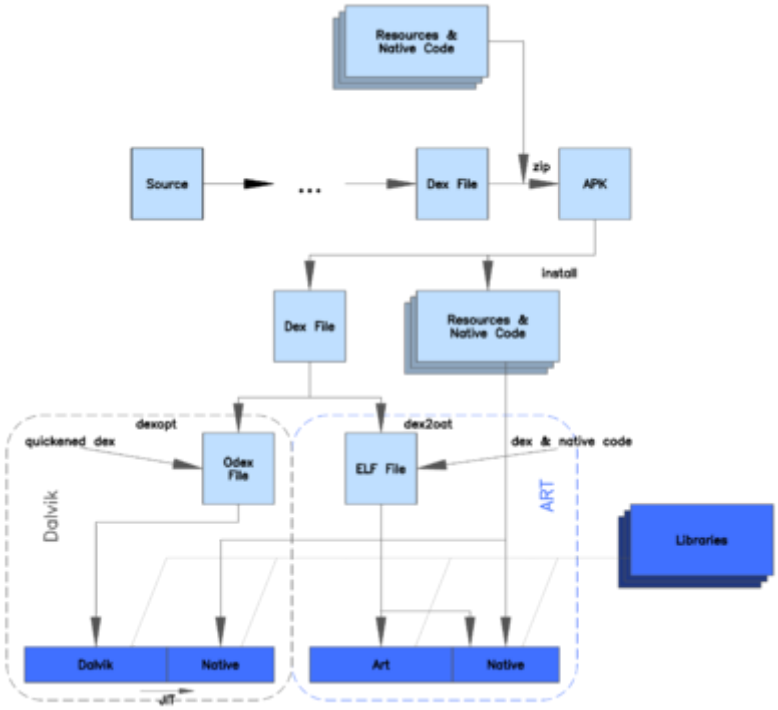
Android Runtime

Android Runtime (**ART**) is an application runtime environment used by the Android operating system. Replacing Dalvik, the process virtual machine originally used by Android, ART performs the translation of the application's bytecode into native instructions that are later executed by the device's runtime environment.[1]

Android Runtime

Website	source.android.com/devices/tech/dalvik/art.html (https://source.android.com/devices/tech/dalvik/art.html)

Android 2.2 "Froyo" brought trace-based just-in-time (JIT) compilation into Dalvik, optimizing the execution of applications by continually profiling applications each time they run and dynamically compiling frequently executed short segments of their bytecode into native machine code. While Dalvik interprets the rest of application's bytecode, native execution of those short bytecode segments, called "traces", provides significant performance improvements.[2][3]



A comparison of Dalvik and ART architectures

Unlike Dalvik, ART introduces the use of ahead-of-time (AOT) compilation by compiling entire applications into native machine code upon their installation. By eliminating Dalvik's interpretation and trace-based JIT compilation, ART improves the overall execution efficiency and reduces power consumption, which results in improved battery autonomy on mobile devices. At the same time, ART brings faster execution of applications, improved memory allocation and garbage collection (GC) mechanisms, new applications debugging features, and more accurate high-level profiling of applications.[1][4][5]

To maintain backward compatibility, ART uses the same input bytecode as Dalvik, supplied through standard .dex files as part of APK files, while the .odex files are replaced with Executable and Linkable Format (ELF) executables. Once an application is compiled by using ART's on-device `dex2oat` utility, it is run solely from the compiled ELF executable; as a result, ART eliminates various application execution overheads associated with Dalvik's interpretation and trace-based JIT compilation. As a downside, ART requires additional time for the compilation when an application is installed, and applications take up slightly larger amounts of secondary storage (which is usually flash memory) to store the compiled code.[1][4][5]

Android 4.4 KitKat brought a technology preview of ART, including it as an alternative runtime environment and keeping Dalvik as the default virtual machine.[6][7] In the subsequent major Android release, Android 5.0 Lollipop, Dalvik was entirely replaced by ART.[8]

Android 7.0 Nougat introduced JIT compiler with code profiling to ART, which lets it constantly improve the performance of Android apps as they run. The JIT compiler complements ART's current Ahead of Time compiler and helps improve runtime performance.[9]

See also

- Android software development – various concepts and software development utilities used for the creation of Android applications
- Android version history – a history and descriptions of Android releases, listed primarily by their official API levels
- Comparison of application virtualization software – various portable and scripting language virtual machines
- Virtual machine – an emulation of a particular computer system, with different degrees of implemented functionality

References

1. Andrei Frumusanu (July 1, 2014). "A Closer Look at Android RunTime (ART) in Android L" (http://anandtech.com/show/8231/a-closer-look-at-android-runtime-art-in-android-l/). AnandTech. Retrieved July 5, 2014.
2. Ben Cheng; Bill Buzbee (May 2010). "A JIT Compiler for Android's Dalvik VM" (http://www.android-app-developer.co.uk/android-app-development-docs/android-jit-compiler-androids-dalvik-vm.pdf) (PDF). *android-app-developer.co.uk*. Google. pp. 5–14. Retrieved March 18, 2015.
3. Phil Nickinson (May 26, 2010). "Google Android developer explains more about Dalvik and the JIT in Froyo" (http://www.androidcentral.com/google-android-developer-explains-more-about-dalvik-and-jit-froyo). *androidcentral.com*. Retrieved July 8, 2014.
4. "Android Developers: ART and Dalvik" (https://source.android.com/devices/tech/dalvik/index.html). *source.android.com*. March 9, 2015. Retrieved March 18, 2015.
5. "Android Developers: Configuring ART – How ART works" (https://source.android.com/devices/tech/dalvik/configure.html#how_art_works). *source.android.com*. March 9, 2015. Retrieved March 18, 2015.
6. Sean Buckley (November 6, 2013). "'ART' experiment in Android KitKat improves battery life and speeds up apps" (https://www.engadget.com/2013/11/06/new-android-runtime-could-improve-battery-life/). Engadget. Retrieved July 5, 2014.
7. Daniel P. (November 7, 2013). "Experimental Google ART runtime in Android KitKat can bring twice faster app executions" (http://www.phonearena.com/news/Experimental-Google-ART-runtime-in-Android-KitKat-can-bring-twice-faster-app-executions_id49139). *phonearena.com*. Retrieved July 5, 2014.
8. Brad Linder (October 15, 2014). "What's new in Android 5.0 Lollipop?" (http://liliputing.com/2014/10/whats-new-android-5-0-lollipop.html). *liliputing.com*. Retrieved October 15, 2014.
9. "Implementing ART Just-In-Time (JIT) Compiler" (https://source.android.com/devices/tech/dalvik/jit-compiler.html). *source.android.com*. January 22, 2017. Retrieved January 22, 2017.

External links

- Official website (https://source.android.com/devices/tech/dalvik/index.html)
- Android Basics 101: Understanding ART, the Android Runtime (https://www.youtube.com/watch?v=USgXkI-NRPo) on YouTube, XDA Developers, February 12, 2014
- ART: Android's Runtime Evolved (https://www.youtube.com/watch?v=EBlTzQsUoOw) on YouTube, Google I/O 2014, by Anwar Ghuloum, Brian Carlstrom and Ian Rogers
- A JIT Compiler for Android's Dalvik VM (https://www.youtube.com/watch?v=Ls0tM-c4Vfo) on YouTube, Google I/O 2010, by Ben Cheng and Bill Buzbee
- Delivering Highly Optimized Android Runtime (ART) and Web Runtime on Intel Architecture (https://intel.lanyonevents.com/sz15/connect/fileDownload/session/4BBA45B063EB52C028E40C64F4270EC3/SZ15_STTS004_100_ENGf.pdf), Intel, August 4, 2015, by Haitao Feng and Jonathan Ding
- Android 7.1 for Developers: Profile-guided JIT/AOT compilation (https://developer.android.com/preview/api-overview.html#jit_aot), Android Developers, describes ART changes in Android 7