Dalvik VM

Reinhard Penn, Sebastian Ratzenbck

***Abstract*—**

I. Allgemeines

Bei Dalvik handelt es sich um eine Open Source Software die von Dan Bornstein entwickelt wurde. Benannt ist sie nach einem Fischerdorf in Eyjafjrur, Island. Verffentlicht wurde die Software unter Apache License 2.0. Die Ausfhrung findet auf einem Linux Kernel statt.

Verwendung findet Dalvik in Googles Betriebssystem Android. Einsatz findet Dalvik hierbei als virtuelle Maschine. Der Hauptverwendungsbereich liegt im Mobilbereich, wie zum Beispiel bei Smartphones, Tablets und seit neuestem auch bei Smart Tvs und Wearables, wie Smartwatches.

II. Architektur

In diesem Kapitel wird die Architektur der Dalvik Virtual Machine erklrt. Besonderes Augenmerk wird hierbei auf den Aufbau der *.dex* Datei gelegt.

A. Funktion

In Android bekommt jeder Prozess seine eigene virtuelle Maschine, dabei handelt es sich um eine Dalvik VM. Sie hnelt teilweise einer Java VM. Ein bedeutender Unterschied ist allerdings, dass eine Java VM stapelbasiert und eine Dalvik VM registerbasiert arbeitet. Diese registerbasierte Arbeitsweise lehnt sich an moderne Prozessorarchitekturen an. Sie verarbeitet Registermaschinencode, dadurch wird die Dalivk VM schneller als die Java VM und ist ressourcenschonender.

Ein weiterer bedeutender Unterschied liegt darin, dass eine Dalvik VM klassische Java Bibliotheken nicht untersttzt, zum Beispiel *AWT* und *Swing*. Es werden eigene Bibliotheken verwendet, die Apache Harmony als Grundlage verwenden.

Ein wichtiger Bestandteil der *SDK* ist das Tool *dx*. Es sorgt dafr, dass Java Binrdaten in Dalvik Executables umgewandelt werden. Das heit es wandelt *.class* Dateien in *.dex* Dateien um. Bei dieser Umwandlung knnen auch mehrere *.class* Dateien zu einer *.dex* Datei zusammengefasst werden, beziehungsweise kann der Speicherbedarf, mithilfe von *.odex* Dateien optimiert werden.

B. .dex Format

In .dex Dateien werden die Klassen Definitionen und die dazugehrigen Daten gespeichert. Es handelt sich dabei um die ausfhrbaren Dateien der Dalvik VM. Android Programme werden zuerst in Java Bytecode kompiliert. Dieser wird im Anschluss in Dalvik Bytecode bersetzt.

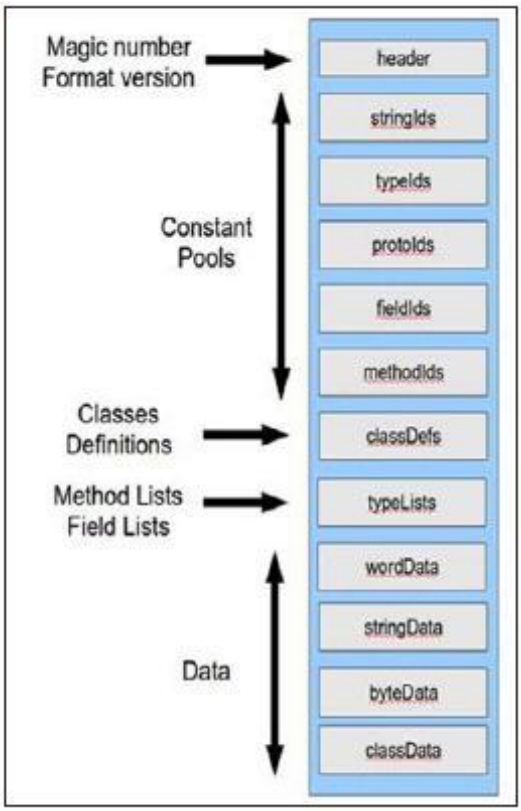

Fig. 1. Dex Dateiformat

```
ubyte[8] DEX_FILE_MAGIC =
{ 0x64 0x65 0x78 0x0a 0x30 0x33 0x35 0x00 }
= "dex\n035\0"
```

Ausschnitt 1. Dex Header

In Abbildung 1 ist der Aufbau einer .dex Datei im Vergleich zu einer .class Datei von Java zu sehen. Fr alle folgenden Listen der Datei drfen keine doppelten Eintrge vorhanden sein.

1) Header: In Ausschnitt 1 ist der Header der *.dex* Datei zu sehen. Dabei handelt es sich um eine bestimmte Bytefolge die den Start der Datei bestimmt.

2) StringIds: Hierbei handelt es sich um eine Liste aller Strings die in der Datei verwendet werden. Die Liste muss sortiert sein und darf keine doppelten Eintrge enthalten. Mgliche Inhalte sind zum Beispiel konstante Strings und

Funktionsnamen.

3) TypeIds: Dies ist eine Liste aller verwendeten Typen dieser Datei. Dazu gehren Klassen, Arrays, und Primitive Datentypen. Diese Liste ist nach den *StringIds* sortiert.

4) ProtoIds: In dieser Liste befinden sich alle Methoden Prototypen, die in *.dex* Datei verwendet werden. Die Prototypen sind primr anhand ihrer Rckgabetypen sortiert und danach anhand ihrer Argumente.

5) FieldIds: Diese Liste enthlt alle Felder die von der *.dex* Datei referenziert werden. Diese Liste wird anhand des Feldtypen und Feldnamen sortiert.

6) MethodIds: In der Methoden Liste werden alle von der *.dex* Datei referenzierten Methoden aufgelistet. Diese Liste wird nach dem Methodennamen und dem Methodenprototypen sortiert.

7) ClassDefs: In dieser Liste werden die Klassendefinitionen der *.dex* Datei angegeben. Die Liste muss so geordnet werden, dass bergeordnete Klassen und Interfaces vor den davon ableitenden Klassen angefhrt werden.

8) Data: In dem Data werden zustzliche Daten fr die oben angefhrten Listen gespeichert. Die verschiedenen Elemente haben verschiedene Alignements und padding bytes werden, wenn notwendig, eingefgt. Des weiteren befinden sich in diesem Bereich die Daten von statisch verlinkten Dateien. Bei nicht verlinkten Dateien ist dieser Teilbereich leer.