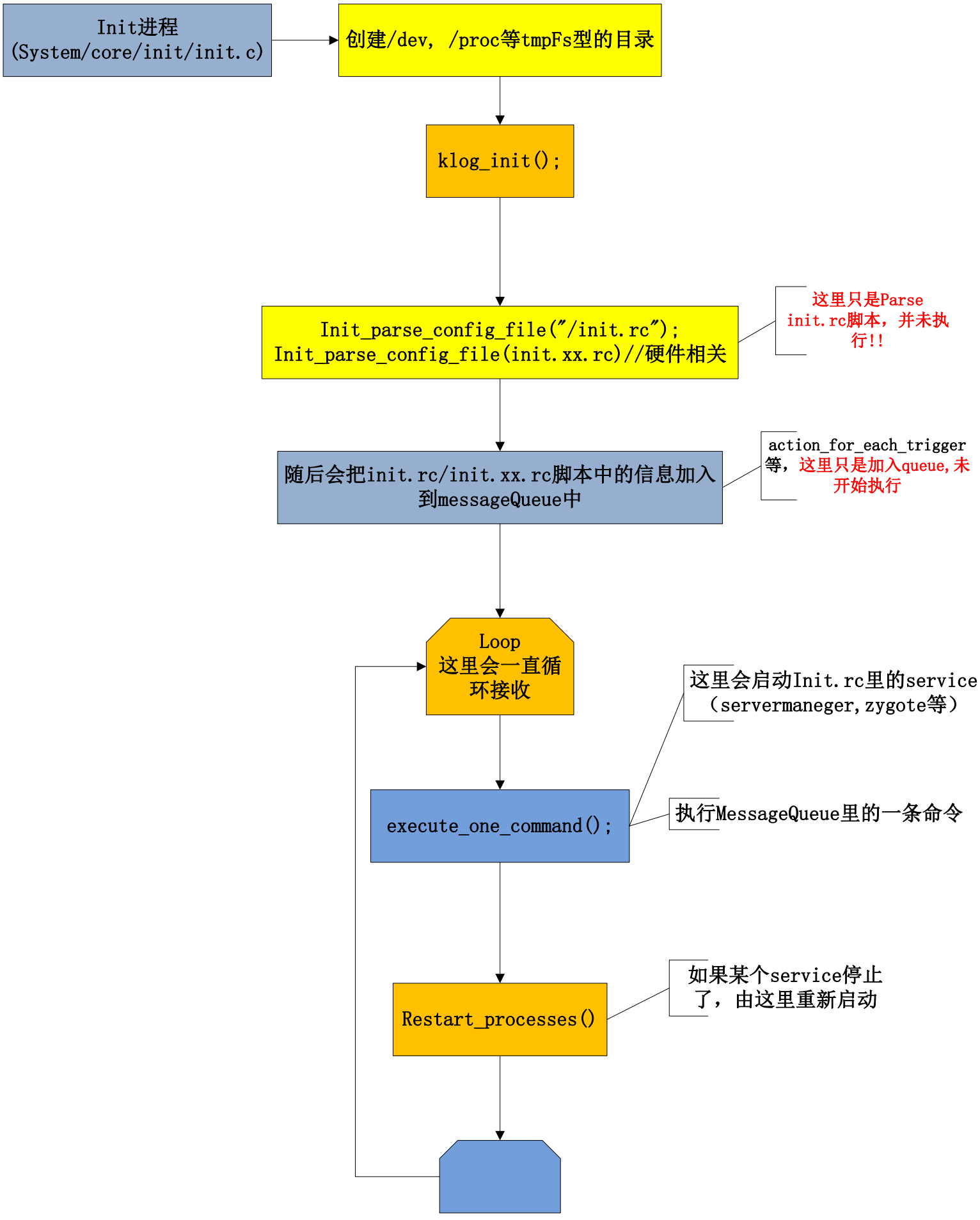
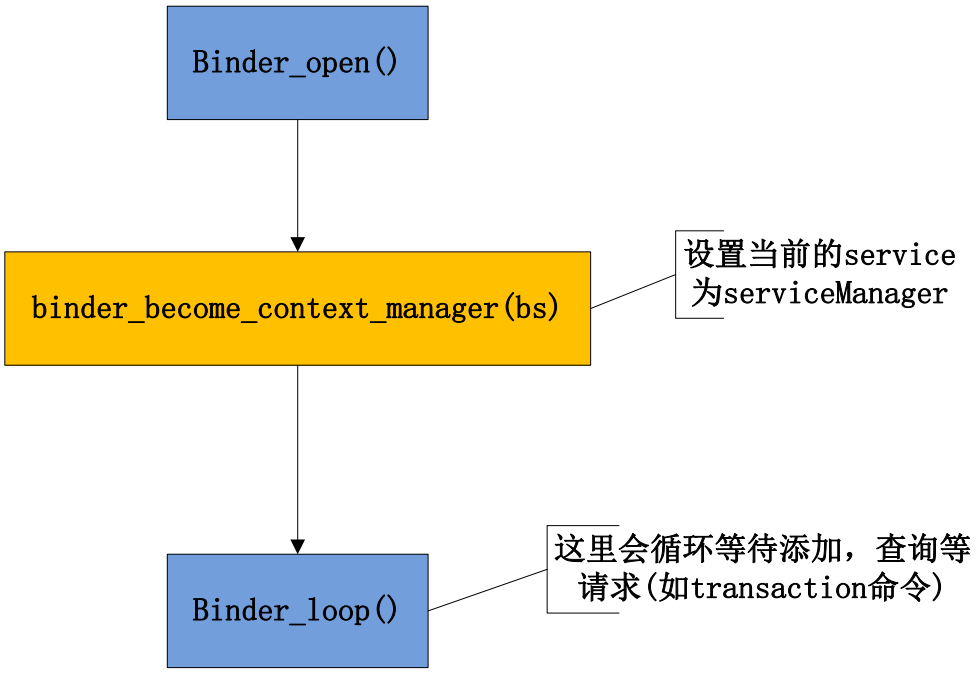
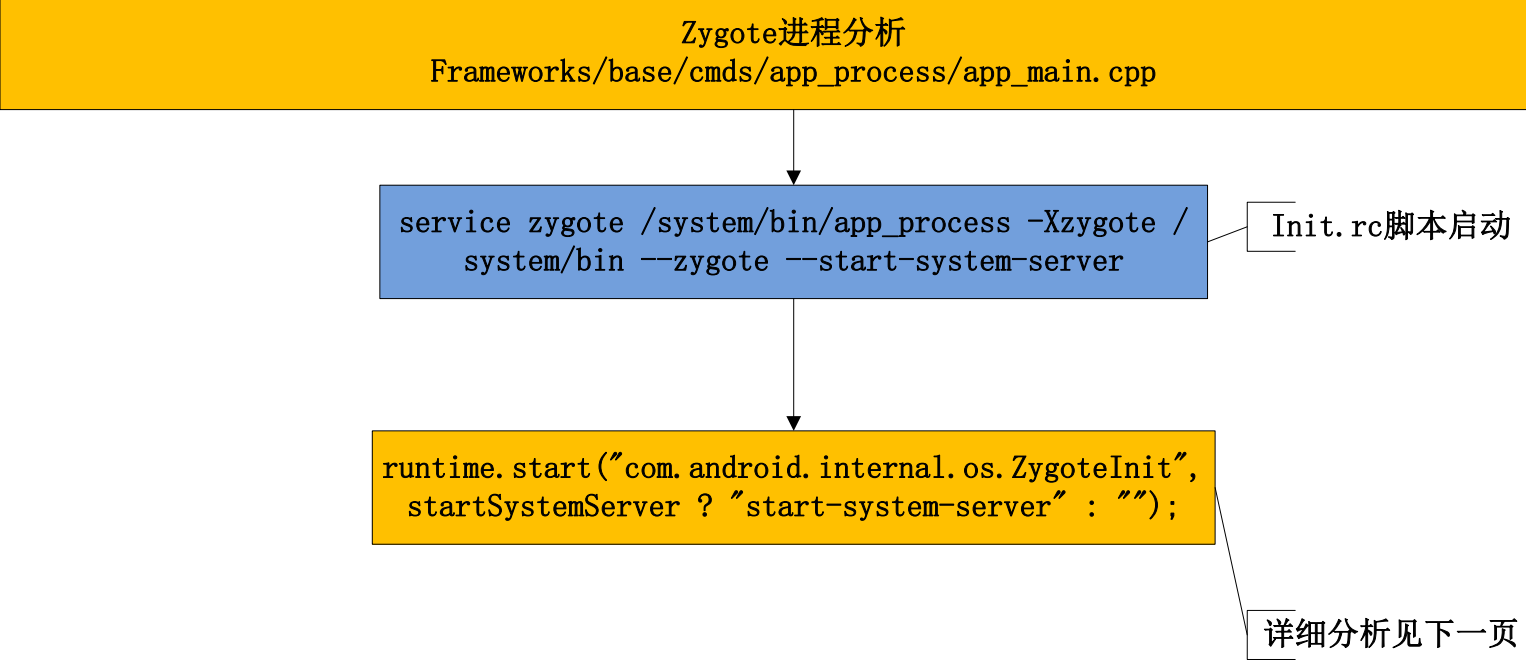
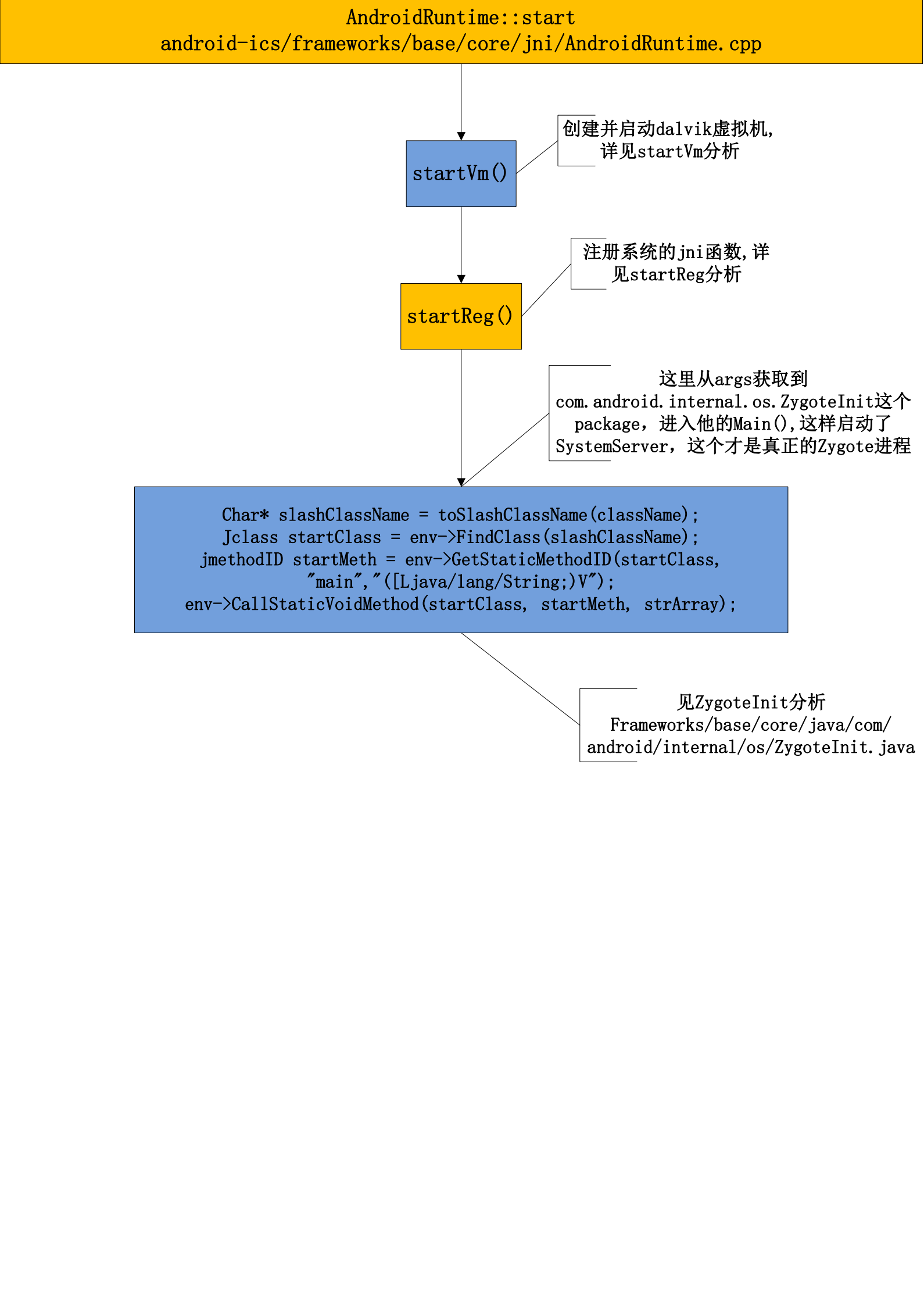
android系统启动流程

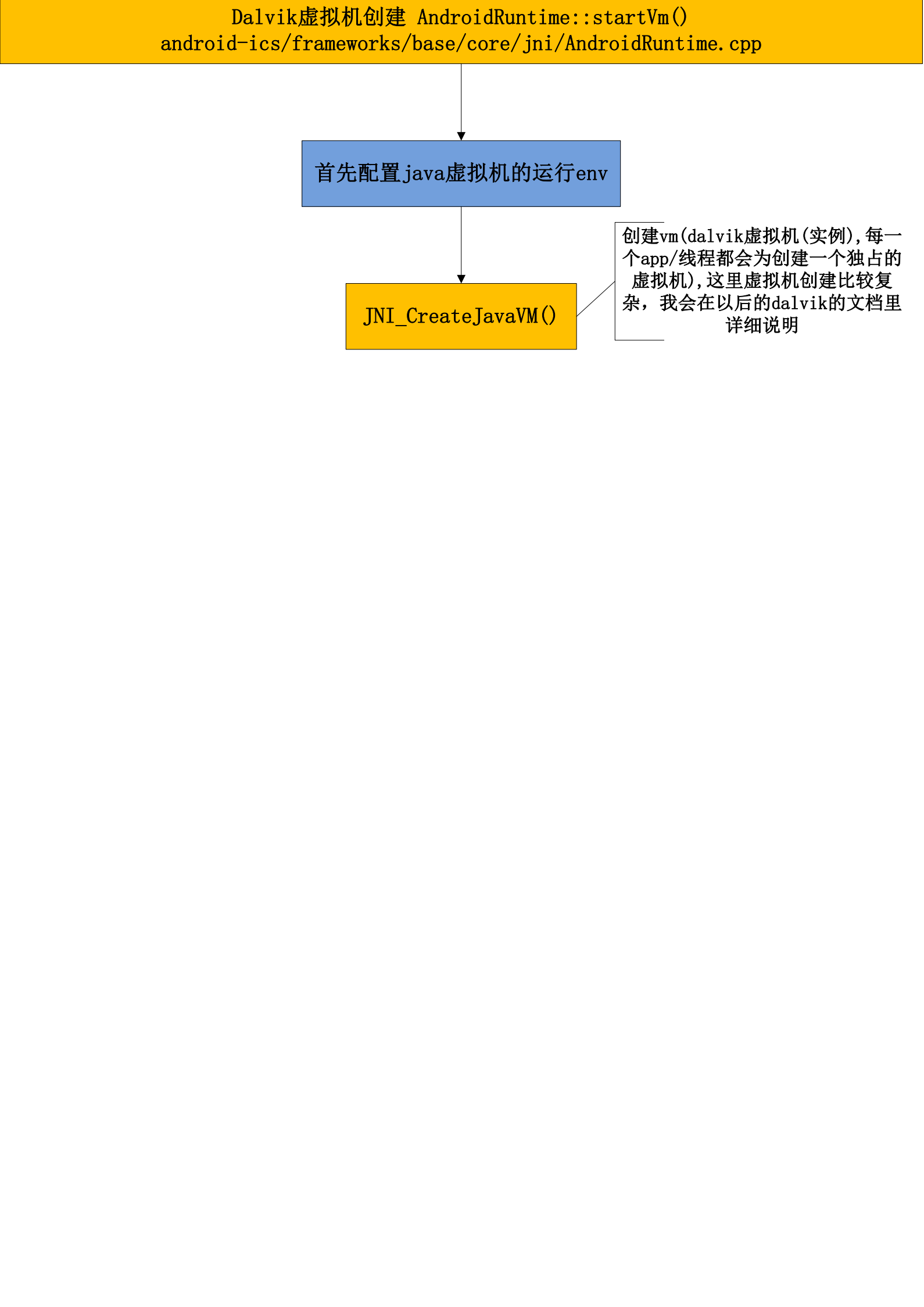


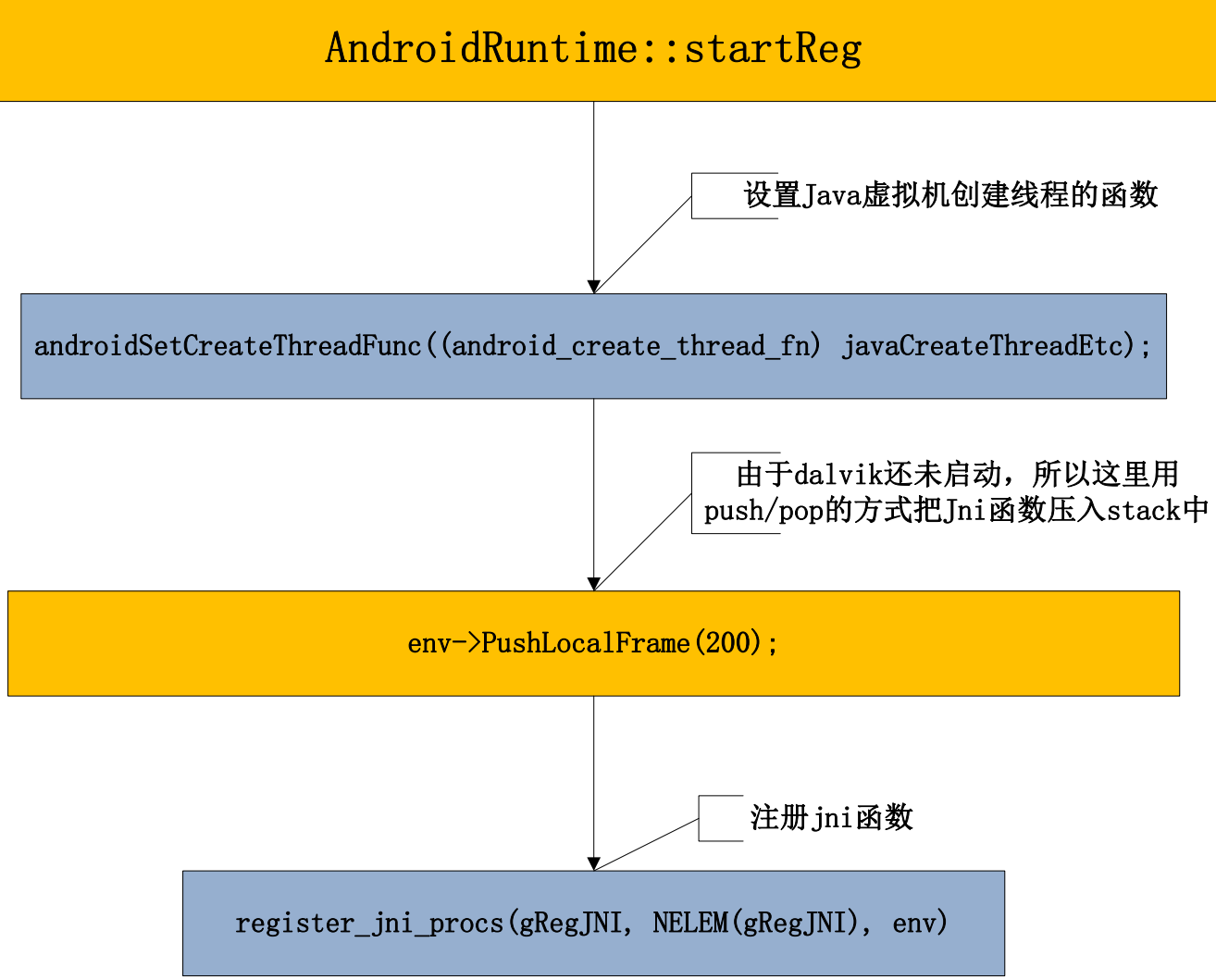
serviceManager服务创建
Frameworks/base/cmds/servicemanager/service_manager.c



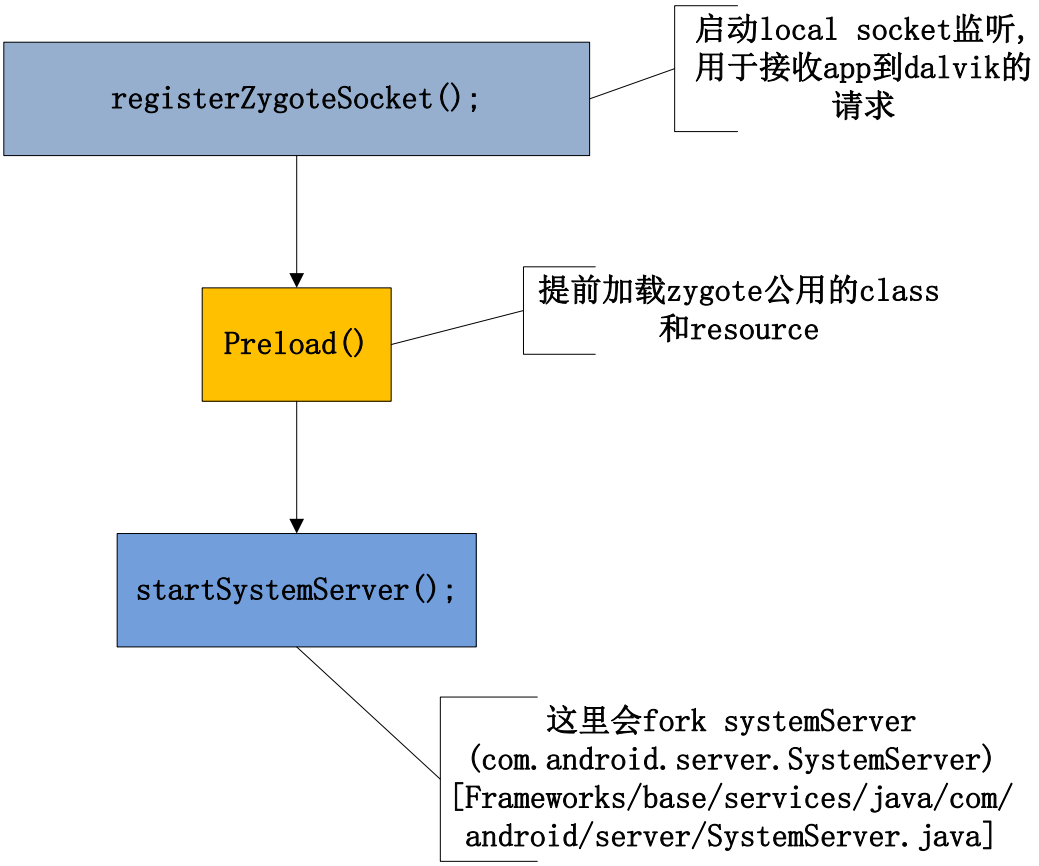

Zygote进程分析
Frameworks/base/cmds/app_process/app_main.cpp
service zygote /system/bin/app_process -Xzygote /
system/bin --zygote --start-system-server
Init.rc脚本启动
runtime.start("com.android.internal.os.ZygoteInit",
startSystemServer ? "start-system-server" : "");
详细分析见下一页

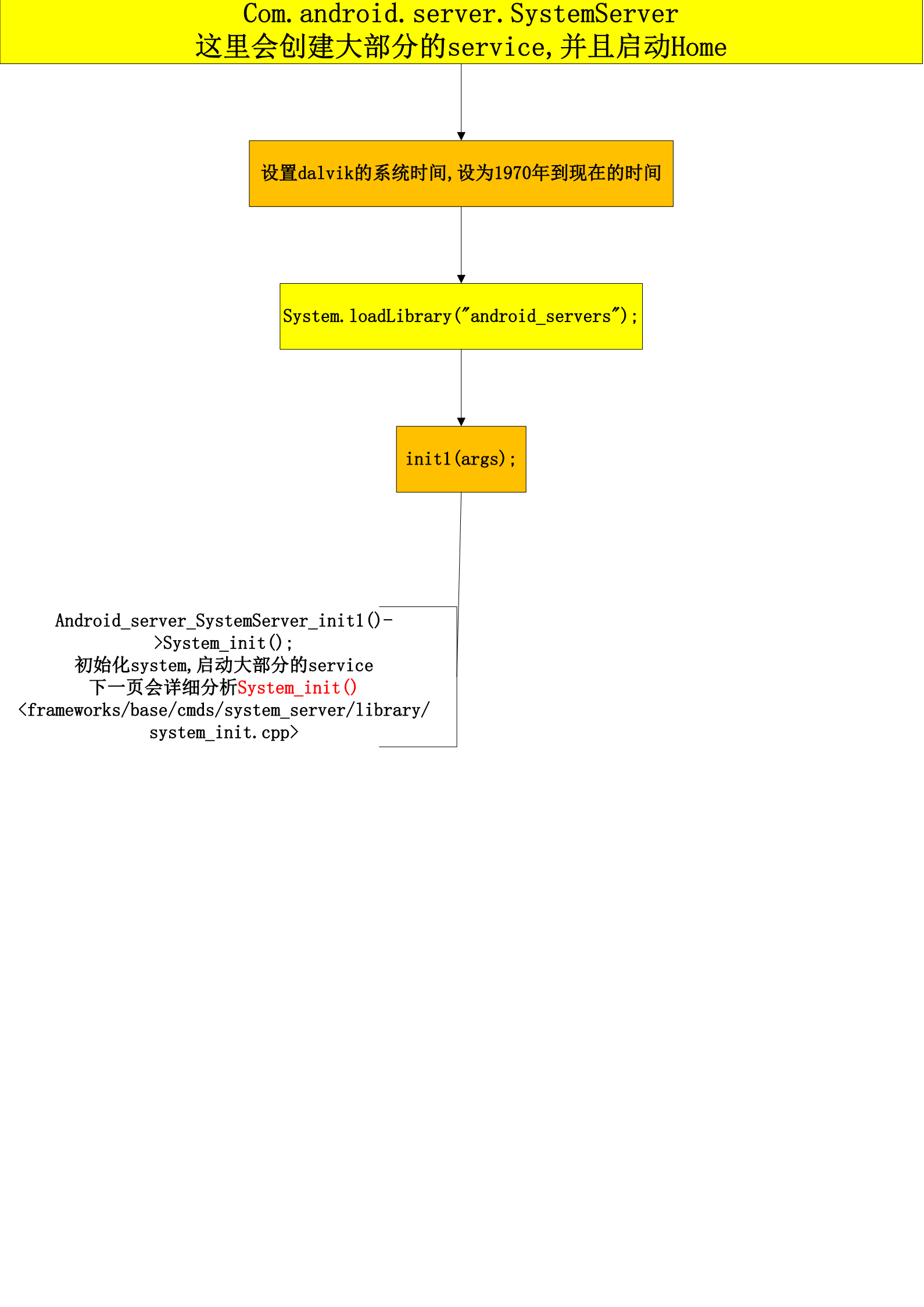

AndroidRuntime::start
android-ics/frameworks/base/core/jni/AndroidRuntime.cpp
startVm()
创建并启动dalvik虚拟机,
详见startVm分析
startReg()
注册系统的jni函数,详
见startReg分析
这里从args获取到
com.android.internal.os.ZygoteInit这个
package，进入他的Main(),这样启动了
SystemServer，这个才是真正的Zygote进程
Char* slashClassName = toSlashClassName(className);
Jclass startClass = env->FindClass(slashClassName);
jmethodID startMeth = env->GetStaticMethodID(startClass,
"main","([Ljava/lang/String;)V");
env->CallStaticVoidMethod(startClass, startMeth, strArray);
见ZygoteInit分析
Frameworks/base/core/java/com/
android/internal/os/ZygoteInit.java

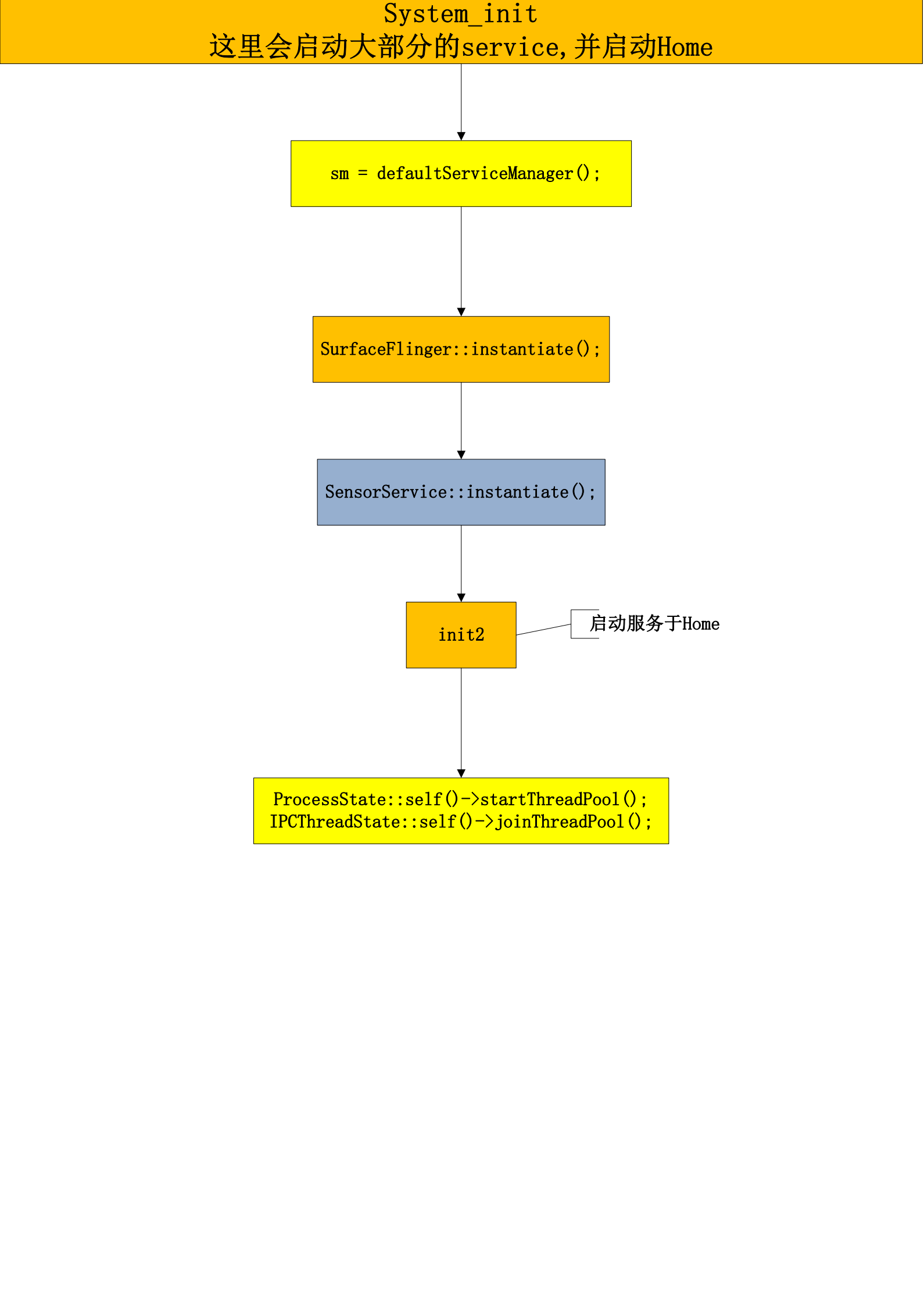

Dalvik虚拟机创建 AndroidRuntime::startVm()
android-ics/frameworks/base/core/jni/AndroidRuntime.cpp
首先配置java虚拟机的运行env
JNI_CreateJavaVM()
创建vm(dalvik虚拟机(实例),每一个app/线程都会为创建一个独占的虚拟机),这里虚拟机创建比较复杂，我会在以后的dalvik的文档里详细说明

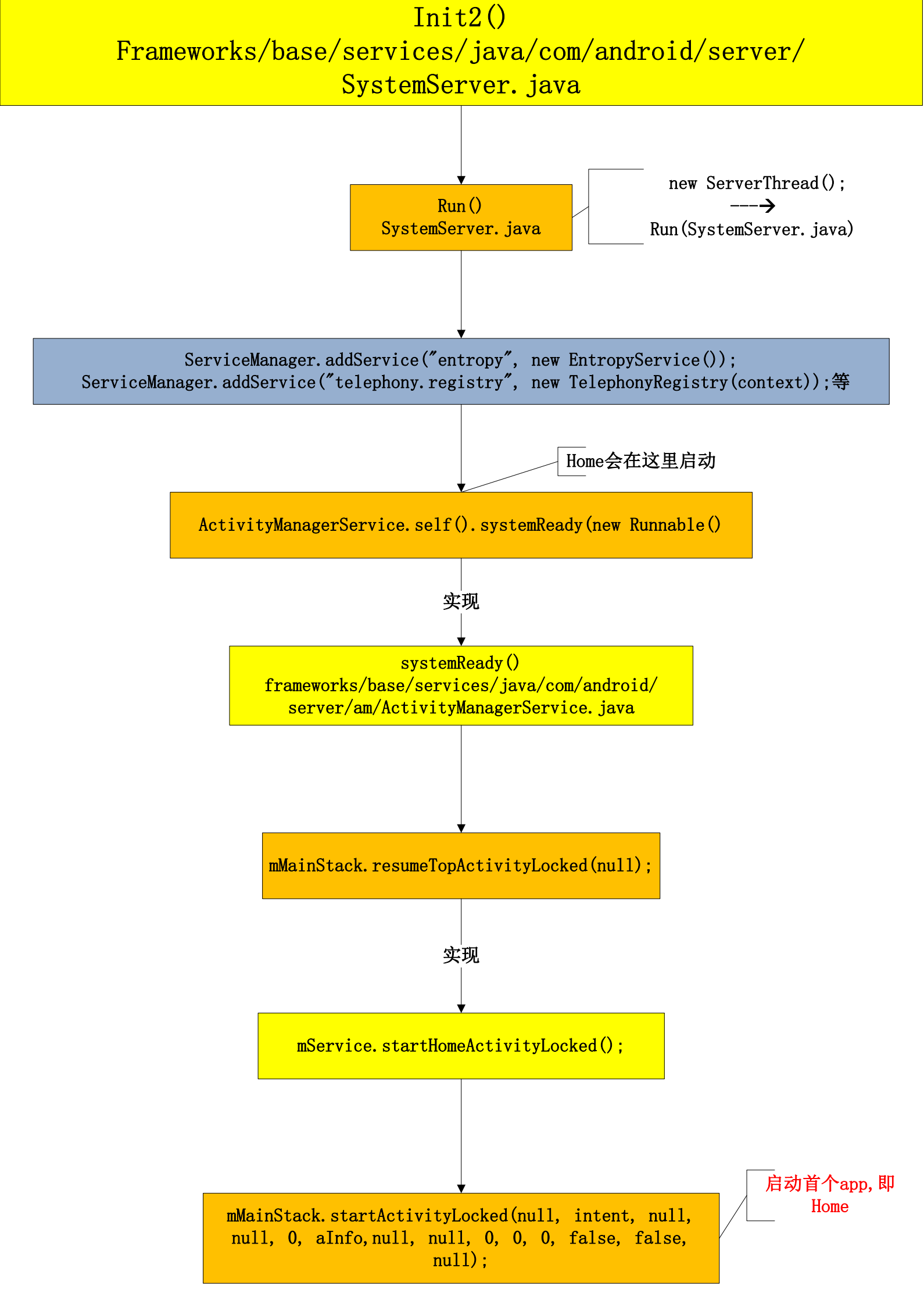

AndroidRuntime::startReg
设置Java虚拟机创建线程的函数
androidSetCreateThreadFunc((android_create_thread_fn) javaCreateThreadEtc);
由于dalvik还未启动，所以这里用
push/pop的方式把Jni函数压入stack中
env->PushLocalFrame(200);
注册jni函数
register_jni_procs(gRegJNI, NELEM(gRegJNI), env)

ZygoteInit
这里是真正的Zygote服务




Com.android.server.SystemServer
这里会创建大部分的service,并且启动Home
设置dalvik的系统时间,设为1970年到现在的时间
System.loadLibrary("android_servers");
init1(args);
Android_server_SystemServer_init1()->System_init();
初始化system,启动大部分的service
下一页会详细分析System_init()
<frameworks/base/cmds/system_server/library/system_init.cpp>


System_init
这里会启动大部分的service,并启动Home
sm = defaultServiceManager();
SurfaceFlinger::instantiate();
SensorService::instantiate();
init2
启动服务于Home
ProcessState::self()->startThreadPool();
IPCThreadState::self()->joinThreadPool();


Init2()
Frameworks/base/services/java/com/android/server/
SystemServer.java
Run()
SystemServer.java
new ServerThread();
--->
Run(SystemServer.java)
ServiceManager.addService("entropy", new EntropyService());
ServiceManager.addService("telephony.registry", new TelephonyRegistry(context));等
Home会在这里启动
ActivityManagerService.self().systemReady(new Runnable()
实现
systemReady()
frameworks/base/services/java/com/android/
server/am/ActivityManagerService.java
mMainStack.resumeTopActivityLocked(null);
实现
mService.startHomeActivityLocked();
mMainStack.startActivityLocked(null, intent, null,
null, 0, aInfo,null, null, 0, 0, 0, false, false,
null);
启动首个app,即
Home