In the Dalvik

Dainius Jocas

April 10, 2013

Today we are going to start to look at the code which is at the very core of Dalvik VM (Virtual Machine). To be more specific, we are going to talk about Android's run-time system, whose core part is the Dalvik VM.

By definition run-time system is a software designed to support the execution of computer programs. Intuitively, we can think of a runtime as a set of libraries that is responsible for low level tasks, e.g. dynamic memory allocation in C. In addition to the basic low-level tasks of the program, a runtime system may also implement higher-level behaviour and even support type checking, debugging, or code generation and optimization. Dalvik as a runtime, besides providing low-level libraries, implements lots of higher-level functionality, e.g. type checking.

To investigate implementation of Dalvik, means to investigate an implementation of the run-time system. And, therefore, to investigate the implementation of the run-time system means to investigate the implementation of a set of tools of the run-time system, e.g. dexopt – a tool which verifies and optimizes all of the classes in the DEX file.

Dalvik project, at the source code level, is a rich set of tools:

- dalvikvm – program to support a command-line invocation of the Dalvik VM.
- dexdump – this tool is intended to mimic "objdump". Objdump displays information about one or more object files.
- dexgen – the dex code generator project. It provides API for creating dex classes in runtime which is needed e.g. for class mocking.
- dexlist – tool that lists all methods in all concrete classes in one or more DEX files.
- dexopt – a tool which verifies and optimizes all of the classes in the DEX file.
- dx – Dalvik eXchange, the thing that takes in class files and reformulates them for consumption in the VM.
- libdex – tool which is responsible for accessing .dex (Dalvik Executable Format) files.
- hit – ??
- opcode-gen – set of scripts for modification of opcodes.

- tools – This tool runs a host build of dalvikvm in order to preoptimize dex files that will be run on a device.
  - dexdeps – DEX external dependency dump. This tool dumps a list of fields and methods that a DEX file uses but does not define.
  - dmtracedump – is a tool that gives you an alternate way of generating graphical call-stack diagrams from trace log files (instead of using Traceview).
  - gdbjithelper – kind of disassembler.
  - hprof-conv – tool to strip Android-specific records out of hprof data.
- vm – actual implementation of a VM:
  - alloc – Garbage-collecting memory allocator.
  - analysis – Dalvik bytecode structural verifier.
  - compiler –
  - hprof – Preparation and completion of hprof data generation.
  - interp – Main interpreter entry point and support functions.
  - jdwp – Java Debug Wire Protocol support. Prints a list of available JDWP processes on a given device.
  - mterp – the opcode interpreter
  - static opcode check;
  - ...

As for today, we'll see what happens, when a new VM is created.