VirtualApp源码阅读

# 基础知识

Java动态代理

<https://blog.csdn.net/ganyao939543405/article/details/76146760>

VA利用**ContentProvider的同步特性构建了一套跨进程同步通信机制**；这使得整个框架的核心得以摆脱AIDL Service异步过程的苦恼

# 调试环境准备

## 源码下载

源码地址 <https://github.com/asLody/VirtualApp.git>

commit HASH: 17fc81c628594ecd2194cba774f1780c2047abbf

## 编译

Windows 7 Professional SP1

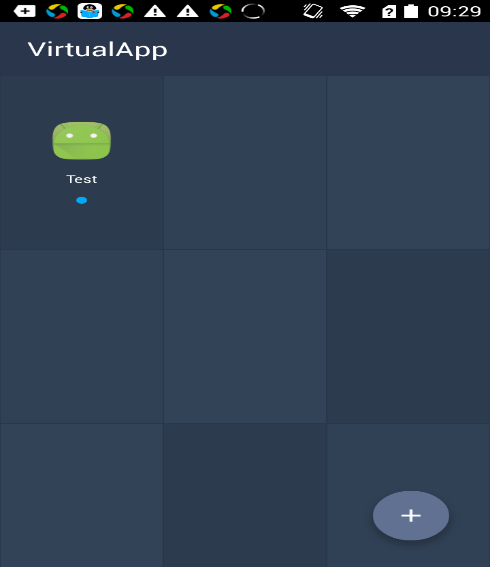
Android Studio 2.3.3

Android SDK 最新版本

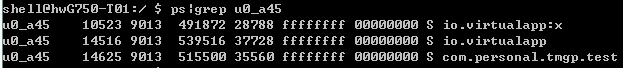
android-ndk-r14b

# 概览

安装运行VirtualApp,通过VirtualApp选择一个已经安装的App,并启动。如图：



查看进程列表：



* 宿主进程：io.virtualapp，这个进程并没什么特殊之处，为了类比，对应于Android系统的Launcher应用程序（桌面）。
* 服务进程：io.virtualapp:x，替换了系统的AMS和PMS，为了类比，对应于Android系统的服务进程system\_server。
* VApp进程：为了类比，对应于Android系统中的常规APP。

在使用VirtualApp过程中，先选择APP进行安装，然后点击对应的APP，运行。本文主要分析它的安装实现和启动实现。分析过程中，会类比Android系统安装和启动APP的过程

# 安装应用程序

## Android 系统

## VirtualApp

程序启动扫描已经安装的VApp

AppRepository.java

public Promise<List<AppData>,Throwable,Void> getVirtualApps()

{

}

UI部分代码

class ListAppPresenterImpl

{

private AppDataSource mRepository;

}

class ListAppFragment

{

}

ListAppActivity

HomeActivity

VirtualCore.java

installPackage

# 运行应用程序

## Android系统

## VirtualApp

只要在一个新的进程中调起了 apk 中任何一个组件，App 将被初始化，Application 将被初始化

ActiityManagerNative.geDefault

HOOK startActivity方法

UI部分

LaunchpadAdapter.java

HomeAcitivity.java

HomePresenterImpl.java

public void launchApp(AppData data)

{

// packageAppData

packageName

icon

name

isLoading

isFirstOpen

fastOpen

}

***io.virtualapp/home/LoadingActivity.java***

public static void launch(Context context, String packageName,int userId)

{

Intent intent = VirtualCore.*get*().getLaunchIntent(packageName, userId);  
if (intent != null) {  
 Intent loadingPageIntent = new Intent(context, LoadingActivity.class);  
 loadingPageIntent.putExtra(*PKG\_NAME\_ARGUMENT*, packageName);  
 loadingPageIntent.addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 loadingPageIntent.putExtra(*KEY\_INTENT*, intent);  
 loadingPageIntent.putExtra(*KEY\_USER*, userId);  
 context.startActivity(loadingPageIntent);  
}

}

com.lody.virtual/client/core/VirtualCore.java

public Intent getLaunchIntent(String packageName, int userId)

{

VPackageManager pm = VPackageManager.get();

pm.queryIntentAcitivities(intentToResolve,intentToResolve.resolveType(content),0,userId);

Intent intent = new Intent(intentToResolve);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

}

com.lody.virtual/client/ipc/VPackageManager.java

public List<ResolveInfo> queryIntentAcitivities(Intent intent,

String resolvedType，

int flags,

int userId)

{

getInterface().queryIntentActivities();

}

VPackageManagerService.java

ServiceManagerNative.java

LoadingAcitivity.java

protected void onCreate(Bundle savedInstanceState)

{

// step1

appModel = PackageAppDataStorage.get().acquire(pkg)

VUiKit.defer().when(()->{

if (!appModel.fastOpen)

{

// step2

VirtualCore.get().preOpt(appModel.packageName);

}

})

.done((res)->VAativityManager.get().startAcitivity(intent,userId));

}

***io.virtualapp/home/repo/PackageAppDataStoreage.java***

public PackageAppData acquire(String packageName)

{

PackageAppData data;

data = loadAppData(packageName);

}

private PackageAppData loadAppData(String packageName)

{

InstalledAppInfo setting = VirtualCore.get().getInstalledAppInfo(packageName,0);

PackageAppData data = new PackageAppData(VApp.getApp(),setting)

}

com.lody.virtualapp/client/core/VirtualCore.java

public InstalledAppInfo ***getInstalledAppInfo***(String pkg, int flags)

{

return getService().getInstalledAppInfo(pkg,flags);

}

IAppManager

第二步 优化

***com.lody.virtualapp/client/core/VirtualCore.java***

public void preOpt(String pkg)

{

InstalledAppInfo info = getInstalledAppInfo(pkg, 0);  
if (info != null && !info.dependSystem && !info.artFlyMode) {  
 DexFile.*loadDex*(info.apkPath, info.getOdexFile().getPath(), 0).close();  
}

}

第三步 startActivity

***com.lody.virtualapp/client/core/VirtualCore.java***

public int ***startActivity***(Intent intent, int userId)

{

ActivityInfo info = VirtualCore.get().resolveActivityInfo(intent, userId);

return startActivity(intent,info,null,null,null,0,userId);

}

public synchronized ActivityInfo ***resolveActivityInfo***(Intent intent,int userId)

{

ActivityInfo activityInfo = null;

activityInfo = resolveActivityInfo(intent.getComponent(),userId);

}

public ActivityInfo resolveActivityInfo(ComponentName componentName, int userId) {  
 return VPackageManager.*get*().getActivityInfo(componentName, 0, userId);  
}

***com.lody.virtualapp/client/ipc/VPackageManager.java***

public ActivityInfo getActivityInfo(ComponentName componentName, int flags, int userId) {  
 try {  
 return getInterface().getActivityInfo(componentName, flags, userId);  
 } catch (RemoteException e) {  
 return VirtualRuntime.*crash*(e);  
 }  
}

***com.lody.virtualapp/client/ipc/VActivityManager.java***

public int startActivity(Intent intent, ActivityInfo info, IBinder resultTo, Bundle options, String resultWho, int requestCode, int userId) {  
 try {  
 return getService().startActivity(intent, info, resultTo, options, resultWho, requestCode, userId);  
 } catch (RemoteException e) {  
 return VirtualRuntime.*crash*(e);  
 }  
}

**服务进程：**

com.lody.virtual/server/pm/VPackageManagerService.java

***com.lody.virtual/server/pm/VAppManagerService.java***

public InstalledAppInfo getInstalledAppInfo(String packageName, int flags)

{

//

}

***com.lody.virtual/server/pm/VPackageManagerService.java***

public ActivityInfo getActivityInfo(ComponentName component, int flags, int userId) {  
 checkUserId(userId);  
 flags = updateFlagsNought(flags);  
 synchronized (mPackages) {  
 VPackage p = mPackages.get(component.getPackageName());  
 if (p != null) {  
 PackageSetting ps = (PackageSetting) p.mExtras;  
 VPackage.ActivityComponent a = mActivities.mActivities.get(component);  
 if (a != null) {  
 ActivityInfo activityInfo = PackageParserEx.*generateActivityInfo*(a, flags, ps.readUserState(userId), userId);  
 ComponentFixer.*fixComponentInfo*(ps, activityInfo, userId);  
 return activityInfo;  
 }  
 }  
 }  
 return null;  
}

public int startActivity(Intent intent, ActivityInfo info, IBinder resultTo, Bundle options, String resultWho, int requestCode, int userId) {  
 try {  
 return getService().startActivity(intent, info, resultTo, options, resultWho, requestCode, userId);  
 } catch (RemoteException e) {  
 return VirtualRuntime.*crash*(e);  
 }  
}

***com.lody.virtual/server/am/VActivityManagerService.java VAMS***

public int startActivity(Intent intent, ActivityInfo info, IBinder resultTo, Bundle options, String resultWho, int requestCode, int userId) {  
 synchronized (this) {  
 return mMainStack.startActivityLocked(userId, intent, info, resultTo, options, resultWho, requestCode);  
 }  
}

***com.lody.virtual/server/am/ActivityStack.java***

int startAcitivtyLocked(int userId,

Intent intent,

ActivityInfo info,

IBinder resultTo,

Bundle options,

String resultWho,

int requestCode)

{

}

private Intent startActivityProcess(int userId,ActivityRecord sourceRecord,

Intent intent,

ActivityInfo info)

{

// 启动一个真正的进程

}

基础库

apk安装流程、四大组件启动流程、反射、动态代理等的深入理解

## android-promise——模块化异步调用的Android类库