B-002

Android における
アプリケーション起動履歴を用いた終了プロセス選定

A New Process Termination Policy on Android Devices

野村　駿†　　永田　恭輔†　　中村　優太†　　山口　実靖†

Shun Nomura　　Kyosuke Nagata　　Yuta Nakamura　　Saneyasu Yamaguchi

1. はじめに

Android はスマートフォン，タブレット PC，音楽プレイヤーなど様々なデバイスの OS として採用されており，その重要性が高まっている．

Android には，low memory killer と呼ばれる独自のメモリ管理システムが搭載されており，独自のルールに従ってメモリの管理を行なっている．low memory killer では，メモリの空き容量を確保するためにプロセスを強制終了する．このプロセスの強制終了により，ユーザが再度同じアプリケーションを使用する場合に，プロセスの再起動が必要となりユーザの利便性を低下させることがある．

本研究では，強制終了するアプリケーションの選定の改善によりプロセス再起動にともなうユーザの利便性の低下を軽減することを目的として，新しい選定手法を提案しその評価を行う．

2. low memory killer

low memory killer は，メモリの空き容量が閾値以下まで下がった場合に起動され，*adj* と *minfree* の関係に基づいてプロセスを選定し強制終了するプログラムである． low memory killer が起動されると，強制終了するプロセスの選定のために起動している全てのプロセスの *adj* を比較し，*adj* の数値がより高いプロセスから順に強制終了する．最高 *adj* のプロセスが複数存在する場合，メモリ使用量を比較し，メモリ使用量のより多いプロセスを強制終了する．

adj と *minfree* の関係は，/init.rc で定義されている．*adj* はプロセスの状態，種類により定まり，プロセスの状態が変化するたびに数値が増減する． *adj* が小さいプロセスほど強制終了されづらい．例えば，フォアグラウンド状態のアプリケーションは *adj* が最小であり，ホームアプリケーションも *adj* が小さく設定されている．対してバックグラウンド状態のアプリケーションは *adj* が大きく設定されている．*minfree* は，プロセス強制終了実行の閾値となる空きページ数であり，*adj* によってランク分けされている．

3. アプリケーションの起動手順

Android のアプリケーションを新規に起動する場合，図 1 の手順に従って起動される．

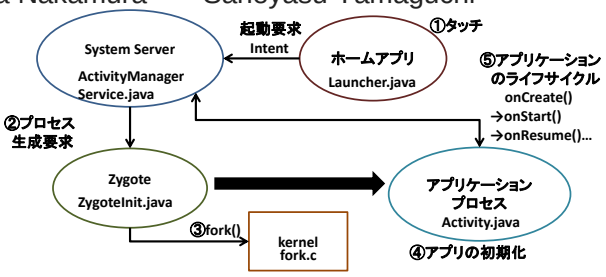

図 1 アプリケーションの起動手順

①ユーザがアプリケーションのアイコンをタッチする．ホームアプリケーション(Launcher)が起動要求のインテントを ActivityManager に送信する．②ActivityManager が Zygote にプロセス生成要求を送信する．③Zygote が自分自身を fork し，子プロセスを生成する．④新しいプロセスが初期化される．⑤アプリケーションのライフサイクルに従って onCreate()，onStart()，onResume()が呼び出され，アプリが起動する[1].

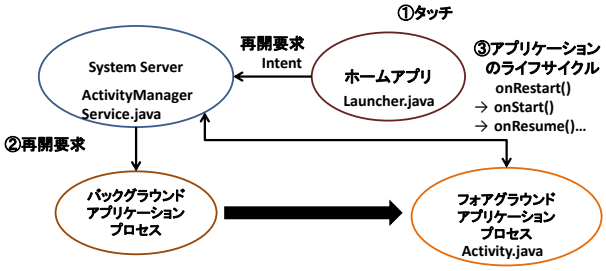

図 2 アプリケーションの再開手順

また，起動済みであるがバックグラウンド状態にあるプロセスの再開は，図 2 の手順に従って行われる．①ユーザがアプリケーションのアイコンにタッチする．そしてホームアプリケーションが再開要求のインテントを ActivityManager に送信する．②ActivityManager は対象のバックグラウンドアプリケーションプロセスに再開要求を出す．バックグラウンドアプリケーションプロセスは再開要求を受けてアプリケーションプロセスとして起動する．③アプリケーションのライフサイクルに従って onRestart()，onStart()，onResume()が呼び出され，フォアグラウンド状態となる．

本稿では前者を「新規起動」，後者を「再開」と呼び，新規起動の場合はアプリケーションのライフサイクルにおける onCreate()の開始から

† 工学院大学大学院 工学研究科 電気・電子工学専攻
Electrical Engineering and Electronics, Kogakuin University Graduate School

onResume()の開始までの時間を新規起動にかかった時間とし，再開の場合は onRestart()の開始から onResume()の開始までの時間を再開にかかった時間とする．

low memory killer によりプロセスが強制終了された場合，再度アプリケーションを起動する際に再開ではなく新規起動の手順で起動が行われる．一般的に，新規起動の方が再開よりも起動時間が長いため[1]，再度使用する可能性のあるアプリケーションはなるべく再開状態で開始されることが望ましい．

4. 提案手法

提案手法として，LRU により強制終了するプロセスを選定する手法を提案する．

本手法では，ユーザが起動したアプリケーションを LRU により管理し，LRU 順にて新しいアプリケーションプロセスの adj を下げ強制終了の対象となりにくくする．具体的には，長さ 15 の配列を用意し，ユーザが起動したアプリケーション履歴を保持する．配列内の順は LRU にて管理する．そして，配列の先頭(0 番目)から 3 番目までに格納されているアプリケーション(最近使われたアプリケーション)の adj を-5 する．同様に，4 番目から 9 番目に格納されているアプリケーションの adj を-3, 10 番目から 12 番目に関して adj を-2, 13 番目から 14 番目に関して adj を-1 する．履歴内に記録のないアプリケーションの adj は変更を加えない．履歴の長さや，履歴順の閾値，adj の下げ幅は調整可能である．

5. 評価

5.1 評価方法

標準 low memory killer と提案手法が実装されている Android スマートフォンにおいて，複数のアプリケーションを順に起動し，その起動に要した時間を測定した．実験は，表 1 に示す仕様のスマートフォンを用いて行った．

表 1 測定環境

Device name	Nexus S
OS	Android4.0.3
CPU	Cortex A8 （ Hummingbird ） Processer 1GHz
Memory	512MB

起動するアプリケーションの順番を定めたものを本稿では“シナリオ”と呼び，今回はユーザから許可を得て実生活における 1 日分のアプリケーションの使用履歴を取得し，それをシナリオとした．

5.2 評価結果

上記シナリオの通りアプリケーションを起動し，アプリケーション起動に要した時間を測定した．観測時間を図 3 に示す．図の縦軸は，シナリオの全アプリケーションの起動時間の合計であり，少ないほど優れている．図より，標準手法より提案手法の合計起動時間が短く，提案手法が有効であることが確認された．

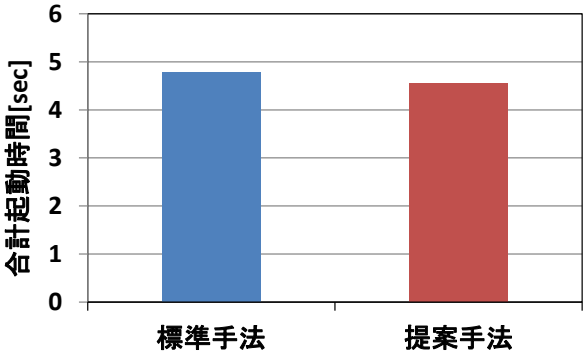

図 3 合計起動時間の比較

また，シナリオにおける新規起動数と再開を比較した表を表 2 として示す．

表 2 起動状態数の比較

手法	新規起動数	再開数
標準手法	22	132
提案手法	19	135

表 2，図 3 より，新規起動の数が減ったことで，合計起動時間が短縮されていることが確認された．

6. まとめ

本研究では low memory killer における強制終了プロセスの選定手法に着目し，アプリケーションの起動履歴を用いた強制終了プロセスの選定手法を提案し評価した．評価の結果，提案手法が標準手法よりもアプリケーションの連続起動における新規起動の数を減らすことができることを確認した．

今後は，onResume()の開始後に発生する処理，ブラウザのページ読み込みなども考慮した選定手法について考察していく予定である．

謝辞

本研究は JSPS 科研費 24300034, 25280022 の助成を受けたものである。

参考文献

[1] Kyosuke Nagata, Saneyasu Yamaguchi, "An Android Application Launch Analyzing System ," 8th ICCM: 2012 International Conference on Computing Technology and Information Management, (2012/04/24)