1. **当App进程 attach 到AMS, AMS如何将App进程 与保存在 AMS里的表示进程相关的ProcessRecord对应起来？**

* AMS在fork一个进程来运行 apk程序时，AMS会将启动的进程的pid保存到对应的ProcessRecord中，并将该 ProcessRecord 记到到AMS中的 mPidsSelfLocked

final SparseArray<ProcessRecord> mPidsSelfLocked = new SparseArray<ProcessRecord>();

Process.ProcessStartResult startResult = Process.*start*(entryPoint,  
 app.processName, uid, uid, gids, debugFlags, mountExternal,  
 app.info.targetSdkVersion, app.info.seinfo, requiredAbi, instructionSet,  
 app.info.dataDir, entryPointArgs);

app.setPid(startResult.pid);

mPidsSelfLocked.put(startResult.pid, app);

* 当apk进程将IAppliationThread attach到AMS时，AMS通过Binder获得calling Pid. 然后根据获得的pid 从 AMS的 mPidsSelfLocked里获得Apk进程对应的ProcessRecord

**int** callingPid = Binder.*getCallingPid*();

attachApplicationLocked(thread, callingPid);

然后将APK attach过来的IAppliationThread保存到对应的ProcessRecord的thread中

这样AMS就通过 ProcessRecord与APK进程完全联系起来了

1. **ProcessRecord如何和ActivityRecord联系起来的呢？**

首先ProcessRecord是对一个进程的表示，而一个apk进程里可以运行多个activity, 所以他们应该是一对多的关系，即

Activity

Activity

Activity

ProcessRecord

在realStartActivityLocked 函数里

r.app = app;

**int** idx = app.activities.indexOf(r);  
**if** (idx < 0) {  
 app.activities.add(r);  
}