

スレッドプライオリティ(優先度)を変更する

mhidaka | 2013年10月4日 | [アプリケーション](#) | [1 Comment](#)

B! 6 **いいね!** 0 **ツイート**

Androidでスレッドの優先度を指定する方法を紹介します。Androidアプリケーションでは、UIスレッドで時間のかかる処理を行ってしまうと応答性に影響が出るのは周知の事実です。たとえば次のような処理はUIスレッドに適さないでしょう。

- アプリ外部：ネットワークやローカルファイル、データベース、ソケットへのアクセスする場合
- アプリ内部：ブロッキング (synchronized等) を利用したクリティカルセクションがある場合
- CPU時間：CPUを長く拘束する処理。たとえば画像処理のような時間がかかることがあらかじめ分かっている場合

さまざまな理由があってマルチスレッド化を検討すると思いますが、スレッドの優先度まで気にすることは稀です。それは上記で示した多くの例では**UIスレッドを守る**ため別スレッドに追い出す処理だから (=UIスレッドに比べて優先度が低いから) です。他のリソースに依存した待ち時間は優先度を高くしても効果が見込めません。

しかしながらVoIPの音声デコードやストリーム動画の受信などデータや処理に時間的な制約がある場合、スレッドの優先度を変える選択肢が取り得ます。その場合はスレッドを1つ作成し、ごく短い時間で処理を完了することをおすすめします。たくさんスレッドをつかって細かく優先度を変えることは、制御の複雑化につながる可能性があります。

優先度を設定する

Androidではandroid.os.ProcessクラスのsetThreadPriorityメソッドを使えばスレッドの優先度を変更できます。

- android.os.Process.setThreadPriority (int tid, int priority)メソッド
- android.os.Process.**setThreadPriority (int priority)**メソッド

即値	定数	説明
19	THREAD_PRIORITY_LOWEST	最も優先度が低いスレッドです。通常利用しません
10	THREAD_PRIORITY_BACKGROUND	バックグラウンド処理に最適な優先度です。デフォルトより低い優先度にすることで応答性への影響を最小限にしています
1	THREAD_PRIORITY_LESS_FAVORABLE	標準的な優先度より、ほんの少し優先を下げた設定値です
0	THREAD_PRIORITY_DEFAULT	アプリのスレッドとして標準的な優先度です
-1	THREAD_PRIORITY_MORE_FAVORABLE	標準的な優先度より、ほんの少し優先を上げた設定値です
-2	THREAD_PRIORITY_FOREGROUND	UIスレッドの設定値です。システムがUIスレッドを作る際に利用します。通常、アプリ開発者は利用しません
-4	THREAD_PRIORITY_DISPLAY	システムがUIを更新するために使う表示用スレッドの設定値です。通常、アプリ開発者は利用しません
-8	THREAD_PRIORITY_URGENT_DISPLAY	システムが画面描画やInputDevice情報を処理するための設定値です。非常に優先度が高いため、通常、アプリ開発者は使用しません
-16	THREAD_PRIORITY_AUDIO	オーディオ処理のための設定値です。アプリケーションは通常、この優先度に変更できません
-19	THREAD_PRIORITY_URGENT_AUDIO	オーディオ処理を最優先で行うための設定値です。アプリケーションは通常、この優先度に変更できません

Linuxの優先度に準じており、優先度(nice値)は最も高い-20から最も低い19まで変更可能です。表からはUIスレッドを超える優先度を設定する場合は描画処理や音声処理など一部の用途が利用しているだけということがわかります。高い優先度をつけたい場合は、処理を分散できないか、ロジックの高速化して対応できないかなど十分に検討してください。

サンプルコードで優先度を指定したスレッドを作るThreadFactoryを見てみましょう。

■PriorityThreadFactory.java

```
1 public class PriorityThreadFactory implements ThreadFactory {
2     private final int mPriority;
3     private final AtomicInteger mNumber = new AtomicInteger(0);
4     private final String mName;
5
6     public PriorityThreadFactory(String name, int priority) {
7         mName = name;
8         mPriority = priority;
9     }
10
11     @Override
12     public Thread newThread(Runnable r) {
13         return new Thread(r, mName + "-" + mNumber.getAndIncrement());
14     }
15
16     @Override
17     public void run() {
18         // android.os.Processを使う場合 : Process.THREAD_PRIORITY_URGENT_DISPLAY
19         // android.os.Process.setThreadPriority(mPriority);
20
21         // java.lang.Threadを使う場合 : Thread.NORM_PRIORITY
22         // this.setPriority( mPriority );
23
24         super.run();
25     }
26 }
27 }
```

PriorityThreadFactoryクラスのコンストラクタで受けとった優先度(mPriority)をandroid.os.Process.setThreadPriorityメソッドで指定しています。


サンプルでは参考までにjava.lang.Thread.setPriorityメソッドでの設定方法も紹介しています。この場合は定数値が異なりますがスレッドの優先度は次の通りです。

- java.Lang.Thread.setPriority(int priority)メソッド

定数値	説明
MIN_PRIORITY	android.os.Process.THREAD_PRIORITY_LOWEST同等 (定数値は異なる)
NORM_PRIORITY	android.os.Process.THREAD_PRIORITY_DEFAULT同等 (定数値は異なる)
MAX_PRIORITY	android.os.Process.THREAD_PRIORITY_URGENT_DISPLAY同等 (定数値は異なる)

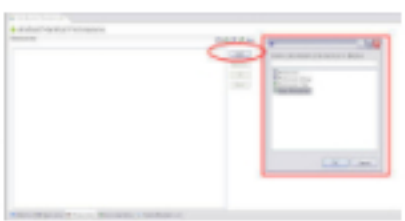
java.lang.Thread.setPriorityメソッドでも設定は可能ですが、表のとおり数が少ないため移植性等を考慮しなければandroid.os.Process.setThreadPriorityメソッドを優先すると良いでしょう (反対にjava.Lang.Thread.setPriorityメソッドを使っているライブラリなど他環境から移植した場合でも、不具合が無ければandroid.os.Process.setThreadPriorityメソッドに置き換える必要はありません) 。

以上、おつかれさまでした。




手書きのデザインがデジタルに

広告 アドビ システムズ...




Android開発初心者がよくやる失敗【コーディング編】 その3

techbooster.org




テレワーク特集

Lenovo




Androidでハードウェアキーを取得する方法 (dispatchKeyEventとgetKeyCodeの使い方)

techbooster.org



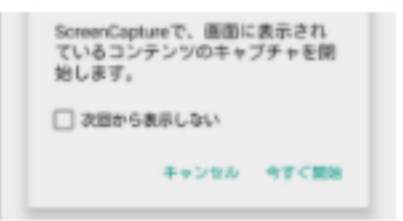
AI/ディープラーニング開発向け - 世界最強のDeep Learning System

広告 hpc.co.jp



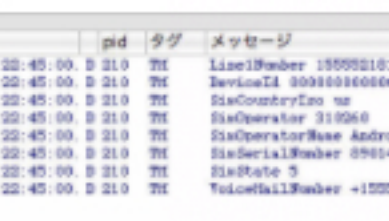
Androidでセンサーの値を取得する方法 (照度、近接、環境光、温度センサーなど)

techbooster.org



Android 5.0 アプリからスクリーンショットを撮影する

techbooster.org



TelephonyManagerを使って端末の情報を取得する

techbooster.org

ABOUT THE AUTHOR



ひつじ (Mhidaka)

Software Engineerだよ。DroidKaigi Organizer / Androidと組込とRe:VIEW. techbooster主宰。mhidaka's writings

<http://booklog.jp/users/mhidaka> 技術書典! <http://techbookfest.org>

ONE COMMENT

2



GOOGLE I/O KEYNOTE ANDROIDまとめ：新機能と背景



GOOGLE I/O KEYNOTE まとめ：現実にも検索を。情報を整理する新機能が発表



GOOGLE I/O 2019まもなくスタート



コミックマーケット95 新刊情報と予約情報



10月8日 技術書典5 新刊情報のおしらせ



GOOGLE I/O DEVELOPER KEYNOTE ANDROID開発のまとめ

新刊発売中



「設計に関する議論の土台となる本を」という著者陣の熱い思いから、ついにAndroidアプリ設計の決定版が登場！



AndroidモダンプログラミングはKotlin & Gradle を特集した解説書です。改訂版Gradle逆引きリファレンス (Android Gradle Plugin 3.0.x対応) など開発に使える技術を中心に解説をお届けします



The Web Explorer 4はWebページの表示速度向上の取り組み、AngularやReact NativeなどWebの冒険者にふさわしい内容です



ConstraintLayoutやreCAPTCHA、Kotlinやドメインを考慮した設計技法、Google I/Oセッションレポートの全13章を収録



わかる!ドメイン駆動設計はソフトウェア設計手法であるドメイン駆動設計 (DDD) の解説書です

スポンサーリンク

フリーランスエンジニア 専門のお仕事検索サイト

最高単価 **200万円**
掲載案件 **6万件**
支払いサイト **最短3日**~
利用ユーザー **12000人**

エンジニアのお仕事探しは **フリーランススタート**

スポンサーリンク

