



迷你仓



天津积分落户

怎么当健身教



app开发报价

澳大利亚房产

访问： 93646次

积分： 1068

等级： BLOG > 4

排名： 千里之外

原创： 12篇 转载： 0篇

译文： 0篇 评论： 16条

文章搜索

文章分类

[Android基础框架 \(4\)](#)
[Android应用 \(8\)](#)

文章存档

[2011年05月 \(4\)](#)
[2011年03月 \(8\)](#)

阅读排行

View的刷新机制

(20740)

Activity的启动流程

(19372)

Android的CursorAdapter

(12336)

AsyncQueryHandler异步

(7849)

Activity的View构建过程

(6815)

Android性能优化总结

(5184)

Android ListView快速查找

(4943)

Android数据库随笔

(3784)

Android数据库接口性能分析

(3078)

Activity的启动流程（二）

(2781)

评论排行

Activity的启动流程

(7)

View的刷新机制

(3)

Activity的View构建过程

(2)

赠书 | 异步2周年,技术图书免费选

程序员8月书讯

项目管理+代码托管+文档协作, 开发更流畅

Activity的启动流程（二）

标签： [string](#) [classloader](#) [null](#) [thread](#) [exception](#) [android](#)

2011-05-31 14:46 2783人阅读 评

分类： [Android基础框架 \(3\)](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

在前面的介绍的Activity启动流程中，跳过了进程的构建和ActivityThread开始运行的部分，本文就详细展开该过程。

ActivityManagerService启动一个新的进程作为容器

ActivityManagerService.java

```
private final void startProcessLocked(ProcessRecord app, String hostingType, String hostingNameStr) {  
  
    // Process里面通知Zygote服务，Zygote真正产生新的进程（准确说是复制一个）  
  
    int pid = Process.start("Android.app.ActivityThread",  
                           mSimpleProcessManagement ? app.processName : null, uid, uid,  
  
    }  
  
Process.java  
  
public static final int start(final String processClass, final String niceName,  
                             int uid, int gid, int[] gids, int debugFlags, String[] zygoteArgs)  
{  
  
    // 现在的android系统都是支持多进程的，所以应用程序都在单独的进程控件中运行  
    if (supportsProcesses()) {  
  
        // 由Zygote服务启动新的进程  
        return startViaZygote(processClass, niceName, uid, gid, gids,  
                             debugFlags, zygoteArgs);  
    } else {  
  
        // 单进程模式，所有的Android 应用都共享同一个进程 空间  
        Runnable runnable = new Runnable() {  
  
            public void run() {  
                Process.invokeStaticMain(processClass;  
                }  
            }  
  
        // Thread constructors must not be called with null names (see s  
        if (niceName != null) {
```

种牙多少钱







迷你仓



天津积分落户



怎么当健身教



app开发报价



澳大利亚房产

学习，机器翻译还需要啥？

最新评论

- [Activity的启动流程](#)
Boolean布尔: 博主，我要给你生孩子
- [Activity的启动流程](#)
qluojieq: 怎么这么厉害，你是怎么做到的，这个知识能告诉我是怎么样领悟到的呀，看源码吗！
- [Android性能优化总结](#)
JiXianWanMei999: 3.数据交换与网络和数据库的数据交换，不要为了方便，把类型都定义为String。其实我也不赞同这样子...
- [Android ListView快速查找](#)
起个名字好难: 同求例子
- [View的刷新机制](#)
xiaoxia19920920: android源码是最好的代码
- [Activity的启动流程](#)
放风筝的骚年: 很好 很感谢
- [Activity的启动流程（二）](#)
放风筝的骚年: 比较合口味 很好
- [Android性能优化总结](#)
kuangjiu: 说实话 就第一条有点参考意义，嘿嘿希望写点 组件优化之类的，，比如android四大组件的调用，销毁...
- [Activity的View构建过程](#)
kuangjiu: 感谢楼主无私分享
- [Activity的View构建过程](#)
kuangjiu: 但是这里需要注意的是，setContentView创建这些布局结构后，并没有立即跟系统的事件系统接上...

```
new Thread(runnable, niceName).start();
} else {
    new Thread(runnable).start();
}

return 0;
}
}

private static int startViaZygote(final String processClass, final String niceName,
    final int uid, final int gid, final int[] gids, int debugFlags, String[] extraArgs)
{
    synchronized(Process.class) {
        ArrayList<String> argsForZygote = new ArrayList<String>();

        // --runtime-init
        argsForZygote.add("--runtime-init");

        // 设置进程的uid和gid

        argsForZygote.add("--setuid=" + uid);
        argsForZygote.add("--setgid=" + gid);

        // --setgroups is a comma-separated list
        if (gids != null && gids.length > 0) {
            StringBuilder sb = new StringBuilder();
            sb.append("--setgroups=");

            int sz = gids.length;
            for (int i = 0; i < sz; i++) {
                if (i != 0) {
                    sb.append(',');
                }
                sb.append(gids[i]);
            }

            argsForZygote.add(sb.toString());
        }

        if (niceName != null) {
            argsForZygote.add("--nice-name=" + niceName);
        }

        argsForZygote.add(processClass);

        // 通过socket将参数列表传给Zygote并返回新生的进程ID

        pid = zygoteSendArgsAndGetPid(argsForZygote);
    }
}
```

关闭



种牙多少钱



Zygote服务监听本地/dev/socket设备

```
private static void runSelectLoopMode() throws MethodAndArgsCaller {
    while (true) {
        fdArray = fds.toArray(fdArray);
        index = selectReadable(fdArray);

        if (index == 0) {
            ZygoteConnection newPeer = acceptCommandPeer();
            peers.add(newPeer);
            fds.add(newPeer.getFileDescriptor());
        } else {
            boolean done;

            //
            done = peers.get(index).runOnce();
            if (done) {
                peers.remove(index);
                fds.remove(index);
            }
        }
    }
}
```

```
boolean runOnce() {

    // fork调用，复制zygote进程产生子进程

    pid = Zygote.forkAndSpecialize(parsedArgs.uid, parsedArgs.gid,
        parsedArgs.gids, parsedArgs.debugFlags, rlimits);

    if (pid == 0) {
        // in child
        handleChildProc(parsedArgs, descriptors, newStderr);
        // should never happen
        return true;
    } else {
        return handleParentProc(pid, descriptors, parsedArgs);
    }
}
```

```
private void handleChildProc(Arguments parsedArgs,
    FileDescriptor[] descriptors, PrintStream newStderr){

    // 前面Process::startViaZygote时，传入参数"--runtime-init"，所以这里入过了)

    if (parsedArgs.runtimeInit) {
        RuntimeInit.zygoteInit(parsedArgs.remainingArgs);
    } else {
        ClassLoader loader;
```





```
if (parsedArgs.classpath != null) {
    cloder
        = new PathClassLoader(parsedArgs.classpath,
            ClassLoader.getSystemClassLoader());
} else {
    cloder = ClassLoader.getSystemClassLoader();
}

String className;
className = parsedArgs.remainingArgs[0];

String[] mainArgs
    = new String[parsedArgs.remainingArgs.length - 1];

System.arraycopy(parsedArgs.remainingArgs, 1,
    mainArgs, 0, mainArgs.length);

ZygoteInit.invokeStaticMain(cloder, className, mainArgs);

}

}

RuntimeInit.java

public static final void zygoteInit(String[] argv) throws ZygoteInit.MethodAndArgsCaller {
    System.setOut(new AndroidPrintStream(Log.INFO, "System.out"));
    System.setErr(new AndroidPrintStream(Log.WARN, "System.err"));

    commonInit();
    zygoteInitNative();

    int curArg = 0;
    for ( /* curArg */; curArg < argv.length; curArg++) {
        String arg = argv[curArg];

        if (arg.equals("--")) {
            curArg++;
            break;
        } else if (!arg.startsWith("--")) {
            break;
        } else if (arg.startsWith("--nice-name=")) {
            String niceName = arg.substring(arg.indexOf('=') + 1);
            Process.setArgV0(niceName);
        }
    }

    if (curArg == argv.length) {
        Slog.e(TAG, "Missing classname argument to RuntimeInit!");
        // let the process exit
        return;
    }
}
```





// Remaining arguments are passed to the start class's static main

```
String startClass = argv[curArg++];
String[] startArgs = new String[argv.length - curArg];

System.arraycopy(argv, curArg, startArgs, 0, startArgs.length);
invokeStaticMain(startClass, startArgs);
}

private static void invokeStaticMain(String className, String[] argv)
    throws ZygotelInit.MethodAndArgsCaller {

    // We want to be fairly aggressive about heap utilization, to avoid
    // holding on to a lot of memory that isn't needed.
    VMRuntime.getRuntime().setTargetHeapUtilization(0.75f);

    Class<?> cl;
    cl = Class.forName(className);

    Method m;
    m = cl.getMethod("main", new Class[] { String[].class });

    /*
     * This throw gets caught in ZygotelInit.main(), which responds
     * by invoking the exception's run() method. This arrangement
     * clears up all the stack frames that were required in setting
     * up the process.
     */
    throw new ZygotelInit.MethodAndArgsCaller(m, argv);
}
```

这之后就是ActivityThread::main被执行，产生ActivityThread对象并注册给ActivityManagerService。

IActivityManager是ActivityThread调用ActivityManagerService的功能接口

ActivityThread--->IActivityManager--->ActivityManagerService

ApplicationThread则是ActivityManagerService调用ActivityThread的功能接口

ActivityThread<--ApplicationThread<--ActivityManagerService

关闭

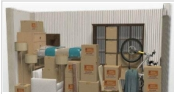
顶 0 踩 0



上一篇 Activity的View构建过程

相关文章推荐

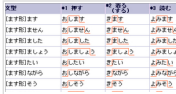
- Android系统应用框架篇：Activity启动流程(二)
- 【直播】大中型UGC信息网站SEO分享--乔向阳
- 浅谈Android的Activity运行流程（生命周期）
- 【直播】打通Linux脉络 进程、线程和调度--宋宝华
- Activity启动流程番外篇
- 【直播】Java最佳学习路线指导--肖海鹏
- Google Android 开发入门与实践
- 【套餐】C++音视频实战技术套餐--夏曹俊
- Activity启动流程详解
- 【套餐】0基础拿下HTML5和CSS3--李仁密
- Activity启动流程源码分析之入门（一）
- 【套餐】机器学习之数学基础系列--AI100
- (上) android 开发入门教程
- Android系统应用框架篇：Activity启动流
- SSH整合activity请假流程
- Android系统应用框架篇：Activity启动流



迷你仓



oa系统



日语学习



拓展魔鬼训练



猎头公司

查看评论

1楼 放风筝的骚年 2012-05-09 14:55发表



比较合口味 很好

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

关闭

