1. 为什么会出现Binder？

Android底层是由Linux Kernel支持，进程间不能直接通讯，主要有下面的原因

1. 进程的地址空间都是相互独立的，A进程不知道B进程会有什么数据
2. 基于安全性的考虑，也不该允许两个进程间相互访问。

但是两个进程间由于某些原因必不可少的需要相互通信，进行数据的交互，这就需要有一个安全又可用的跨进程通讯机制。

传统的Linux通讯方式如IPC、管道，效率比较低，数据需要多次交换，不适合Android的通讯。如IPC通信需要经过两次数据拷贝，一次为进程到内核，一次为内核到进程。

1. Binder是什么？

Binder是Android通信的框架，负责跨进程通讯。在Android中Binder主要可以分为三层，Java层，Native层，驱动层。Binder通过驱动层访问Binder的虚拟驱动/dev/binder实现跨进程的数据交换，如何访问则由Native层控制，native层是一个C/S架构的实现，通过代理模式实现。Java层是建立在Native层基础上，供Java代码调用，最终都会调用Native层的代码。

1. Service与Binder之间的联系

Android中存在许多系统的服务，如ActivityManagerService, WindowManagerService, PackageManagerService等，这些Service对应于Binder的服务端，与客户端的ActivityManager、WindowManager、PackageManager通过Binder进行通信。

这些系统的Service都是通过ServiceManager来进行管理，而Servicemanager自身也是基于Binder进行实现的

1. 实现案例

Java层:

naïve层Binder实现案例

可以尝试用parcel传递超过1MB的数据