**Android Framework 框架系列之 电量优化Doze 模式**

电量优化一直是Android 开发中的头等问题。本篇将分析一下Android M 以上电量优化措施电量优化相关的部分知识点。

注：文章参考MTK手机解决方案文档

通过本篇文章阅读，你将收获以下知识点:

1.Doze 模式  
2.空闲状态下，优化app耗电  
3.Doze 模式下的限制措施  
4.Doze 模式概要  
5.Doze 模式涉及的类如下:  
6.Doze 模式状态  
7.Doze 白名单  
8.Doze 模式测试方法  
9.开启Doze dubug 调试开关

**1.Doze 模式**

当设备处于非充电、灭屏状态下静止一段时间，设备将进入睡眠状态，进入Doze模式，延长电池使用时间。Doze模式下系统会定期恢复正常操作，异步执行app的一些同步数据等操作。比如很长时间不使用，系统会允许设备一天访问一次网络等。当设备处于充电状态下，系统将进入标准模式，app执行操作将不被限制。

**2.空闲状态下，优化app耗电**

在用户没有使用app的情况下，系统会使app处于idle 状态,  
在空闲状态下，系统将会禁止app网络访问以及数据同步

**3.Doze 模式下的限制措施**

1.禁止网络访问  
2.忽略Wake lock  
3.忽略Alarms(setAlarmClock() 、AlarmManager.setAndAllowwhileIdle() 这两个方法除外)  
4.忽略WIFI 扫描  
5.同步作业调度程序将不被执行

**4.Doze 模式概要**

Doze模式概要

**5.Doze 模式涉及的类如下:**

frameworks/base/services/core/java/com/android/server/DeviceIdleController.java

/\*\*

\* Keeps track of device idleness and drives low power mode based on that.

\*/

public class DeviceIdleController extends SystemService

implements AnyMotionDetector.DeviceIdleCallback {

**6. Doze 模式状态**

* ACTIVE：手机设备处于激活活动状态
* INACTIVE：屏幕关闭进入非活动状态
* IDLE\_PENDING：每隔30分钟让App进入等待空闲预备状态
* IDLE：空闲状态
* IDLE\_MAINTENANCE：处理挂起任务

Doze 模式状态

对应的 Doze 模式状态如下：

Doze模式状态图

**active---> inactive ---> idle\_pending**

运动模式检测

void handleMotionDetectedLocked(long timeout, String type) {

// The device is not yet active, so we want to go back to the pending idle

// state to wait again for no motion. Note that we only monitor for motion

// after moving out of the inactive state, so no need to worry about that.

boolean becomeInactive = false;

if (mState != STATE\_ACTIVE) {

scheduleReportActiveLocked(type, Process.myUid());

mState = STATE\_ACTIVE;

mInactiveTimeout = timeout;

mCurIdleBudget = 0;

mMaintenanceStartTime = 0;

EventLogTags.writeDeviceIdle(mState, type);

addEvent(EVENT\_NORMAL);

becomeInactive = true;

}

if (mLightState == LIGHT\_STATE\_OVERRIDE) {

// We went out of light idle mode because we had started deep idle mode... let's

// now go back and reset things so we resume light idling if appropriate.

mLightState = STATE\_ACTIVE;

EventLogTags.writeDeviceIdleLight(mLightState, type);

becomeInactive = true;

}

if (becomeInactive) {

becomeInactiveIfAppropriateLocked();

}

}

**idle\_pending ————>sensing**

@Override

public void onAnyMotionResult(int result) {

if (DEBUG) Slog.d(TAG, "onAnyMotionResult(" + result + ")");

if (result != AnyMotionDetector.RESULT\_UNKNOWN) {

synchronized (this) {

cancelSensingTimeoutAlarmLocked();

}

}

if (result == AnyMotionDetector.RESULT\_MOVED) {

if (DEBUG) Slog.d(TAG, "RESULT\_MOVED received.");

synchronized (this) {

handleMotionDetectedLocked(mConstants.INACTIVE\_TIMEOUT, "sense\_motion");

}

} else if (result == AnyMotionDetector.RESULT\_STATIONARY) {

if (DEBUG) Slog.d(TAG, "RESULT\_STATIONARY received.");

if (mState == STATE\_SENSING) {

// If we are currently sensing, it is time to move to locating.

synchronized (this) {

mNotMoving = true;

stepIdleStateLocked("s:stationary");

}

} else if (mState == STATE\_LOCATING) {

// If we are currently locating, note that we are not moving and step

// if we have located the position.

synchronized (this) {

mNotMoving = true;

if (mLocated) {

stepIdleStateLocked("s:stationary");

}

}

}

}

}

**7.Doze 白名单**

电量优化白名单  
设置 --电池 --电量优化（menu菜单）  
会设置查看app 电池优化使用情况  
白名单是以xml形式存储(deviceidle.xml)  
查看白名单命令

//主要存放app包名

adb shell dumpsys deviceidle whitelist

白名单代码保存部分代码如下

/\*\*

\* Package names the system has white-listed to opt out of power save restrictions,

\* except for device idle mode.

\*/

private final ArrayMap<String, Integer> mPowerSaveWhitelistAppsExceptIdle = new ArrayMap<>();

/\*\*

\* Package names the system has white-listed to opt out of power save restrictions for

\* all modes.

\*/

private final ArrayMap<String, Integer> mPowerSaveWhitelistApps = new ArrayMap<>();

/\*\*

\* Package names the user has white-listed to opt out of power save restrictions.

\*/

private final ArrayMap<String, Integer> mPowerSaveWhitelistUserApps = new ArrayMap<>();

/\*\*

\* App IDs of built-in system apps that have been white-listed except for idle modes.

\*/

private final SparseBooleanArray mPowerSaveWhitelistSystemAppIdsExceptIdle

= new SparseBooleanArray();

/\*\*

\* App IDs of built-in system apps that have been white-listed.

\*/

private final SparseBooleanArray mPowerSaveWhitelistSystemAppIds = new SparseBooleanArray();

/\*\*

\* App IDs that have been white-listed to opt out of power save restrictions, except

\* for device idle modes.

\*/

private final SparseBooleanArray mPowerSaveWhitelistExceptIdleAppIds = new SparseBooleanArray();

/\*\*

\* Current app IDs that are in the complete power save white list, but shouldn't be

\* excluded from idle modes. This array can be shared with others because it will not be

\* modified once set.

\*/

private int[] mPowerSaveWhitelistExceptIdleAppIdArray = new int[0];

/\*\*

\* App IDs that have been white-listed to opt out of power save restrictions.

\*/

private final SparseBooleanArray mPowerSaveWhitelistAllAppIds = new SparseBooleanArray();

/\*\*

\* Current app IDs that are in the complete power save white list. This array can

\* be shared with others because it will not be modified once set.

\*/

private int[] mPowerSaveWhitelistAllAppIdArray = new int[0];

/\*\*

\* App IDs that have been white-listed by the user to opt out of power save restrictions.

\*/

private final SparseBooleanArray mPowerSaveWhitelistUserAppIds = new SparseBooleanArray();

/\*\*

\* Current app IDs that are in the user power save white list. This array can

\* be shared with others because it will not be modified once set.

\*/

private int[] mPowerSaveWhitelistUserAppIdArray = new int[0];

**8. Doze 模式测试方法**

1.开启Doze  
adb shell dumpsys deviceidle enable  
// or MTK addadb shell setprop persist.config.AutoPowerModes 1  
2.拔掉电池  
adb shell dumpsys battery unplug  
3.调试Doze状态

Active ---idle\_pending---sensing--location---idle --idle\_mantenance

adb shell dumpsys deviceidle step  
4.Dump Doze 状态分析

Doze模式下的信息，包括电池电量优化白名单等  
adb shell dumpsys deviceidle

**9. 开启Doze dubug 调试开关**

默认false 关闭，设置为true 开启DeviceIdleController.java  
private static final boolean DEBUG = false;