一、Android中常用的五种布局？

1) FrameLayout 框架布局,放入其中的所有元素都被放置在最左上的区域，而且无法为这些元素指定一个确切的位置,下一个子元素会重叠覆盖上一个子元素，适合浏览单张图片。

2) LinearLayout 线性布局,是应用程序中最常用的布局方式，主要提供控件水平或者垂直排列的模型，每个子组件都是以垂直或水平的方式来定位.(默认是垂直)

3) AbsoluteLayout 绝对定位布局,采用坐标轴的方式定位组件，左上角是（0，0）点，往右x轴递增，往下Y轴递增,组件定位属性为android:layout\_x 和 android:layout\_y来确定坐标。

4) RelativeLayout 相对布局,根据另外一个组件或是顶层父组件来确定下一个组件的位置。和CSS里面的类似。

5) TableLayout 表格布局,类似Html里的Table.使用TableRow来布局，其中TableRow代表一行，TableRow的每一个视图组件代表一个单元格。

二、进程与线程之间的联系与区别？

进程是系统进行资源分配和调度的一个独立单位，线程是CPU调度和分派的基本单位

进程和线程的关系：

（1）一个线程只能属于一个进程，而一个进程可以有多个线程，但至少有一个线程。

（2）资源分配给进程，同一进程的所有线程共享该进程的所有资源。

（3）线程在执行过程中，需要协作同步。不同进程的线程间要利用消息通信的办法实现同步。

（4）处理机分给线程，即真正在处理机上运行的是线程。

（5）线程是指进程内的一个执行单元，也是进程内的可调度实体。

线程与进程的区别：

（1）调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位。

（2）并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可以并发执行。

（3）拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源。

（4）系统开销：在创建或撤销进程的时候，由于系统都要为之分配和回收资源，导致系统的明显大于创建或撤销线程时的开销。

但进程有独立的地址空间，进程崩溃后，在保护模式下不会对其他的进程产生影响，而线程只是一个进程中的不同的执行路径。

线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，

所以多进程的程序要比多线程的程序健壮，但是在进程切换时，耗费的资源较大，效率要差些。

三、Fragment和Activity的区别？

fragment是3.0以后的东西，为了在低版本中使用fragment就要用到android-support-v4.jar兼容包,而fragmentActivity就是这个兼容包里面的，

它提供了操作fragment的一些方法，其功能跟3.0及以后的版本的Activity的功能一样。

1、fragmentactivity 继承自activity，用来解决android3.0 之前没有fragment的api，所以在使用的时候需要导入support包，

同时继承fragmentActivity，这样在activity中就能嵌入fragment来实现你想要的布局效果。?

2、当然3.0之后你就可以直接继承自Activity，并且在其中嵌入使用fragment了。

3、获得Manager的方式也不同?

3.0以下：getSupportFragmentManager()

3.0以上：getFragmentManager()

四，单线程模型中Message、MessageQueue、Looper之间的关系？

Andriod提供了Handler和Looper来满足线程间的通信.Handler先进先出原则.

Looper类用来管理特定线程内对象之间的消息交换(Message Exchange).

Message Queue(消息队列)：用来存放通过Handler发布的消息，通常附属于某一个创建它的线程，可以通过Looper.myQueue()得到当前线程的消息队列.

Handler：可以发布或者处理一个消息或者操作一个Runnable，通过Handler发布消息，消息将只会发送到与它关联的消息队列，

然也只能处理该消息队列中的消息.

Looper：是Handler和消息队列之间通讯桥梁，程序组件首先通过Handler把消息传递给Looper，Looper把消息放入队列。

Looper也把消息队列里的消息广播给所有的Handler：Handler接受到消息后调用handleMessage进行处理.