2.3.1Parcelable和Serializable

Parcelable和Serializable都能实现序列化并且都可用于Intent间的数据传递。

Serializable是Java的序列化接口，使用起来简单但是开销大，序列化和反序列化过程需要大量I/O操作。而Parcelable是Android中的序列化方式，效率高，所有更适合，但使用稍微麻烦。

2.3.2Binder

Binder是Android中的一个类，它实现了IBinder接口，从ipc角度来说，它是Android中的一种跨进程通讯方式。从Android framework角度 ，Binder是ServiceManager连接各个Manager（ActivityManager、WindowManager等等）的桥梁。

Android开发中，binder主要用在Service中，包括AIDL和Messenger,其中Service中的Binder不涉及进程间通讯，所有简单。

//Book.java

Public class Book implements Parcelable{

Public int bookId;

Public

}

//Book.aidl

parcelable Book;

//IBookManager.aidl

Interface IBookManaget{

List<Book> getBookList();

Void addBook(in Book book);

}

根据IBookManager.aidl系统会自动生成IBookManager.java类，它继承了Iinterface接口

它其中的内部类Stub就是一个Binder类，而Stub的内部代理类Proxy就是处理进程通讯的

//IBookManager.java类

1. asInterface 将服务端的Binder对象转换成客户端的AIDL对象，这种转换区分进程，当他们在同一个进程时返回的是Stub本身，否则返回的是Stub.Proxy对象
2. asBinder 返回当前Binder对象
3. onTransact 这个方法运行在服务端中的Binder线程池中，当客户端发出请求时，系统底层会交由此方法处理。在该方法中通过code判断，去执行不同的目标方法，执行结束后将结果写入reply中。
4. getBookList /addBook 该方法运行在客户端，客户端调用此方法将服务端的数据返回回来。