

# 濃厚ソースが絡むシェフこだわりスパゲティー

プログラミングとかその他

プロフィール



id:boredbone

プログラミングとかその他

検索

リンク

[はてなブログ](#)

[ブログをはじめる](#)

[週刊はてなブログ](#)

[はてなブログPro](#)

最新記事

[Android上のXamarinアプリでSSL接続できない](#)

[C#, Interactive Extensionsでフィボナッチ数列のn番目の項を求める](#)

[\(Windowsデスクトップアプリ\)タグ管理可能な画像ビューア"ShibugakiViewer"リリース](#)

[Visual Studio 2015 Update3で\(GUIから\)テストできない](#)

[WPFでベクタ画像\(EMF\)を表示](#)

月別アーカイブ

参加グループ

[プログラミング](#)

カテゴリー

[C# \(11\)](#)

[Java \(7\)](#)

[Android \(6\)](#)

[その他 \(4\)](#)

[WPF \(3\)](#)

[WinRT \(2\)](#)

[Python \(1\)](#)

[MATLAB \(1\)](#)

[Xamarin \(1\)](#)

[SciPy \(1\)](#)

2014-01-26

## HandlerThreadで非同期処理

[Android](#) [Java](#)

Androidで、メインスレッドとは別にHandlerThreadを立てて、随時そっちにメッセージを投げる。

メッセージはどんどんキューに溜められて行って、古い物から一つずつ処理されていく。同じHandlerThreadに投げられたメッセージが同時に処理されることはない。

まずはスレッドを立ててみる。フィールドにはこんなのを用意。

```
private HandlerThread myThread;
private Handler myThreadHandler;
```

で、どこか適当なところでスレッド開始。

```
myThread = new HandlerThread("testThread");
myThread.start();
myThreadHandler = new Handler(myThread.getLooper());
```

あとは好きなだけメッセージを投げる。

```
myThreadHandler.post(new Runnable() {
    public void run() {
        Log.d("async", "processed");
    }
});
```

アプリが終わる時(Activity.onPause()とか)にはスレッドを終わらせてやる。

```
myThread.quit();
```

quit()が呼ばれた瞬間に処理中のメッセージがあった場合、それが終わってからスレッドが終了するっぽい？ので、場合によっては本人にその他の後片付けと合わせて終了処理をさせた方がいいのかも。

```
myThreadHandler.post(new Runnable() {
    public void run() {
        myThread.quit();
        myThread = null;
    }
});
```

名前からして安全そうなquitSafely()メソッドもあるけど、これはAPI level18からなので古い端末では使えない。

上のquitメソッドを含むメッセージを投げた後、実際にquitが実行される前にさらに他のメッセージを投げても、普通にpostは成功する。でも実際に処理はされない。そりゃ当然。

一方quitが実行された後、もう終わってるスレッドにメッセージを投げると、logに警告は出るけどアプリが落ちたりはしない模様。これは都合が良いような良くないような。

boredbone 2014-01-26 02:54

1

Tweet

コメントを書く

« hypot

コード表示 »