在**Android**系统中，每一个应用程序都是由一些Activity和Service组成的，这些Activity和Service有可能运行在同一个进程中，也有可能运行在不同的进程中。不在同一个进程的Activity或者Service的通信就采用Binder进程间通信机制了。

binder通信是一种client-server的通信结构，

1.从表面上来看，是client通过获得一个server的代理接口，对server进行直接调用；

2.实际上，代理接口中定义的方法与server中定义的方法是一一对应的；

3.client调用某个代理接口中的方法时，代理接口的方法会将client传递的参数打包成为Parcel对象；

4.代理接口将该Parcel发送给内核中的binder driver.

5.server会读取binder driver中的请求数据，如果是发送给自己的，解包Parcel对象，处理并将结果返回；

6.整个的调用过程是一个同步过程，在server处理的时候，client会block住。

任何service在被使用之前，均要向SM(Service Manager)注册，同时客户端需要访问某个service时，应该首先向SM查询是否存在该服务。如果SM存在这个service，那么会将该service的handle返回给client，handle是每个service的唯一标识符。

Service Manager主要工作：

1.初始化binder，打开/dev/binder设备；在内存中为binder映射128K字节空间；

2.指定SM对应的代理binder的handle为0，当client尝试与SM通信时，需要创建一个handle为0的代理binder，这里的代理binder其实就是代理接口；

3.通知binder driver(BD)使SM成为BD的context manager；

4.维护一个死循环，在这个死循环中，不停地去读内核中binder driver，查看是否有可读的内容；即是否有对service的操作要求, 如果有，则调用svcmgr_handler回调来处理请求的操作。

5.SM维护了一个svclist列表来存储service的信息。