Android应用分身

现在市面流程应用分身的机制分为2类:

- 基于Android虚拟技术 如VirtualApp

  VirtualApp
  是一款运行于Android系统的沙盒产品，可以理解为轻量级的“Android虚拟机”。其产品形态为高可扩展，可定制的集成SDK，您可以基于VA或者使用VA定制开发各种看似不可能完成的项目。VA目前被广泛应用于插件化开发、无感知热更新、云控自动化、多开、手游租号、手游手柄免激活、区块链、移动办公安全、军队政府保密、手机模拟信息、脚本自动化、自动化测试等技术领域

```
克隆能力
   可以克隆外部系统中已经安装的App，并在内部运行，互不干扰。典型应用场景为App双开。

免安装能力
  除了克隆已安装之外，VA可以直接在内部安装(外部无感知)apk，并在内部直接运行。典型应用场景为插件化，独立应用市场等。

多开能力
  VA不仅可以“双开”，独特的多用户模式支持用户在内部无限多开同一个App。

内外隔离能力
  VA是一个标准的沙盒，或者说“虚拟机”，提供了一整套内部与外部的隔离机制，包括但不限于(文件隔离/组件隔离/进程通讯隔离)，简单的说VA

对于内部App的完全控制能力
  VA对于内部的App具有完全的监控和控制能力，这点在未Root的外部环境中是绝对无法实现的
```

```
https://github.com/asLody/VirtualApp
```

- 基于Android原生UserManager多用户机制

  华为,oppo xiami，也是使用的原生的多用户机制，但原生的多用户从部分逻辑到界面,都不满足双开的需求, 需要做不少的定制。

我们同样采用原生的多用户机制下的子帐号功能实现应用双开

Code 修改

- 1:在当前用户下面创建子用户

```
public void openSonSpace() {
      mUserManager = (UserManager) mContext.getSystemService(Context.USER_SERVICE);

      // step 1 : call UMS.createProfileForUser() to create Managed Profile User
      UserHandle userHandle = android.os.Process.myUserHandle();

      mShadowSpaceUserInfo = mUserManager.createProfileForUser(UserManager.SHADOW_SPACE_USER_NAME,
              UserManager.USER_TYPE_PROFILE_MANAGED,
              FLAG_MANAGED_PROFILE,
              userHandle.getIdentifier()
              );

      // step 2 call AMS.startUserInBackground() to start the new user.
      int userId = getUserIdFromUserInfo(mShadowSpaceUserInfo);
      try{
        ActivityManager.getService().startUserInBackground(userId);
      } catch (
        RemoteException e) {
            Slog.w(TAG, "could not start pre-created user " + userId, e);
      }

      Log.d(TAG, "startUserInBackground() userId = " + userId + " | isOk = ");

  }
```

- 2：固定子用户的用户ID，供后面分配特定权限

  frameworks/base/services/core/java/com/android/server/pm/UserManagerService.java

```
private UserInfo createUserInternalUncheckedNoTracing(@Nullable String name,

    //hzy_user
    if ((flags & UserInfo.FLAG_CUSTOM_PROFILE) != 0) {
        userId = UserManager.SHADOW_SPACE_USER_ID;
    }else {
        userId = getNextAvailableId();
    }
    Environment.getUserSystemDirectory(userId).mkdirs();
 }
```

```
并确定那些系统服务必须，并安装.
```

```
static {
    mBaseAppforShadowUser.add("android");
    mBaseAppforShadowUser.add("com.android.printspooler");
    mBaseAppforShadowUser.add("com.google.android.gms");
    mBaseAppforShadowUser.add("com.google.android.gsf");
    mBaseAppforShadowUser.add("com.android.packageinstaller");
    mBaseAppforShadowUser.add("com.android.keychain");
    mBaseAppforShadowUser.add("com.android.gallery3d");
    mBaseAppforShadowUser.add("com.google.android.packageinstaller");
    mBaseAppforShadowUser.add("com.android.webview");
    mBaseAppforShadowUser.add("com.android.inputmethod.latin");
    mBaseAppforShadowUser.add("com.android.permissioncontroller");
    mBaseAppforShadowUser.add("com.android.externalstorage");
    mBaseAppforShadowUser.add("com.android.providers.media");
    mBaseAppforShadowUser.add("com.android.providers.media.module");
    mBaseAppforShadowUser.add("com.android.certinstaller");
    mBaseAppforShadowUser.add("com.android.settings");
    mBaseAppforShadowUser.add("com.android.providers.telephony");
    mBaseAppforShadowUser.add("com.android.providers.calendar");
    mBaseAppforShadowUser.add("com.android.htmlviewer");
    mBaseAppforShadowUser.add("com.android.companiondevicemanager");
    mBaseAppforShadowUser.add("com.android.mms.service");
    ...


    if ((flags & UserInfo.FLAG_CUSTOM_PROFILE) != 0) {
         mBaseAppforShadowUser.add("packageName");
         mPm.createNewUser(userId, mBaseAppforShadowUser, disallowedPackages);
     }else{
         mPm.createNewUser(userId, userTypeInstallablePackages, disallowedPackages);
     }
```

- 3、修改新创建子帐号SHADOW_SPACE_USER_ID 的ContentProvider为主帐号共用

> frameworks/base/core/java/android/app/ActivityThread.java

```
 public final IContentProvider acquireProvider(
      Context c, String auth, int userId, boolean stable) {
```

```
    if(userId == UserManager.SHADOW_SPACE_USER_ID &&
            ("packageName".equals(packages[0])||
                    "com.android.gallery3d".equals(packages[0])  )){
        Slog.e(TAG, "hzy acquireProvider change the user to 0 from " + userId);
        userId = 0;
    }

    final IContentProvider provider = acquireExistingProvider(c, auth, userId, stable);
    if (provider != null) {
        return provider;
    }
```

```
frameworks/base/services/core/java/com/android/server/content/ContentService.java
```

```
  @Override
  public void registerContentObserver(Uri uri, boolean notifyForDescendants,
          IContentObserver observer, int userHandle, int targetSdkVersion) {
      if (observer == null || uri == null) {
          throw new IllegalArgumentException("You must pass a valid uri and observer");
      }

      final int uid = Binder.getCallingUid();
      final int pid = Binder.getCallingPid();

      //hzy_user

      String[] packages = mContext.getPackageManager().getPackagesForUid(uid);

      if(userHandle == UserManager.SHADOW_SPACE_USER_ID){
          userHandle = 0 ;
      }else {
          userHandle = handleIncomingUser(uri, pid, uid,
                  Intent.FLAG_GRANT_READ_URI_PERMISSION, true, userHandle);
      }

  }
```

- 4、解决用户交叉访问权限问题

  frameworks/base/core/java/android/content/pm/parsing/ParsingPackageUtils.java

```
  private ParseResult<ParsingPackage> parseBaseApk(ParseInput input, String apkPath,
          String codePath, Resources res, XmlResourceParser parser, int flags)
          throws XmlPullParserException, IOException, PackageParserException {
          ...
          final ParseResult<ParsingPackage> result =
          parseBaseApkTags(input, pkg, manifestArray, res, parser, flags);
        if (result.isError()) {
            return result;
        }
      //hzy_user
      if("packageName".equals(pkg.getPackageName())) {

          if (!pkg.getRequestedPermissions().contains(INTERACT_ACROSS_USERS_FULL_PERMISSION)) {
              pkg.addRequestedPermission(INTERACT_ACROSS_USERS_FULL_PERMISSION.intern());
          }
          if (!pkg.getRequestedPermissions().contains(INTERACT_ACROSS_USERS_PERMISSION)) {
              pkg.addRequestedPermission(INTERACT_ACROSS_USERS_PERMISSION.intern());

          }
      }

      ...
  }
```

```
frameworks/base/services/core/java/com/android/server/pm/permission/PermissionManagerService.java
```

```
private boolean grantSignaturePermission(String perm, AndroidPackage pkg,
        PackageSetting pkgSetting, BasePermission bp, PermissionsState origPermissions) {

        ...
        if (allowed && privilegedPermission &&
          !vendorPrivilegedPermission && pkg.isVendor()) {
              Slog.w(TAG, "Permission " + perm + " cannot be granted to privileged vendor apk "
               + pkg.getPackageName()
               + " because it isn't a 'vendorPrivileged' permission.");
              allowed = false;
            }
          }
        }
        if (!allowed) {

           //hzy_user
           if("packageName".equals(pkg.getPackageName()))) {
               if(perm.equals(ParsingPackageUtils.INTERACT_ACROSS_USERS_FULL_PERMISSION) ||
                       perm.equals(ParsingPackageUtils.INTERACT_ACROSS_USERS_PERMISSION)){
                    allowed =true;
               }
           }
        }
        ...
}
```

- 5：解决子帐号访问帐号ContentProvider权限

  frameworks/base/core/java/android/content/ContentProvider.java

```
protected int enforceReadPermissionInner(Uri uri, String callingPkg,
      @Nullable String attributionTag, IBinder callerToken) throws SecurityException {
  final Context context = getContext();
  final int pid = Binder.getCallingPid();
  final int uid = Binder.getCallingUid();
  String missingPerm = null;
  int strongestMode = MODE_ALLOWED;

  if (UserHandle.isSameApp(uid, mMyUid)) {
    return MODE_ALLOWED;
  }

  //hzy_user_v
  if ( UserHandle.getUserId(uid) == UserManager.SHADOW_SPACE_USER_ID){

     return MODE_ALLOWED;
  }
  ....

}
```

```
@VisibleForTesting
 public Uri validateIncomingUri(Uri uri) throws SecurityException {
     String auth = uri.getAuthority();
     if (!mSingleUser) {
         int userId = getUserIdFromAuthority(auth, UserHandle.USER_CURRENT);
         if(userId == UserManager.SHADOW_SPACE_USER_ID ){ // hzy_user
             Log.w(TAG, "validateIncomingUri skip the usr check for " + UserManager.SHADOW_SPACE_USER_ID );
         }else if (userId != UserHandle.USER_CURRENT && userId != mContext.getUserId()) {
             throw new SecurityException("trying to query a ContentProvider in user "
                     + mContext.getUserId() + " with a uri belonging to user " + userId);
         }
     }
     validateIncomingAuthority(auth);

    ...
 }
```

- 6.试图解决访问主用户多媒体夹/storage/emulated/0/问题
  以图库为例错误如下

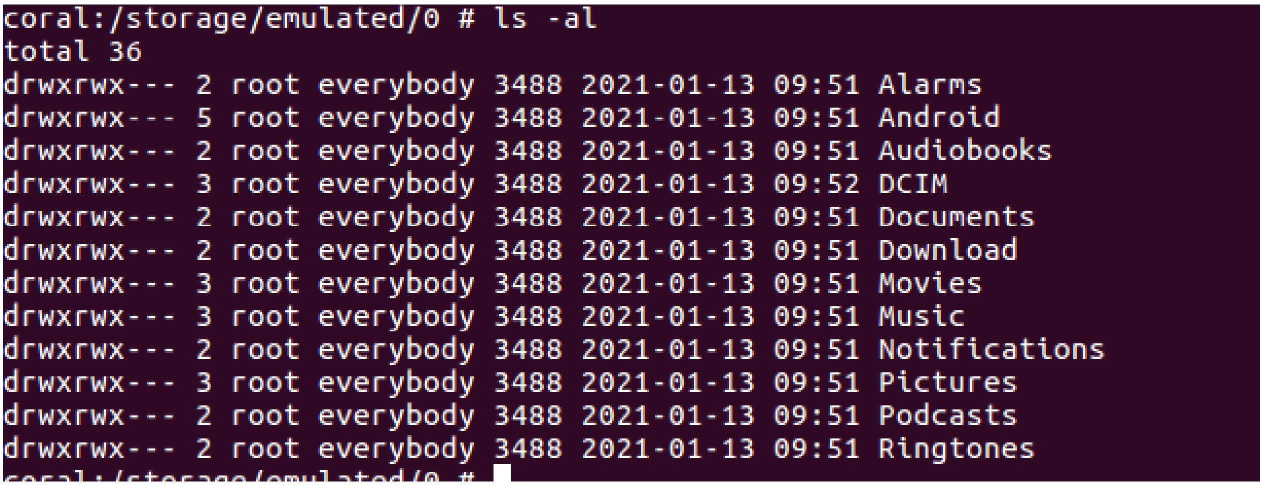
```
01-13 15:33:33.366  4127  4195 W LocalImage: failed to find file to read thumbnail: /storage/emulated/0/DCI
01-13 15:33:33.368  4127  4195 W DecodeUtils: java.io.FileNotFoundException: /storage/emulated/0/DCIM/Camer
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.IoBridge.open(IoBridge.java:492)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at java.io.FileInputStream.<init>(FileInputStream.java:16
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at java.io.FileInputStream.<init>(FileInputStream.java:11
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.data.DecodeUtils.decodeThumbnail
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.data.LocalImage$LocalImageReques
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.data.ImageCacheRequest.run(Image
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.data.ImageCacheRequest.run(Image
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.util.ThreadPool$Worker.run(Threa
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at java.util.concurrent.ThreadPoolExecutor.runWorker(Threa
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at java.util.concurrent.ThreadPoolExecutor$Worker.run(Thr
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at java.lang.Thread.run(Thread.java:923)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at com.android.gallery3d.util.PriorityThreadFactory$1.run
01-13 15:33:33.368  4127  4195 W DecodeUtils: Caused by: android.system.ErrnoException: open failed: EPERM
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.Linux.open(Native Method)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.ForwardingOs.open(ForwardingOs.java:166)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.BlockGuardOs.open(BlockGuardOs.java:254)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.ForwardingOs.open(ForwardingOs.java:166)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at android.app.ActivityThread$AndroidOs.open(ActivityThre
01-13 15:33:33.368  4127  4195 W DecodeUtils:     at libcore.io.IoBridge.open(IoBridge.java:478)
01-13 15:33:33.368  4127  4195 W DecodeUtils:     ... 11 more
```

6.1 关掉selinux 权限检测，问题还是存在，排除selinux影响。

```
adb shell setenforce 0
```

6.2 Linux DAC权限排查

查看/storage/emulated/0文件夹， 用户root和everybody用户组都具有rw权限如下



```
coral:/storage/emulated/0 # ls -al
total 36
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Alarms
drwxrwx--- 5 root everybody 3488 2021-01-13 09:51 Android
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Audiobooks
drwxrwx--- 3 root everybody 3488 2021-01-13 09:52 DCIM
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Documents
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Download
drwxrwx--- 3 root everybody 3488 2021-01-13 09:51 Movies
drwxrwx--- 3 root everybody 3488 2021-01-13 09:51 Music
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Notifications
drwxrwx--- 3 root everybody 3488 2021-01-13 09:51 Pictures
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Podcasts
drwxrwx--- 2 root everybody 3488 2021-01-13 09:51 Ringtones
coral:/storage/emulated/0 #
```

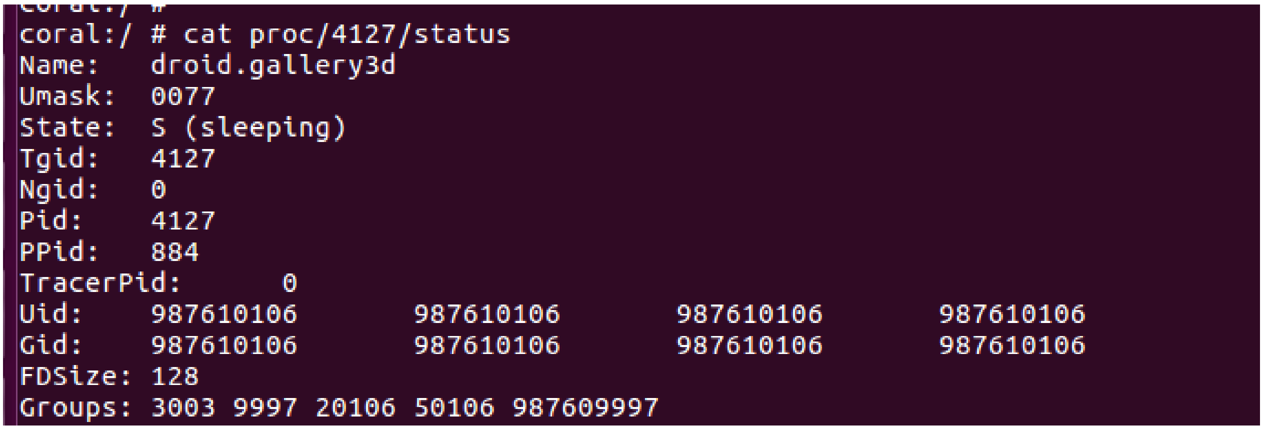
添加图库everybody用户组

frameworks/base/data/etc/platform.xml

```
<!--   hzy_user-->
  <permission name="android.permission.INTERACT_ACROSS_USERS_FULL">
      <group gid="everybody" />
  </permission>

  <permission name="android.permission.INTERACT_ACROSS_USERS">
      <group gid="everybody" />
  </permission>
```

生效之后,还是不能访问，现在Kernel由google提供，日志偏少,还在继续排查原因。



```
coral:/ #
coral:/ # cat proc/4127/status
Name:   droid.gallery3d
Umask:  0077
State:  S (sleeping)
Tgid:   4127
Ngid:   0
Pid:    4127
PPid:   884
TracerPid:      0
Uid:    987610106       987610106       987610106       987610106
Gid:    987610106       987610106       987610106       987610106
FDSize: 128
Groups: 3003 9997 20106 50106 987609997
```

7.键盘共享问题

现在子用户无法访问，主用户的键盘，需要在子用户下面安装单独的键盘。这样存在用户体验不一致问题，需要后面解决。