1. init.rc

/system/core/rootdir/init.rc

import /init.${ro.zygote}.rc

ro.zygote 位于属性配置中，64和32位都支持的手机该配置的值为：zygote64\_32

所以使用init.zygote64\_32.rc来运行zygote程序。

service zygote /system/bin/app\_process64 -Xzygote /system/bin --zygote --start-system-server --socket-name=zygote

service zygote\_secondary /system/bin/app\_process32 -Xzygote /system/bin --zygote --socket-name=zygote\_secondary

System\_server 从app\_process64中fork出来，socket名称不一样，ams就是连接不通的socket来fork不同的进程。

1. ams fork进程：

**private final void startProcessLocked(ProcessRecord app, String hostingType)** {

String requiredAbi = (abiOverride != null) ? abiOverride : app.info.primaryCpuAbi;

String instructionSet = null;

if (app.info.primaryCpuAbi != null) {

instructionSet = VMRuntime.getInstructionSet(app.info.primaryCpuAbi);

}

Process.ProcessStartResult startResult = Process.start(

app.info.targetSdkVersion, app.info.seinfo, requiredAbi, instructionSet);

}

**public static final ProcessStartResult start(......String abi,String instructionSet,.....)** {

return startViaZygote(......abi, instructionSet......);

}

**private static ProcessStartResult startViaZygote(String abi,String instructionSet,......)**{

return zygoteSendArgsAndGetResult(openZygoteSocketIfNeeded(abi), argsForZygote);

}

**private static ZygoteState openZygoteSocketIfNeeded(String abi)** throws ZygoteStartFailedEx {

primaryZygoteState = ZygoteState.connect(ZYGOTE\_SOCKET);

if (primaryZygoteState.matches(abi)) {

return primaryZygoteState;

}

secondaryZygoteState = ZygoteState.connect(SECONDARY\_ZYGOTE\_SOCKET);

if (secondaryZygoteState.matches(abi)) {

return secondaryZygoteState;

}

}