### 源码分析相关面试题

* [Volley源码分析](http://www.jianshu.com/p/ec3dc92df581)
* [注解框架实现原理](http://www.jianshu.com/p/20da6d6389e1)
* [okhttp3.0源码分析](http://www.jianshu.com/p/9ed2c2f2a52c)
* [onSaveInstanceState源码分析](http://www.jianshu.com/p/cbf9c3557d64)
* [静默安装和源码编译](http://www.jianshu.com/p/2211a5b3c37f)

### Activity相关面试题

* [保存Activity的状态](http://www.jianshu.com/p/cbf9c3557d64)
* [深刻剖析activity启动模式(一)](http://www.jianshu.com/p/b33fd8c550bf)
* [深刻剖析activity启动模式(二)](http://www.jianshu.com/p/e1ea9e542112)
* [深刻剖析activity启动模式(三)](http://www.jianshu.com/p/d13e3d552d4b)
* [Activity Task和Process之间的关系](http://www.jianshu.com/p/d13e3d552d4b)
* [service里面startActivity抛异常？activity不会](http://www.jianshu.com/p/16e880ceb3a4)

### 与XMPP相关面试题

* [XMPP协议优缺点](http://www.jianshu.com/p/2c04ac3c526a)
* [极光消息推送原理](http://www.jianshu.com/p/d88dc66908cf)

### 与性能优化相关面试题

* [内存泄漏和内存溢出区别](http://www.jianshu.com/p/5dd645b05c76)
* [UI优化和线程池实现原理](http://www.jianshu.com/p/c22398f8587f)
* [代码优化](http://www.jianshu.com/p/ebd41eab90df)
* [内存性能分析](http://www.jianshu.com/p/2665c31b9c2f)
* [内存泄漏检测](http://www.jianshu.com/p/1514c7804a06)
* [App启动优化](http://www.jianshu.com/p/f0f73fefdd43)
* [与IPC机制相关面试题](http://www.jianshu.com/p/de4793a4c2d0)

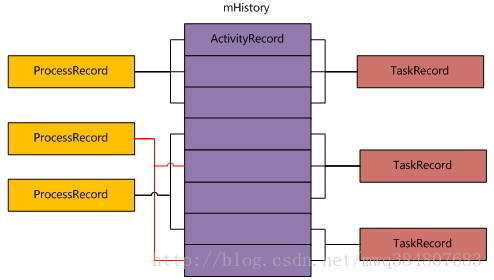
### 与登录相关面试题

* [oauth认证协议原理](http://www.jianshu.com/p/2a6ecbf8d49d)
* [token产生的意义](http://www.jianshu.com/p/9b7ce2d6c195)
* [微信扫一扫实现原理](http://www.jianshu.com/p/a9d1f21bd5e0)

### 与开发相关面试题

* [迭代开发的时候如何向前兼容新旧接口](http://www.jianshu.com/p/cbecadec98de)
* [手把手教你如何解决as jar包冲突](http://www.jianshu.com/p/30fdc391289c)
* [context的原理分析](http://www.jianshu.com/p/2706c13a1769)
* [解决ViewPager.setCurrentItem中间很多页面切换方案](http://www.jianshu.com/p/38ab6d856b56)
* [字体适配](http://www.jianshu.com/p/33d499170e25)
* [软键盘顶出去解决方案](http://www.jianshu.com/p/640bac6f58ab)与人事相关面试题
* [人事面试宝典](http://www.jianshu.com/p/d61b553ff8c9)

AMS提供了一个ArrayList mHistory来管理所有的activity，activity在AMS中的形式是ActivityRecord，task在AMS中的形式为TaskRecord，进程在AMS中的管理形式为ProcessRecord。如下图所示



img

从图中我们可以看出如下几点规则：

1. 所有的ActivityRecord会被存储在mHistory管理；
2. 每个ActivityRecord会对应到一个TaskRecord，并且有着相同TaskRecord的ActivityRecord在mHistory中会处在连续的位置；
3. 同一个TaskRecord的Activity可能分别处于不同的进程中，每个Activity所处的进程跟task没有关系；

Activity启动时ActivityManagerService会为其生成对应的ActivityRecord记录，并将其加入到回退栈(back stack)中，另外也会将ActivityRecord记录加入到某个Task中。请记住，ActivityRecord，backstack，Task都是ActivityManagerService的对象，由ActivityManagerService进程负责维护，而不是由应用进程维护。

在回退栈里属于同一个task的ActivityRecord会放在一起，也会形成栈的结构，也就是说后启动的Activity对应的ActivityRecord会放在task的栈顶

执行adb shell dumpsys activity命令，发现有以下输出：

Task id #30  
 TaskRecord{7f2f34a #30 A=com.maweiqi.second U=0 sz=2}  
 Intent { flg=0x10000000 cmp=com.open.android.task1/.SecondActivity }  
 Hist #1: ActivityRecord{a0f9ded u0 com.open.android.task3/.OtherActivity t30}  
 Intent { flg=0x10400000 cmp=com.open.android.task3/.OtherActivity }  
 ProcessRecord{12090b5 27543:com.open.android.task3/u0a62}  
 Hist #0: ActivityRecord{1048af6 u0 com.open.android.task1/.SecondActivity t30}  
 Intent { flg=0x10000000 cmp=com.open.android.task1/.SecondActivity }  
 ProcessRecord{5bc013e 26035:com.open.android.task1/u0a59}  
  
 Task id #31  
 TaskRecord{dce52bb #31 A=com.open.android.task3 U=0 sz=1}  
 Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.open.android.task3/.MainActivity }  
 Hist #0: ActivityRecord{f9e58c5 u0 com.open.android.task3/.MainActivity t31}  
 Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.open.android.task3/.MainActivity }  
 ProcessRecord{12090b5 27543:com.open.android.task3/u0a62}  
  
 Task id #29  
 TaskRecord{5b063d8 #29 A=com.open.android.task1 U=0 sz=1}  
 Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.open.android.task1/.MainActivity }  
 Hist #0: ActivityRecord{689947d u0 com.open.android.task1/.MainActivity t29}  
 Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x10000000 cmp=com.open.android.task1/.MainActivity }  
 ProcessRecord{5bc013e 26035:com.open.android.task1/u0a59}  
  
Running activities (most recent first):  
 TaskRecord{7f2f34a #30 A=com.maweiqi.second U=0 sz=2}  
 Run #3: ActivityRecord{a0f9ded u0 com.open.android.task3/.OtherActivity t30}  
 TaskRecord{dce52bb #31 A=com.open.android.task3 U=0 sz=1}  
 Run #2: ActivityRecord{f9e58c5 u0 com.open.android.task3/.MainActivity t31}  
 TaskRecord{7f2f34a #30 A=com.maweiqi.second U=0 sz=2}  
 Run #1: ActivityRecord{1048af6 u0 com.open.android.task1/.SecondActivity t30}  
 TaskRecord{5b063d8 #29 A=com.open.android.task1 U=0 sz=1}  
 Run #0: ActivityRecord{689947d u0 com.open.android.task1/.MainActivity t29}  
  
mResumedActivity: ActivityRecord{a0f9ded u0 com.open.android.task3/.OtherActivity t30}

从图对应文字在对应adb输出，这几个基本概念大家估计就清楚了。

* 欢迎关注微信公众号,长期推荐技术文章和技术视频
* 微信公众号名称：Android干货程序员

img

img