

存只在性能方面优于 Binder，其他方面都劣于 Binder，这些就是 Android 要使用 Binder 来进行进程间通信的原因，当然系统中并不是所有进程通信都采用了 Binder，而是根据场景选择最合适的，比如 Zygote 进程与 AMS 通信使用的是 Socket，Kill Process 采用的是信号。

5.1.3 学习 Binder 的原因

Binder 机制在 Android 中的地位举足轻重，我们需要掌握的很多原理都和 Binder 有关。

- (1) 系统中的各个进程是如何通信的。
- (2) Android 系统启动过程。
- (3) AMS、PMS 的原理。
- (4) 四大组件的原理，比如 Activity 是如何启动的。
- (5) 插件化原理。
- (6) 系统服务的客户端和服务端是如何通信的（比如 MediaPlayer 和 MeidaPlayerService）。

以上只是列举了一小部分，简单来讲，比如系统在启动时，SystemServer 进程启动后会创建 Binder 线程池，其目的是通过 Binder 可以使在 SystemServer 进程中的服务能够和其他进程间通信。再比如我们常说的 AMS、PMS 都是基于 Binder 来实现的，拿 PMS 来说，PMS 运行在 SystemServer 进程，如果它想要和 DefaultContainerService 通信（用于检查和复制可移动文件的系统服务），就需要通过 Binder 来实现，因为 DefaultContainerService 运行在 com.android.defcontainer 进程。

还有一个比较常见的 C/S 架构间通信的问题，客户端的 MediaPlayer 和服务端的 MeidaPlayerService 不是运行在一个进程中的，同样需要 Binder 来实现通信。

可以说 Binder 机制是掌握系统底层原理的基石。根据 Android 系统的分层，Binder 机制主要分为 Java Binder、Native Binder、Kernel Binder，如图 5-5 所示。