Click to verify

Dalvik Virtual Machine Register Based VM The language is designed by Dan Bornstein, some cool features and nature of Apple Swift iOS Programming Language he gave it a try with the initial compiler implementations examples contributed from other Google engineers at its Android mobile phone platform. The Dalvik virtual machine was named in that way after Bornstein, a resident of the nearby village "Dalvík", born on ''Eyjafjörður''.What is Dalvik Virtual Machine?Dalvik Virtual Machine (DVM) is the custom program introduced for Android apps. It takes the Java code and creates an optimized version of it in a file with .dex(extension) which is known as Dalvik executable. This format allows the apps to run quickly with fewer resources, i.e. on mobile phones and low-memory, slower devices. This is different from a typical Java Virtual Machine (JVM), as it optimized especially for Android to run apps that use less memory and are compatible with the millions of devices running on various versions.Simple Working of DVMJava Source CodeThe original development of Android apps is mostly done using Java programming language that developers use to write the instructions your app executes.CompilationWe need to convert this textual Java code into form understandable by computer. This is commonly referred to as compilation. This Java bytecode is an executable version of your code.Conversion to Dalvik BytecodeThe Java bytecode is transformed into a different format known as Dalvik byte code. They use a tool called dx, to do this and make the code run smoothly on Android devices.Dalvik Virtual Machine (DVM)Last, but not least the Dalvik bytecode is run on the Dalvik Virtual Machine (DVM). DVM has the special ability to run the applications on mobile devices which have limited capabilities of memory and battery.Features of Dalvik Virtual MachineRegister-Based Architecture: Dalvik is a register-based virtual machine unlike the traditional stack based one (like JVM). This helps minimize overhead that would be incurred by constantly pushing and popping values off the stack making execution faster, especially on low memory mobile devices.Optimized for Mobile Devices: It is an optimizing BSJ virtual machine that was designed specifically to run on the Android operating system, with low memory and power requirements. It virtualizes multiple apps by running a separate instance of the VM for each application as it does best in simultaneously handling many applications.Dex Bytecode Execution: Dalvik uses. dex (Dalvik Executable) files—these compile the source into a more compact bytecode in order to make it better-suited for low-memory environments. We compile multiple Java classes into a single. dex file so that it was just 63% of the original size, meaning a small parse and load time on mobile devices.Just In Time (JIT) Compilation: Dalvik was released from Android 2.2 (Froyo) and later included a JIT compiler to translate the bytecode into machine code directly at execution runtime, which could eventually boost frequently executed codes performance.Garbage Collection: The above code ran on the DALVIK which has automatic garbage collection to make sure that apps will not consume large amount of resources and thus there is no effect in system stability.Support for Multithreading: Dalvik is multithreading mobile machine can have multiple threads at the same time for an application to work on. This feature is one of the most required things when it comes to mobile apps e.g., background process along with foreground user tracking.Advantages of DVM(Dalvik Virtual Machine)DVM supports the Android operating system only.In DVM executable is APK.Execution is faster.From Android 2.2 SDK Dalvik has it's own JIT (Just In Time) compiler.Applications are given their own instances.Disadvantages of DVM(Dalvik Virtual Machine)DVM supports only Android Operating System.For DVM very few Re-Tools are available.App Installation takes more time due to dex.More internal storage is required.Application of Dalvik Virtual MachineRunning Android Apps: It improves how apps run by using processing power and less memory, ensuring smooth work.Support for Multitasking: It allows multiple android apps to run at the same time without interrupting each other sped , making multi tasking smoother on mobile devices. Memory Management: It manages memory more efficiently than older system, making sure that apps can run well even on devices with limited memory.ConclusionThe Dalvik Virtual Machine was essential for the early Android phones. DVM is very handy to build an android apps because it could be optimized for mobile, take care of low-memory & handling the multiple app. In more recent Android versions, DVM has been replaced by ART but it paved the road on how apps get executed in an Android system. 838bd34 Merge "[owners] Remove gavra@google.com from OWNERS" into main am: 237fe9f1e4 by Owner Cleanup Bot · 3 months ago main master237fe9f Merge "[owners] Remove gavra@google.com from OWNERS" into main by Owner Cleanup Bot · 3 months ago1af8ede [owners] Remove gavra@google.com from OWNERS by Owner Cleanup Bot · 3 months ago055d459 Merge "Support v41 DEX version for dexdeps." into main am: 1bed4bab8b am: e5ed58210b am: f075e91770 by Keyi Gui · 1 year, 5 months ago android15-automotiveos-dev android15-automotiveos-release android15-d4-release android15-platform-release android15-prebuilt-test android15-qpr1-release android15-qpr1-s3-release android15-qpr1-s4-release android15-qpr1-s5-release android15-qpr2-release android15-qpr2-s1-release android15-qpr2-s10-release android15-qpr2-s2-release android15-qpr2-s3-release android15-qpr2-s4-release android15-qpr2-s5-release android15-qpr2-s6-release android15-qpr2-s7-release android15-qpr2-s8-release android15-qpr2-s9-release android15-release android15-s1-release android15-security-release android15-tests-dev android15-tests-release aml_tz6_351010000 aml_tz6_351400020 android-15.0.0_r1 android-15.0.0_r10 android-15.0.0_r11 android-15.0.0_r12 android-15.0.0_r13 android-15.0.0_r14 android-15.0.0_r15 android-15.0.0_r16 android-15.0.0_r17 android-15.0.0_r2 android-15.0.0_r20 android-15.0.0_r21 android-15.0.0_r22 android-15.0.0_r23 android-15.0.0_r24 android-15.0.0_r25 android-15.0.0_r26 android-15.0.0_r27 android-15.0.0_r28 android-15.0.0_r29 android-15.0.0_r3 android-15.0.0_r30 android-15.0.0_r31 android-15.0.0_r32 android-15.0.0_r33 android-15.0.0_r34 android-15.0.0_r35 android-15.0.0_r36 android-15.0.0_r4 android-15.0.0_r5 android-15.0.0_r6 android-15.0.0_r7 android-15.0.0_r8 android-15.0.0_r9 android-automotiveos-15.0.0_lts1 android-cts-15.0_r1 android-cts-15.0_r2 android-cts-15.0_r3 android-cts-15.0_r4 android-platform-15.0.0_r1 android-platform-15.0.0_r2 android-platform-15.0.0_r3 android-platform-15.0.0_r4 android-platform-15.0.0_r5 android-platform-15.0.0_r6 android-platform-15.0.0_r7 android-platform-15.0.0_r8 android-security-15.0.0_r1 android-security-15.0.0_r2 android-security-15.0.0_r3 android-security-15.0.0_r4 android-security-15.0.0_r5 android-security-15.0.0_r6 android-security-15.0.0_r7 android-security-15.0.0_r8 android-vts-15.0_r1 android-vts-15.0_r2 android-vts-15.0_r3 android-vts-15.0_r4f075e91 Merge "Support v41 DEX version for dexdeps." into main am: 1bed4bab8b am: e5ed58210b by Keyi Gui · 1 year, 5 months agoe5ed582 Merge "Support v41 DEX version for dexdeps." into main am: 1bed4bab8b by Keyi Gui · 1 year, 5 months ago1bed4ba Merge "Support v41 DEX version for dexdeps." into main by Keyi Gui · 1 year, 5 months agod7e9884 Support v41 DEX version for dexdeps. by kgui · 1 year, 5 months ago504be36 Merge "Update `OWNERS` files under `dalvik`." into main am: bd9032155f am: 5a6c67fad8 am: 467e199ca7 by Roland Levillain · 1 year, 5 months ago467e199 Merge "Update `OWNERS` files under `dalvik`." into main am: bd9032155f am: 5a6c67fad8 by Roland Levillain · 1 year, 5 months ago5a6c67f Merge "Update `OWNERS` files under `dalvik`." into main am: bd9032155f by Roland Levillain · 1 year, 5 months agobd90321 Merge "Update `OWNERS` files under `dalvik`." into main by Roland Levillain · 1 year, 5 months agodec6655 Update `OWNERS` files under `dalvik`. by Roland Levillain · 1 year, 5 months ago916766e [automerger skipped] Remove unused dx-doc-stubs am: f7a6e166d2 -s ours by Paul Duffin · 1 year, 9 months agof7a6e16 Remove unused dx-doc-stubs by Paul Duffin · 1 year, 9 months agoe3c574c Merge "Remove references to the pdk product variable" into main am: e6ca015645 am: bcd1192abc am: e13f95ec26 am: dfa1dd31a6 am: f001dfd9f3 by Treehugger Robot · 1 year, 10 months agof001dfd Merge "Remove references to the pdk product variable" into main am: e6ca015645 am: bcd1192abc am: e13f95ec26 am: dfa1dd31a6 by Treehugger Robot · 1 year, 10 months agodfa1dd3 Merge "Remove references to the pdk product variable" into main am: e6ca015645 am: bcd1192abc am: e13f95ec26 by Treehugger Robot · 1 year, 10 months agoe13f95e Merge "Remove references to the pdk product variable" into main am: e6ca015645 am: bcd1192abc by Treehugger Robot · 1 year, 10 months agobcd1192 Merge "Remove references to the pdk product variable" into main am: e6ca015645 by Treehugger Robot · 1 year, 10 months ago Android runtime (ART) is the managed runtime used by applications and some system services on Android. ART and its predecessor Dalvik were originally created specifically for the Android project. ART and Dalvik are compatible runtimes running Dex bytecode, so apps developed for Dalvik should work when running with ART. Dalvik Dalvik Virtual Machine or DVM is a Register-Based virtual machine that was designed and written by Dan Bornstein. The Dalvik virtual machine was named by Bornstein after the fishing village “Dalvík” in Eyjafjörður, Iceland, where some of his ancestors used to live. Dalvik is a discontinued process virtual machine (VM) in the Android OS that executes applications written for Android. Dalvik bytecode format is still used as a distribution format, but no longer at runtime in newer Android versions. Android itself is a Linux system with Dalvik sitting on top of it. DVM takes android app, turns them from java code into bytecode that the Linux system can run. Basically, a plain java code is compiled in the JIT compiler to bytecode during runtime to run on the machine. This could lead to a slowdown because compilation at runtime especially during runtime is time-consuming. Hence, manufacturers and OEMs sometimes put out their applications as odexed. There are 2 types of files: .dex(Dalvik Executable file) file is an android's compiled code file. These .dex files are then zipped into a single .apk file..odex file is created by the Android operating system to save space and increase the boot speed of an Android app (a .apk file). dexopt is used to optimize DEX to ODEX (optimized DEX)which contains the optimized bytecode. So the whole process in DVM can be summarized as: JAVA source code(.java) --> Bytecode(.dex) --> DVM DVM is better for low storage devices. But it is slower as compilation is done after installation. ART With a newer android version specifically from 4.4 version KitKat, there is the concept of ART as an alternative to DVM. ART(Android Run Time) is a successor of DVM which uses the same bytecode and .dex files (but not .odex files), with the succession aiming at performance improvements transparent to the end-users. Android 5.0 “Lollipop” is the first version in which ART is the only included runtime. Now the thing that ART does is bring apps that are fully compiled when they're installed on the device. Therefore, higher performance as no need to convert code to bytecode then compile. But the downside is you need more storage space and a little longer to install because of compilation during installation meaning it has to live on the device all the time. Hence, instead of relatively small java code, we have larger bytecode/machine code. You may have heard the terms odexed and de-odexed. What is done in this instance is you take a small part of the application and then precompile it they can go ahead and make a portion of their application optimized to run on their device, and so they've now precompiled that section of the app and the rest of its compiled at runtime. So this makes it just a little faster and more performant than in Dalvik. But this approach takes a little more storage space. Ex: Samsung with TouchWiz .. A lot of stuff, TouchWiz is based on is, precompiled and hence when these are de-odexed, you can retheme/reskin them while losing some performance benefits. Before reading ahead, remember that the dex2oat is used to optimize and compile .dex into a .oat file which may contain machine code in the ELF format. ART compiles apps using the on-device dex2oat tool. This utility accepts DEX files as input and generates a compiled app executable for the target device. When an app is installed, Android automatically optimizes app data and creates a corresponding OAT file. An OAT file is created by the Android operating system in order to speed up the loading time of an Android app (.APK file). Android uses this file to load the app more quickly, creating a better user experience. Difference Between DVM and ARTDALVIK VIRTUAL MACHINE ANDROID RUN TIME Faster Booting timeRebooting is significantly longerCache builds up overtimeThe cache is built during the first bootOccupies less space due to JITConsumes a lot of storage space internally due to AOTWorks best for small storage devicesWorks best for Large storage devicesStable and tested virtual machineExperimental and new - not much app support comparativelyLonger app loading timeExtremely Faster and smoother Faster and app loading time and lower processor usageUses JIT compiler(JIT: Just-In-Time)Thereby resulting in lower storage space consumptionUses AOT compiler(Ahead-Of-Time) thereby compiling apps when installedApplication lagging due to garbage collector pauses and JITReduced application lagging and better user experienceApp installation time is comparatively lower as the compilation is performed laterApp installation time is longer as compilation is done during installationDVM converts bytecode every time you launch a specific app. ART converts it just once at the time of app installation. That makes CPU execution easier. Improved battery life due to faster execution. It is slower than ART.It is faster.It does not provide optimized battery life as it consumes more power.It provides optimized battery performance as it consumes less power.While considering Booting, then this device is fast.It lags in term of booting. Conclusion: Which one is better It depends on the device and one's own device and preferences: in the long run, ART is better, but the apps do get bigger, requiring large storage space over time, unlike flappy bird which was just 1MB and required less space. Therefore, the devices in the future need to expand storage if ART is going to dominate the market in the next few years. In Android Lolipop, DVM has been replaced by ART. Because DVM converts bytecode every time you launch a specific app. But ART converts it just once during the installation of the app. That makes CPU execution easier. ARM is an architecture. Like MIPS, x86, etc. DVM/ART runs on top of ARM. Both of them can not replace each other. Dalvik Virtual Machine Register Based VM The language is designed by Dan Bornstein, some cool features and nature of Apple Swift iOS Programming Language he gave it a try with the initial compiler implementations examples contributed from other Google engineers at its Android mobile phone platform. The Dalvik virtual machine was named in that way after Bornstein, a resident of the nearby village "Dalvík", born on ''Eyjafjörður''.What is Dalvik Virtual Machine?Dalvik Virtual Machine (DVM) is the custom program introduced for Android apps. It takes the Java code and creates an optimized version of it in a file with .dex(extension) which is known as Dalvik executable. This format allows the apps to run quickly with fewer resources, i.e. on mobile phones and low-memory, slower devices. This is different from a typical Java Virtual Machine (JVM), as it optimized especially for Android to run apps that use less memory and are compatible with the millions of devices running on various versions.Simple Working of DVMJava Source CodeThe original development of Android apps is mostly done using Java programming language that developers use to write the instructions your app executes.CompilationWe need to convert this textual Java code into form understandable by computer. This is commonly referred to as compilation. This Java bytecode is an executable version of your code.Conversion to Dalvik BytecodeThe Java bytecode is transformed into a different format known as Dalvik byte code. They use a tool called dx, to do this and make the code run smoothly on Android devices.Dalvik Virtual Machine (DVM)Last, but not least the Dalvik bytecode is run on the Dalvik Virtual Machine (DVM). DVM has the special ability to run the applications on mobile devices which have limited capabilities of memory and battery.Features of Dalvik Virtual MachineRegister-Based Architecture: Dalvik is a register-based virtual machine unlike the traditional stack based one (like JVM). This helps minimize overhead that would be incurred by constantly pushing and popping values off the stack making execution faster, especially on low memory mobile devices.Optimized for Mobile Devices: It is an optimizing BSJ virtual machine that was designed specifically to run on the Android operating system, with low memory and power requirements. It virtualizes multiple apps by running a separate instance of the VM for each application as it does best in simultaneously handling many applications.Dex Bytecode Execution: Dalvik uses. dex (Dalvik Executable) files—these compile the source into a more compact bytecode in order to make it better-suited for low-memory environments. We compile multiple Java classes into a single. dex file so that it was just 63% of the original size, meaning a small parse and load time on mobile devices.Just In Time (JIT) Compilation: Dalvik was released from Android 2.2 (Froyo) and later included a JIT compiler to translate the bytecode into machine code directly at execution runtime, which could eventually boost frequently executed codes performance.Garbage Collection: The above code ran on the DALVIK which has automatic garbage collection to make sure that apps will not consume large amount of resources and thus there is no effect in system stability.Support for Multithreading: Dalvik is multithreading mobile machine can have multiple threads at the same time for an application to work on. This feature is one of the most required things when it comes to mobile apps e.g., background process along with foreground user tracking.Advantages of DVM(Dalvik Virtual Machine)DVM supports the Android operating system only.In DVM executable is APK.Execution is faster.From Android 2.2 SDK Dalvik has it's own JIT (Just In Time) compiler.Applications are given their own instances.Disadvantages of DVM(Dalvik Virtual Machine)DVM supports only Android Operating System.For DVM very few Re-Tools are available.App Installation takes more time due to dex.More internal storage is required.Application of Dalvik Virtual MachineRunning Android Apps: It improves how apps run by using processing power and less memory, ensuring smooth work.Support for Multitasking: It allows multiple android apps to run at the same time without interrupting each other sped , making multi tasking smoother on mobile devices. Memory Management: It manages memory more efficiently than older system, making sure that apps can run well even on devices with limited memory.ConclusionThe Dalvik Virtual Machine was essential for the early Android phones. DVM is very handy to build an android apps because it could be optimized for mobile, take care of low-memory & handling the multiple app. In more recent Android versions, DVM has been replaced by ART but it paved the road on how apps get executed in an Android system. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Dalvik Virtual Machine (DVM) is a specialized virtual machine used by the Android operating system to execute applications written in Java. It is designed to be lightweight and optimized for mobile devices with limited resources, such as battery life and memory.The DVM was named after a fishing village in Iceland, which was the hometown of the founder of the Android project, Andy Rubin. The DVM was the primary virtual machine used by Android until Android 5.0 Lollipop, which introduced the new Android Runtime (ART).Here are some key features and components of the Dalvik Virtual Machine:Bytecode format: The DVM executes code that is compiled into a specific bytecode format called “dex” (Dalvik Executable). This format is designed to be compact and optimized for the mobile environment. The dex format is created from the original Java bytecode using the dx tool, which is part of the Android SDK.JIT compilation: The DVM uses a Just-In-Time (JIT) compiler to convert the dex bytecode into machine code at runtime. This process improves the performance of the application by reducing the overhead of interpreting the bytecode. The JIT compiler is used in the Android 2.2 Froyo and later versions.Garbage collection: The DVM includes a garbage collector that automatically manages memory allocation and deallocation. This feature helps to prevent memory leaks and makes it easier for developers to write memory-efficient applications.Class loading: The DVM uses a class loader to dynamically load classes and their associated resources. This allows applications to be updated and extended without requiring the user to reinstall the entire application.Multiple virtual machines: The DVM allows multiple virtual machines to run simultaneously, which enables multiple applications to be executed concurrently. This feature allows for greater multitasking capabilities and improves the overall user experience.Overall, the Dalvik Virtual Machine played a critical role in the early development of the Android operating system. Its lightweight design and optimized performance helped to make Android a popular mobile platform, and it laid the foundation for the newer Android Runtime (ART) that replaced it in later versions of the operating system. Dalvik Virtual Machine Register Based VM The language is designed by Dan Bornstein, some cool features and nature of Apple Swift iOS Programming Language he gave it a try with the initial compiler implementations examples contributed from other Google engineers at its Android mobile phone platform. The Dalvik virtual machine was named in that way after Bornstein, a resident of the nearby village "Dalvík", born on ''Eyjafjörður''.What is Dalvik Virtual Machine?Dalvik Virtual Machine (DVM) is the custom program introduced for Android apps. It takes the Java code and creates an optimized version of it in a file with .dex(extension) which is known as Dalvik executable. This format allows the apps to run quickly with fewer resources, i.e. on mobile phones and low-memory, slower devices. This is different from a typical Java Virtual Machine (JVM), as it optimized especially for Android to run apps that use less memory and are compatible with the millions of devices running on various versions.Simple Working of DVMJava Source CodeThe original development of Android apps is mostly done using Java programming language that developers use to write the instructions your app executes.CompilationWe need to convert this textual Java code into form understandable by computer. This is commonly referred to as compilation. This Java bytecode is an executable version of your code.Conversion to Dalvik BytecodeThe Java bytecode is transformed into a different format known as Dalvik byte code. They use a tool called dx, to do this and make the code run smoothly on Android devices.Dalvik Virtual Machine (DVM)Last, but not least the Dalvik bytecode is run on the Dalvik Virtual Machine (DVM). DVM has the special ability to run the applications on mobile devices which have limited capabilities of memory and battery.Features of Dalvik Virtual MachineRegister-Based Architecture: Dalvik is a register-based virtual machine unlike the traditional stack based one (like JVM). This helps minimize overhead that would be incurred by constantly pushing and popping values off the stack making execution faster, especially on low memory mobile devices.Optimized for Mobile Devices: It is an optimizing BSJ virtual machine that was designed specifically to run on the Android operating system, with low memory and power requirements. It virtualizes multiple apps by running a separate instance of the VM for each application as it does best in simultaneously handling many applications.Dex Bytecode Execution: Dalvik uses. dex (Dalvik Executable) files—these compile the source into a more compact bytecode in order to make it better-suited for low-memory environments. We compile multiple Java classes into a single. dex file so that it was just 63% of the original size, meaning a small parse and load time on mobile devices.Just In Time (JIT) Compilation: Dalvik was released from Android 2.2 (Froyo) and later included a JIT compiler to translate the bytecode into machine code directly at execution runtime, which could eventually boost frequently executed codes performance.Garbage Collection: The above code ran on the DALVIK which has automatic garbage collection to make sure that apps will not consume large amount of resources and thus there is no effect in system stability.Support for Multithreading: Dalvik is multithreading mobile machine can have multiple threads at the same time for an application to work on. This feature is one of the most required things when it comes to mobile apps e.g., background process along with foreground user tracking.Advantages of DVM(Dalvik Virtual Machine)DVM supports the Android operating system only.In DVM executable is APK.Execution is faster.From Android 2.2 SDK Dalvik has it's own JIT (Just In Time) compiler.Applications are given their own instances.Disadvantages of DVM(Dalvik Virtual Machine)DVM supports only Android Operating System.For DVM very few Re-Tools are available.App Installation takes more time due to dex.More internal storage is required.Application of Dalvik Virtual MachineRunning Android Apps: It improves how apps run by using processing power and less memory, ensuring smooth work.Support for Multitasking: It allows multiple android apps to run at the same time without interrupting each other sped , making multi tasking smoother on mobile devices. Memory Management: It manages memory more efficiently than older system, making sure that apps can run well even on devices with limited memory.ConclusionThe Dalvik Virtual Machine was essential for the early Android phones. DVM is very handy to build an android apps because it could be optimized for mobile, take care of low-memory & handling the multiple app. In more recent Android versions, DVM has been replaced by ART but it paved the road on how apps get executed in an Android system. Starting March 27, 2025, we recommend using android-latest-release instead of aosp-main to build and contribute to AOSP. For more information, see Changes to AOSP. Android runtime (ART) is the managed runtime used by apps and some system services on Android. ART and its predecessor Dalvik were originally created specifically for the Android project. ART as the runtime executes the Dalvik executable (DEX) format and DEX bytecode specification. ART and Dalvik are compatible runtimes running DEX bytecode, so apps developed for Dalvik should work when running with ART. However, some techniques that work on Dalvik do not work on ART. For information about the most important issues, see Verifying app behavior on the Android runtime (ART). ART features Here are some of the major features implemented by ART. Ahead-of-time (AOT) compilation ART introduces ahead-of-time (AOT) compilation, which can improve app performance. ART also has tighter install-time verification than Dalvik. At install time, ART compiles apps using the on-device dex2oat tool. This utility accepts DEX files as input and generates a compiled app executable for the target device. The utility should be able to compile all valid DEX files without difficulty. However, some post-processing tools produce invalid files that may be tolerated by Dalvik but cannot be compiled by ART. For more information, see Addressing Garbage Collection Issues. Improved garbage collection Garbage collection (GC) is very resource intensive, which can impair an app's performance, resulting in choppy display, poor UI responsiveness, and other problems. ART improves garbage collection in several ways: Mostly concurrent design with a single GC pause Concurrent copying to reduce background memory usage and fragmentation The length of the GC pause is independent of the heap size Collector with lower total GC time for the special case of cleaning up recently-allocated, short-lived objects Improved garbage collection ergonomics, making concurrent garbage collections more timely, which makes GC_FOR_ALLOC events extremely rare in typical use cases Development and debugging improvements ART offers a number of features to improve app development and debugging. Support for sampling profiler Historically, developers have used the Traceview tool (designed for tracing app execution) as a profiler. While Traceview gives useful information, its results on Dalvik have been skewed by the per-method-call overhead, and use of the tool noticeably affects run time performance. ART adds support for a dedicated sampling profiler that does not have these limitations. This gives a more accurate view of app execution without significant slowdown. Sampling support was added to Traceview for Dalvik in the KitKat release. Support for more debugging features ART supports a number of new debugging options, particularly in monitor- and garbage collection-related functionality. For example, you can: See what locks are held in stack traces, then jump to the thread that holds a lock. Ask how many live instances there are of a given class, ask to see the instances, and see what references are keeping an object live. Filter events (like breakpoint) for a specific instance. See the value returned by a method when it exits (using “method-exit” events). Set field watchpoint to suspend the execution of a program when a specific field is accessed and/or modified. Improved diagnostic detail in exceptions and crash reports ART gives you as much context and detail as possible when runtime exceptions occur. ART provides expanded exception detail for java.lang.ClassCastException, java.lang.ClassNotFoundException, and java.lang.NullPointerException. (Later versions of Dalvik provided expanded exception detail for java.lang.ArrayIndexOutOfBoundsException and java.lang.ArrayStoreException, which now include the size of the array and the out-of-bounds offset, and ART does this as well.) For example, java.lang.NullPointerException now shows information about what the app was trying to do with the null pointer, such as the field the app was trying to write to, or the method it was trying to call. Here are some typical examples: java.lang.NullPointerException: Attempt to write to field 'int android.accessibilityservice.AccessibilityServiceInfo.flags' on a null object reference java.lang.NullPointerException: Attempt to invoke virtual method 'java.lang.String java.lang.Object.toString()' on a null object reference ART also provides improved context information in app native crash reports, by including both Java and native stack information. Report problems If you run into any issues that aren't due to app JNI issues, report them through the Android Open Source Project Issue Tracker. Include an adb bugreport and link to the app in Google Play store if available. Otherwise, if possible, attach an APK that reproduces the issue. Reminder: Issues (including attachments) are publicly visible. Dalvik Virtual Machine Register Based VM The language is designed by Dan Bornstein, some cool features and nature of Apple Swift iOS Programming Language he gave it a try with the initial compiler implementations examples contributed from other Google engineers at its Android mobile phone platform. The Dalvik virtual machine was named in that way after Bornstein, a resident of the nearby village "Dalvík", born on ''Eyjafjörður''.What is Dalvik Virtual Machine?Dalvik Virtual Machine (DVM) is the custom program introduced for Android apps. It takes the Java code and creates an optimized version of it in a file with .dex(extension) which is known as Dalvik executable. This format allows the apps to run quickly with fewer resources, i.e. on mobile phones and low-memory, slower devices. This is different from a typical Java Virtual Machine (JVM), as it optimized especially for Android to run apps that use less memory and are compatible with the millions of devices running on various versions.Simple Working of DVMJava Source CodeThe original development of Android apps is mostly done using Java programming language that developers use to write the instructions your app executes.CompilationWe need to convert this textual Java code into form understandable by computer. This is commonly referred to as compilation. This Java bytecode is an executable version of your code.Conversion to Dalvik BytecodeThe Java bytecode is transformed into a different format known as Dalvik byte code. They use a tool called dx, to do this and make the code run smoothly on Android devices.Dalvik Virtual Machine (DVM)Last, but not least the Dalvik bytecode is run on the Dalvik Virtual Machine (DVM). DVM has the special ability to run the applications on mobile devices which have limited capabilities of memory and battery.Features of Dalvik Virtual MachineRegister-Based Architecture: Dalvik is a register-based virtual machine unlike the traditional stack based one (like JVM). This helps minimize overhead that would be incurred by constantly pushing and popping values off the stack making execution faster, especially on low memory mobile devices.Optimized for Mobile Devices: It is an optimizing BSJ virtual machine that was designed specifically to run on the Android operating system, with low memory and power requirements. It virtualizes multiple apps by running a separate instance of the VM for each application as it does best in simultaneously handling many applications.Dex Bytecode Execution: Dalvik uses. dex (Dalvik Executable) files—these compile the source into a more compact bytecode in order to make it better-suited for low-memory environments. We compile multiple Java classes into a single. dex file so that it was just 63% of the original size, meaning a small parse and load time on mobile devices.Just In Time (JIT) Compilation: Dalvik was released from Android 2.2 (Froyo) and later included a JIT compiler to translate the bytecode into machine code directly at execution runtime, which could eventually boost frequently executed codes performance.Garbage Collection: The above code ran on the DALVIK which has automatic garbage collection to make sure that apps will not consume large amount of resources and thus there is no effect in system stability.Support for Multithreading: Dalvik is multithreading mobile machine can have multiple threads at the same time for an application to work on. This feature is one of the most required things when it comes to mobile apps e.g., background process along with foreground user tracking.Advantages of DVM(Dalvik Virtual Machine)DVM supports the Android operating system only.In DVM executable is APK.Execution is faster.From Android 2.2 SDK Dalvik has it's own JIT (Just In Time) compiler.Applications are given their own instances.Disadvantages of DVM(Dalvik Virtual Machine)DVM supports only Android Operating System.For DVM very few Re-Tools are available.App Installation takes more time due to dex.More internal storage is required.Application of Dalvik Virtual MachineRunning Android Apps: It improves how apps run by using processing power and less memory, ensuring smooth work.Support for Multitasking: It allows multiple android apps to run at the same time without interrupting each other sped , making multi tasking smoother on mobile devices. Memory Management: It manages memory more efficiently than older system, making sure that apps can run well even on devices with limited memory.ConclusionThe Dalvik Virtual Machine was essential for the early Android phones. DVM is very handy to build an android apps because it could be optimized for mobile, take care of low-memory & handling the multiple app. In more recent Android versions, DVM has been replaced by ART but it paved the road on how apps get executed in an Android system.