**Android Service:**

链接地址：<http://www.cnblogs.com/newcj/archive/2011/05/30/2061370.html>

**使用原因：** 让Activity可以对后台任务进行控制

1.Service主要用于在后台处理一些耗时的逻辑，或者去执行某些需要长期运行的任务。必要的时候我们甚至可以在程序退出的情况下，让Service在后台继续保持运行状态。

2.Thread相当于是主线程的一条分支，主线程kill了，thread也被kill了。所以不能在不同的activity对同一thread进行控制，可以在 Service 里面创建、运行并控制该 Thread，Thread就像寄生虫寄生在Service的生命周期中。Service其实是运行在主线程里的

3.所以Service和Activity一样不能执行耗时操作，耗时操作在Tread中实现。

Service是没有视图的，相当于知道逻辑处理，通过获得Service中的binder实例就可以随意让service执行activity所需要的功能了。

**Service和Thread的区别：**

**Service：**在主线程中不能执行耗时操作，但可以不受Activity的生命周期的影响，实现后台运行

**Thread：**子线程，在里面执行耗时操作，但其生命周期受主线程影响

所以可以在 Service 里面创建、运行并控制该 Thread

**定义：**1.服务不是线程：

可以把Service看成一段代码，用来在后台做些事情，仅此而已。至于这段代码在哪里运行，完全是取决于你自己的喜好。你的local service，如果不建立worker thread，仍然是在你的应用进程的主线程即UI线程里运行。如果你不想阻塞UI线程，你就建一个worker thread。

2.服务不是一个单独的过程,作为应用程序的一部分,

3.Service只是从逻辑上表示一个用来执行"longer-running"的"后台"任务或功能的"application component"。

**什么时候选择local service（即不指定额外的进程），什么时候选择remote service（额外的进程）:**

**local service:**短期运行但仍然超出activity活动周期的任.（发送短信），真正结束是在**主进程（main）**被退出时

**remote service:**把真的需要长期运行的service（例如IM（即时通讯）之类）放在单独的进程里

**使用：**

链接地址：<http://blog.csdn.net/guolin_blog/article/details/11952435>

启动Service的方法和启动Activity很类似，都需要借助Intent来实现

1. 新建一个类继承自Service，并重写父类的onCreate()、onStartCommand()和onDestroy()方法，**onBind()方法**是Service中唯一的一个抽象方法，必须在子类实现。

2..每一个Service都必须在AndroidManifest.xml中注册

3.在Activity中通过一个Intent对象，并调用startService()方法来启动MyService，调用stopService()方法来停止MyService（只是Activity通知了Service一下：“你可以启动了。”Service是没有界面的）

Intent startIntent = new Intent(this, MyService.class);

        startService(startIntent);

服务本身实际上是非常简单的，提供了2个主要功能：

1. 后台运行的能力，告诉系统给服务安排工作，直到系统或某人明确停止它（直到被调用stopService，或自身的stopSelf方法）。

**startService（） 启动的服务**，主要用于启动一个服务执行后台任务，不进行通信。【create（）方法---创建和stop（）方法一次---停止，start（）方法多次---调用】

1. 使应用程序具有将其功能暴露于其他应用程序中的能力。这这相当于调用Context.bindService(),允许和服务进行交互。。

**bindService（） 启动的服务**，该方法启动的服务要进行通信：Activity使用bindService（）绑定到该Service或unbindService（）解除绑定）

**Service和Activity通信：**

在Activity中可以指定让Service去执行什么任务，而不是仅仅告诉它可以启动了。

onBind()方法其实就是用于和Activity建立关联的

**步骤：**

1. 在Service类中建立一个内部类（MyBinder）继承自Binder类
2. 在内部类中编写需要做的任务
3. 在Service中创建一个内部类的实例（new出来），用于onBind（）方法的返回
4. 在Activity中建立与Service的关联：

①创建了一个ServiceConnection的匿名类，在里面重写了onServiceConnected()方法和onServiceDisconnected()方法，这两个方法分别会在Activity与Service建立关联和解除关联的时候调用。

②在onServiceConnected()方法中，我们又通过向下转型（通过service得到binder）得到了我们建立的内部类（MyBinder的实例）

③有个MyBinder实例，在Activity中就可以调用MyBinder中的任何public方法

1. 调用bindService(Intent，ServiceConnection的实例，标志位)方法将Activity和Service进行绑定，就可以使Service执行ServiceConnected()中调用的方法了。

另外需要注意，任何一个Service在整个应用程序范围内都是通用的，即MyService不仅可以和MainActivity建立关联，还可以和任何一个Activity建立关联，而且在建立关联时它们都可以获取到相同的MyBinder实例。

一个Service必须要在既没有和任何Activity关联（unbindService（））又处理停止状态（stopService（））的时候才会被销毁