Android (/tags/#Android) PowerManager (/tags/#PowerManager) Doze (/tags/#Doze)

Android电源管理之Doze模式专题系列（八）

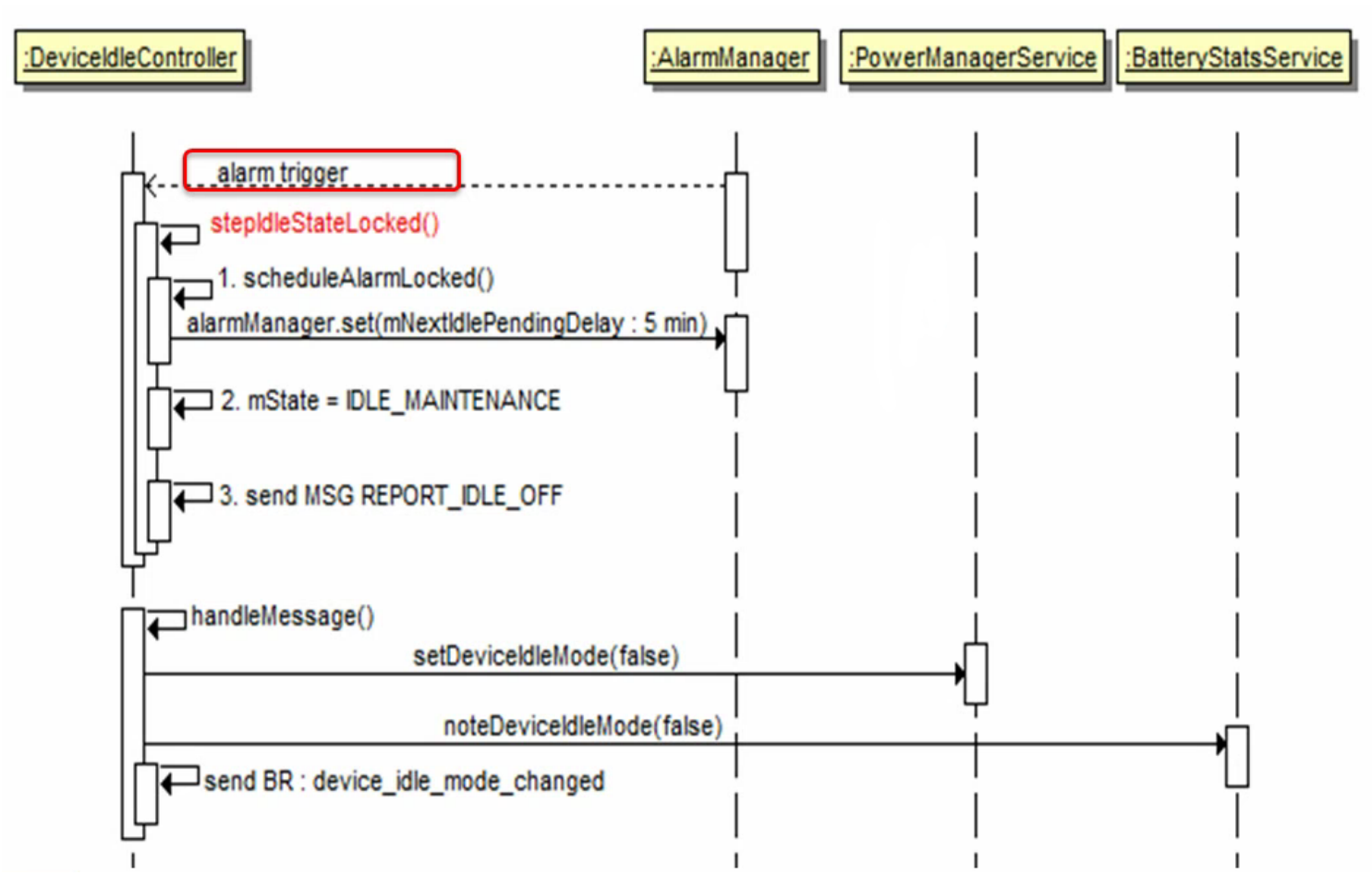
状态切换剖析之IDLE-->IDLE_MAINTENANCE

Posted by Cheson on April 10, 2017

此篇为Doze模式中最后一个状态切换。当设备经过重重判断，辛苦等待，千辛万苦终于进入到了IDLE模式，此时系统执行了一系列的低功耗策略，以致很多应用的任务都被挂起。那么这些挂起的任务怎么处理？设备会定时唤醒一次，持续一段时间来处理之前挂起的任务，这个处理时期就是IDLE_MAINTENANCE状态阶段。

IDLE_MAINTENANCE是如何被触发的？在切换到IDLE状态时会通过scheduleAlarmLocked(mNextIdleDelay, true)来设置一个Alarm，到时之后唤醒系统依旧调用stepIdleStateLocked来进行从IDLE状态往IDLE_MAINTENANCE的切换。

```
case STATE_IDLE:
    // We have been idling long enough, now it is time to do some work.
    scheduleAlarmLocked(mNextIdlePendingDelay, false);
    if (DEBUG) Slog.d(TAG, "Moved from STATE_IDLE to STATE_IDLE_MAINTENANCE. " +
            "Next alarm in " + mNextIdlePendingDelay + " ms.");
    mNextIdlePendingDelay = Math.min(mConstants.MAX_IDLE_PENDING_TIMEOUT,
            (long)(mNextIdlePendingDelay * mConstants.IDLE_PENDING_FACTOR));
    mState = STATE_IDLE_MAINTENANCE;
    EventLogTags.writeDeviceIdle(mState, "step");
    mHandler.sendEmptyMessage(MSG_REPORT_IDLE_OFF);
    break;
```



依照国际惯例，在开始切换IDLE_MAINTENANCE状态时，会先schedule一个Alarm，这个Alarm是用来定时触发进行从IDLE_MAINTENANCE再次切换到IDLE的。

这里也涉及到了状态持续时间的动态变化问题。mNextIdlePendingDelay的值初始为IDLE_PENDING_TIMEOUT(5分钟)，每切换一次就会通过mNextIdlePendingDelay乘以

IDLE_PENDING_FACTOR（2）和MAX_IDLE_PENDING_TIMEOUT（10分钟）取最小值。

```
IDLE_PENDING_TIMEOUT = mParser.getLong(KEY_IDLE_PENDING_TIMEOUT,
            !COMPRESS_TIME ? 5 * 60 * 1000L : 30 * 1000L);
MAX_IDLE_PENDING_TIMEOUT = mParser.getLong(KEY_MAX_IDLE_PENDING_TIMEOUT,
            !COMPRESS_TIME ? 10 * 60 * 1000L : 60 * 1000L);
IDLE_PENDING_FACTOR = mParser.getFloat(KEY_IDLE_PENDING_FACTOR,
            2f);
```

所以从结果来看第一次IDLE_MAINTENANCE会持续5分钟，后面都是10分钟。然后是将当前状态改为IDLE_MAINTENANCE接下来就是和进入IDLE时一个相反的操作，发送一个MSG_REPORT_IDLE_OFF的message以及发送一个广播来通知各个接收器。在收到该消息之后做的处理也是和IDLE状态下相反的操作。

```
case MSG_REPORT_IDLE_OFF: {
        EventLogTags.writeDeviceIdleOffStart("unknown");
        mLocalPowerManager.setDeviceIdleMode(false);
        try {
            mNetworkPolicyManager.setDeviceIdleMode(false);
            mBatteryStats.noteDeviceIdleMode(false, null, Process.myUid());
        } catch (RemoteException e) {
        }
        getContext().sendBroadcastAsUser(mIdleIntent, UserHandle.ALL);
        EventLogTags.writeDeviceIdleOffComplete();
    } break;
```

各个服务中所做的事情详见后面专门介绍Doze功耗策略的篇章。这里需要说明的一点时，在IDLE阶段时，通过setIdleUntil设置了Alarm，之后的Alarm就会被挂起添加到pending list中，那么这些Alarm后面又是如何被处理的呢？当进入IDLE_MAINTENANCE时，也就是这个这个Alarm被触发的时候，在AlarmManagerService中调用triggerAlarmsLocked，判断触发的是之前的Alarm，然后就restore之前被pending的其他Alarm。

```
if (mPendingIdleUntil == alarm) {
    mPendingIdleUntil = null;
    rebatchAllAlarmsLocked(false);
    restorePendingWhileIdleAlarmsLocked();
}
```

这篇要介绍的状态切换的内容也就到此为止，至此就讲完了Doze模式下所有的状态切换的流程和每个状态中所做的事。

PREVIOUS
ANDROID电源管理之DOZE模式专题系列（七）
(/2017/04/07/PM_DOZE_LOCATING_TO_IDLE/)

NEXT
ANDROID电源管理之DOZE模式专题系列（十）
(/2017/04/10/PM_DOZE_SHIELD_WAKELOCK/)

FEATURED TAGS (/tags/)

前端 (/tags/#前端) Android (/tags/#Android) frameworks (/tags/#frameworks) AlarmManager (/tags/#AlarmManager) Performance (/tags/#Performance) systrace (/tags/#systrace) PowerManager (/tags/#PowerManager) Wakelock (/tags/#Wakelock) Guitar (/tags/#Guitar) 民谣 (/tags/#民谣) 赵雷 (/tags/#赵雷) Doze (/tags/#Doze) Android Performance Patterns (/tags/#Android Performance Patterns)

FRIENDS

待遇见志同道合的你 (https://github.com)　小明 (http://www.betterming.cn)

(https://twitter.com/chendongqi)

(https://www.zhihu.com/people/chendongqi)

(http://weibo.com/chendongqi) (https://www.facebook.com/chendongqi)

(https://github.com/chendongqi)

(https://www.linkedin.com/in/firstname-lastname-idxxxx)

Theme by Cheson (https://github.com/chendongqi/blog) | Star 1