Abstract

This paper presents a detailed study of the mechanism to design a compiler of Smali language to generate optimized Android applications. Smali language, which includes the dex bytecode, is the assembly language under Android OS, it is generated from the Java source code. The phases of designing the target compiler are described and the structure of files that are the input and output of the compiler are explained.

The structure of the input files of ART (Android RunTime) compiler is explained, with the focus on the dex file (Dalvik EXecutable) and its corresponding Smali language file, that includes dex bytecode code. The proposed compiler, which is called MySMALI compiler, generates optimized Smali code by replacing some blocks in the Smali code by other blocks more efficient in performance and equivalent in behavior with original blocks. Reverse Engineering techniques are used to decompile and verify the correctness of the generated optimized APKs. As result, an optimized compiler is designed and the experimental evaluation shows that the compiler is
able to save from 4.8% to 12.9% of the overall execution time in various application scenarios. This ratio of improvements increases up with the size and complexity of the optimized code.

References

5. Abhishek Vasisht Bhaskar, 2016, "Automated code extraction from packed android applications", Syracuse University.
6. CENSUS S.A, 2015, "Fuzzing Objects d’ART Digging Into the New Android L Runtime Internals".
25. Malaga, Spain, 2018, "Model Checking Software", 25th International Symposium, SPIN.

**Index Terms**

Computer Science  
Software Engineering

**Keywords**

Design Compiler, lex and yacc, Lexical analyses, Smali language, bytecode optimization, APK Decompiler,