

# Android における アプリケーション起動時間を考慮したプロセスメモリ管理

野村 駿<sup>†</sup> 服部 拓也<sup>‡</sup> 永田 恭輔<sup>‡</sup> 中村 優太<sup>†</sup> 山口 実靖<sup>†</sup>

工学院大学<sup>†</sup> 工学院大学大学院<sup>‡</sup>

## 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, *linit.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. *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, 次の手順に従って起動される. 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Intent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()が呼び出される[1].

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, 次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Intent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. ③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しフォアグラウンド状態となる. 本論文では前者を「新規起動」, 後者を「再開」と呼ぶ.

## 4. 提案手法

提案手法として, アプリケーションの新規起動時間を考慮して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, 新規起動時間が長く利便性低下の程度が大きいアプリケーションを強制終了されづらくする. 強制終了されづらくすることでアプリケーションが再開で起動され, ユーザの利便性の低下を軽減されることが期待できる.

具体的には, 全てのアプリケーションの新規起動時間を AndroidOS 内で測定し, それらをカーネルの low memory killer 内で管理する. 新規起動時間が平均より長いアプリケーションは新規起動時のユーザの利便性低下が大きいと考え, *adj* を 3 減少させ, 強制終了されづらくする.

新規起動時間は前章における onCreate() から onResume() までの時間とする. 新規起動時間の保持数は有限とし, 記録上限に達した場合は, 古い記録から順に削除していく. これにより,

Process Memory Management Based on Application

Launching Time in Android OS

Shun Nomura<sup>†</sup>, Takuya HATTORI<sup>‡</sup>, Kyosuke NAGATA<sup>‡</sup>,

Yuta Nakamura<sup>†</sup>, Saneyasu YAMAGUCHI<sup>†</sup>

<sup>†</sup>Kogakuin University

<sup>‡</sup>, Kogakuin University Graduate School

最近使用していないアプリケーションが長時間高い優先度であり続けることが回避できると期待される。

## 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 Nexus S(OS:Android4.0.3 , CPU:Cortex A8(Hummingbird) Processor 1GHz , Memory:512MB)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。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、本実験では図1のシナリオ A, B と、Opera Mobile browser, Browser ( Android 標準 ) , 2chMate , TkMixiViewer, Gmail, Camera, twicca の中から連続しないようランダムに 100 個選定したシナリオ C により評価を行った。シナリオ A の Benchmark1, 2 とは、測定用に我々が作成したメモリを消費する起動時間の長いアプリケーションである。

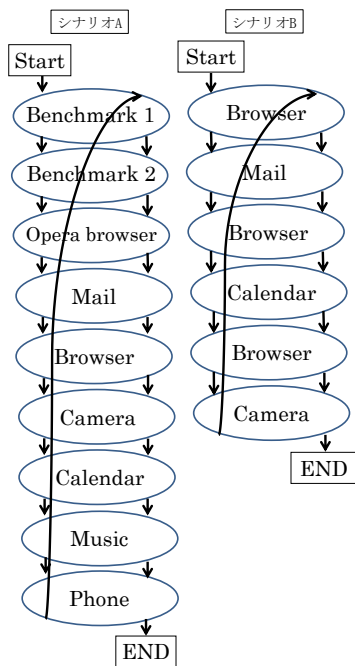


図1 評価シナリオ A, B

### 5.2 評価結果

シナリオ A~C における評価結果を図 2~4 に示す。各図の縦軸は、シナリオのホームアプリケーション(Launcher)以外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、全てのシナリオにおいて標準手法より提案手法の方が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。

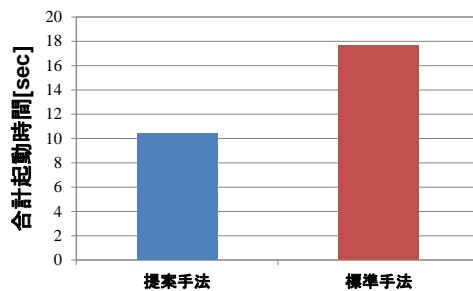


図2 シナリオ A の評価結果

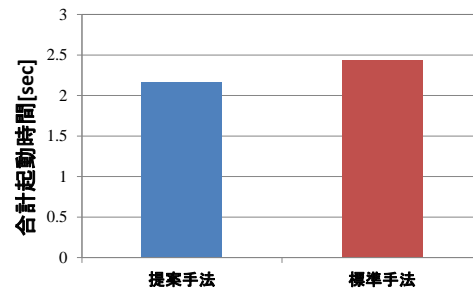


図3 シナリオ B の評価結果

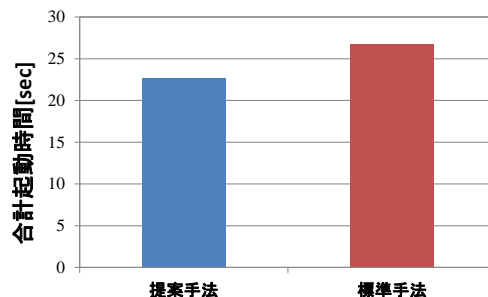


図4 シナリオ C の評価結果

## 6. おわりに

本研究で我々は 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時間を考慮した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起動時間を短縮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に要する時間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 22700039, 24300034 の助成を受けたものである。

## 参考文献

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