原文链接：

http://developer.android.com/intl/zh-CN/training/basics/activity-lifecycle/stopping.html

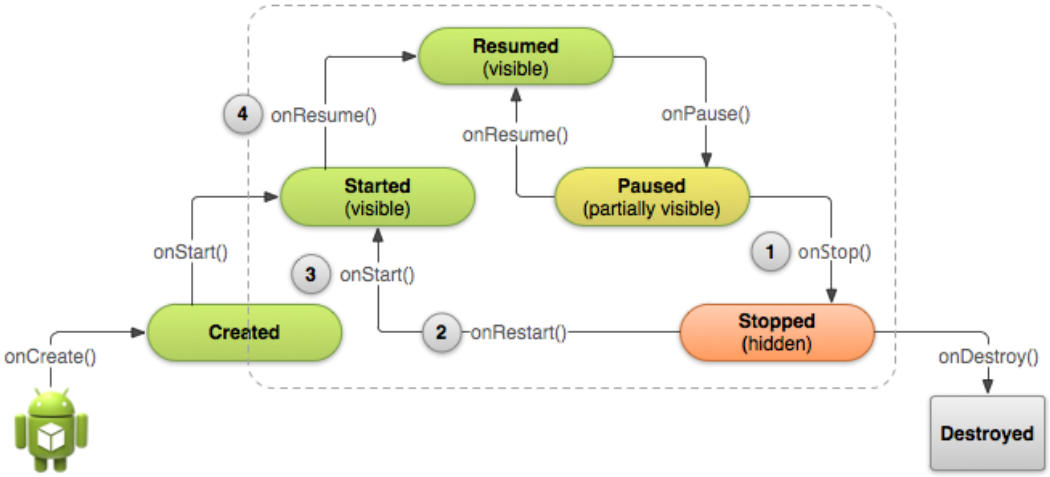
Stopping and Restarting an Activity

在 activity 生命周期中，恰当的停止与重启 activity 是很重要的，这样能确保用户感知到程序的存在并不会丢失他们的进度。在下面一些关键的场景中会涉及到停止与重启：

- 用户打开“最近使用的程序(Recent Apps)”的菜单并从当前 app 切换到另外一个 app，这个时候先前的 app 是被停止的。如果用 户通过“主屏幕加载图标（Home screen launcher icon”或“最近使用的程序(Recent Apps)”重新回到这个 app，则 activity 会重启。
- 用户在当前的 app 里面执行启动一个新的 activity 的操作时，当前 activity 会在第二个 activity 被创建后停止。如果用户点 击 back 按钮，之前的 activtiy 会被重启。
- 用户在运行 app 时接受到一个来电通话。

Activity 类提供了 onStop()方法与 onRestart()方法用于在 activity 停止与重启时进行调用。和暂停状态时部分阻塞用户接口不同，停止状态时 UI 不可见并且用户的焦点转移到另一个 activity 中。

注意： 因为系统在 Activity 停止时会在内存中保存了 Activity 实例，所以很多时候不需要实现 onStop()方法与 onRestart()方法，甚至是 onStart()方法。 因为大多数的 Activity 相对比较简单，Activity 会自动正常停止与重启，只需要使用 onPause()来停止正在运行的动作并断开系统资源链接。



如图 1 所示：当用户离开 Activity 时系统会调用 onStop()来停止 Activity (1)。这个时候如果用户返回，系统会调用 onRestart()(2)， 紧接着会调用 onStart()(3)与 onResume()(4)。需要注意的是：无论什么原因导致 Activity 停止，系统总是会在 onStop()之前调用 onPause()方法。

停止 Activity

当 activity 调用 onStop()方法时，该 activity 不再可见并且应该释放所有不再需要的资源。一旦 activity 停止了，系统可能会摧毁 activity 的实例以回收内存，甚至，系统会不执行 activity 的 onDestroy()回调方法而直接杀死你的 app 进程，因此需要使用 onStop()来释放资源，从而避免内存泄漏。

尽管 onPause()方法是在 onStop()之前调用，通常应该使用 onStop()来执行 CPU 密集型的关闭操作，例如把数据写入数据库。

例如，下面是一个在 onStop()的方法里面保存笔记草稿到永久存储介质的示例:

```
1.  @Override
2.  protected void onStop() {
3.      super.onStop();  // Always call the superclass method first
4.
5.      // Save the note's current draft, because the activity is stopping
6.      // and we want to be sure the current note progress isn't lost.
7.      ContentValues values = new ContentValues();
8.      values.put(NotePad.Notes.COLUMN_NAME_NOTE, getCurrentNoteText());
9.      values.put(NotePad.Notes.COLUMN_NAME_TITLE, getCurrentNoteTitle());
10.
11.     getContentResolver().update(
12.             mUri,    // The URI for the note to update.
13.             values,  // The map of column names and new values to apply to them.
14.             null,    // No SELECT criteria are used.
15.             null     // No WHERE columns are used.
16.             );
17. }
```

当 Activity 已经停止，Activity 对象会保存在内存中，并且在 Activity 恢复的时候被重新调用到。不需要在恢复到 Resumed state 状态前重新初始化那些被保存在内存中的组件。系统同样保存了每一个在布局中的视图的当前状态，如果用户在 EditText 组件中输入了文本，也会被保存，因此不需要保存与恢复它。

注意： 即使系统会在 Activity 停止的时候销毁这个 Activity，系统仍然会保存视图对象（如文本编辑框中的文本)到一个 Bundle 中,并且在用户返回这个 Activity 时恢复他们(下一节会介绍在 Activity 销毁与重建时如何使用 Bundle 来保存其他数据的状态)。

启动与重启 Activity

当 Activity 从 Stopped 状态回到前台时会调用 onRestart(),系统再调用 onStart()方法,onStart()方法在每次 Activity 可见时都会被调用。onRestart()方法则是只在 Activity 从 stopped 状态恢复时才会被调用，因此可以使用它来执行一些特殊的恢复工作，请注意之前是被 stop 而不是 destrory。

使用 onRestart()来恢复 Activity 状态并不常见，因此对于这个方法如何使用没有指导说明。但是，由于 onStop()方法要做清除所有 Activity 资源的操作，在重新启动 Activtiy 时需要重新实例化被清除的资源，同样，在 Activity 第一次创建时要实例化那些资源。因为系统会在创建 Activity 与从停止状态重启 Activity 时都会调用 onStart()，应该使用 onStart()作为 onStop()所对应的方法。

例如：因为用户很可能在回到 Activity 之前需要过一段时间，所以 onStart()方法是一个比较好的用来验证某些必须的功能是否已经准备好的地方。

```
1.  @Override
2.  protected void onStart() {
3.      super.onStart();  // Always call the superclass method first
4.
5.      // The activity is either being restarted or started for the first time
6.      // so this is where we should make sure that GPS is enabled
7.      LocationManager locationManager =
8.              (LocationManager) getSystemService(Context.LOCATION_SERVICE);
9.      boolean gpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDE
    R);
10.
11.     if (gpsEnabled) {
12.         // Create a dialog here that requests the user to enable GPS, and use an inten
    t
13.         // with the android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS action
14.         // to take the user to the Settings screen to enable GPS when they click "OK"
15.     }
16. }
17.
18. @Override
19. protected void onRestart() {
```

```
20.     super.onRestart();  // Always call the superclass method first
21.
22.     // Activity being restarted from stopped state
23. }
```

当系统 Destory 一个 Activity，它会为 Activity 调用 onDestroy()方法。由于会在 onStop 方法里面做释放资源的操作，而 onDestory 方法则是最后去清除那些可能导致内存泄漏的地方，因此需要确保那些线程都被 Destroy 并且所有的操作都被停止。

文章来源：http://wiki.eoe.cn/page/Stopping_and_Restarting_an_Activity