Android アプリのサイズ削減システム

河 崎 雄 大[†1] 大 山 恵 弘[†1]

1. はじめに

近年，スマートフォンが普及してきており注目を集めてきている．中でも Google 社開発の Android は注目を集めており，現在，多くの Android 端末が販売されている．しかし，その端末の内部ストレージは限られており，大きな内部ストレージを持つ端末は限られている．その結果，端末の内部ストレージの容量不足が深刻な問題となっている．例えば，HTC Desire X06HT II ではユーザが使用可能な内部ストレージは 150MB しかない．

しかし，それにも関わらずサイズの大きな Android アプリ (以下，アプリ) は多く存在している．例えば，Google Earth，Skype などのアプリがあり，それらのアプリのサイズは約 20MB，約 10MB である．これらのアプリは便利ではあるが，内部ストレージの少ない端末の容量不足に拍車をかけている．

2. 目的と方針

本研究では，Android 端末の内部ストレージの容量不足を改善するために，Android アプリのサイズ削減システムの提案を行う．本システムでは apk ファイル内に存在するプログラムデータを外部に移し，実行時に，本体側からそのデータを参照する手法をとる．

提案する手法は，apk ファイルからパッケージ名などの必要な情報を調べる部分と，プログラムデータを取り出し，参照できる形に変更する部分から成り立つ．apk ファイルから必要な情報を調べる部分では，既存のツール[1)2)] を使用して処理を行う．また，プログラムデータを取り出し，参照できる形に変更する部分は本システム独自のものを作成し，それを使用して処理を行う．

3. apk ファイル

apk ファイルとは Android の実行ファイルであり，ZIP 形式で圧縮されたファイルとなっている．apk ファイルは以下のような 2 つのディレクトリと 3 つのバイナリファイルから成り立っている．

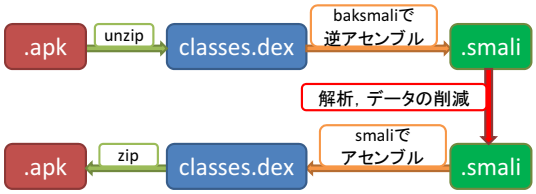

図 1 システムの概要図

- META-INF/: 署名ファイルが入っているディレクトリ
- res/: 画像などのリソースが入っているディレクトリ
- AndroidManifest.xml: アプリ情報を持つファイル
- classes.dex: プログラムデータのファイル
- resources.arsc: 言語リソースなどのファイル

4. 提案システム

提案システムでは，apk ファイルを展開し，プログラムデータを編集し，再度 ZIP 形式に圧縮して元の状態に戻すことによってアプリのサイズを削減させる．

本システムではまず，apk ファイルを展開し，その中の AndoridMainfest.xml を axmlprinter2[1)] というツールを用いて変換する．そして，生成されたファイルの解析を行い，パッケージ名などの必要な情報を得る．次に，図 1 のように classes.dex を baksmali[2)] というツールを用いて逆アセンブルし，smali ファイルを作成する．作成された smali ファイルにはアセンブリコードがテキスト形式で書かれているので，その smali ファイルを解析し，プログラムデータを編集し，データを削減していく．そして，編集が行われた smali ファイルを smali[2)] というツールを用いてアセンブルし，新たな classes.dex を作成する．最後に，作成したファイル群を ZIP 形式に圧縮して apk ファイルを作成し，新たに署名を加える．署名に関しては新たに作成したものを用いる．

5. プログラムデータの削減手法

本システムで用いるプログラムの削減手法は，アプ

†1 電気通信大学 電気通信学部 情報工学科

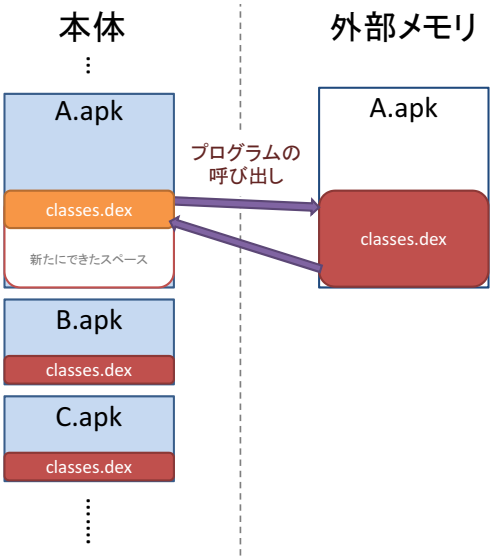

図 2 プログラムデータの削減例

リのプログラムデータである dex ファイルを外部メモリーに移し，アプリの本体側からその dex ファイルを参照するという方法である．本システムによるプログラムデータの削減手法の例を図 2 に示す．

しかし，単純に dex ファイルを外部メモリーに移してから参照しようとするとうまくいかないので，多少の編集を行ってから外部に移す．具体例を述べると，dex ファイルのエントリポイントとなるクラスへの public なメソッドの追加やそのクラスから他クラスを呼び出す際の呼び出し方の変更などである．

6. 関 連 研 究

アプリのプログラムデータを解析する既存研究には Octeau らの研究[3] がある．この研究は dex ファイルから Java の class ファイルを作成し，最終的に Java ファイルに戻してから解析を行っている．しかし，この研究ではセキュリティ的にまずい命令を検出するために解析を行っており，プログラムデータを外部に移すため解析を行う本研究とは解析の目的という点で異なっている．

7. 現状と今後

現状は，ビルド前の Java ファイルを用いてプログラムデータの削減手法の調査を行い，Java での実装を行った．また，予備実験も行い，動作していることも確認できた．

今後は，smali ファイルでの実装を行い，システムによるアプリの圧縮率と実行時のオーバヘッドの評価を行う予定である．また，セキュリティ的にまずい部分が存在するのでその部分の改善も行っていく予定である．

参 考 文 献

1) axmlprinter2.
`http://code.google.com/p/android4me/`
2) smali, baksmali.
`http://code.google.com/p/smali/`
3) D. Octeau, W. Enck, and P. McDaniel: The ded Decompiler, *Technical Report NAS-TR-0140-2010, Network and Security Research Center*, 2010.