

Binder系列5—获取服务 (getService)

Nov 15, 2015

- 源码分析
 - [1] getMediaPlayerService
 - [2] getService
 - [3] checkService
 - [4]死亡通知
 - 4.1 注册
 - 4.2 取消注册
 - 4.3 调用机制

基于Android 6.0的源码剖析，本文Client如何向Server获取服务的过程。

源码分析

相关源码

```
/framework/av/media/libmedia/IMediaDeathNotifier.cpp  
/framework/native/libs/binder/IServiceManager.cpp
```

继续以Media为例，开始讲解获取服务的流程：

[1] getMediaPlayerService

==> /framework/av/media/libmedia/IMediaDeathNotifier.cpp

获取服务MediaPlayerService

```

sp<IMediaPlayerService>&
IMediaDeathNotifier::getMediaPlayerService()
{
    Mutex::Autolock _l(sServiceLock);
    if (sMediaPlayerService == 0) {
        sp<IServiceManager> sm = defaultServiceManager(); //获取Service
Manager
        sp<IBinder> binder;
        do {
            //获取名为"media.player"的服务
            binder = sm->getService(String16("media.player")); //【见流
程2】

            if (binder != 0) {
                break;
            }
            usleep(500000); // 0.5 s
        } while (true);

        if (sDeathNotifier == NULL) {
            sDeathNotifier = new DeathNotifier(); //创建死亡通知
        }
        binder->linkToDeath(sDeathNotifier); //将死亡通知连接到binder
        【见流程4】
        sMediaPlayerService = interface_cast<IMediaPlayerService>(bind
er);
    }
    return sMediaPlayerService;
}

```

关于 defaultServiceManager ，前面已经讲过，返回BpServiceManager。在请求获取名为“ media.player” 的服务过程中，采用不断循环获取的方法。由于 MediaPlayerService服务可能还没向ServiceManager注册完成或者尚未启动完成等情况，故则binder返回为NULL，休眠0.5s后继续请求，直到获取服务为止。

[2] getService

==> /framework/native/libs/binder/IServiceManager.cpp

通过BpServiceManager来获取MediaPlayer服务

```
virtual sp<IBinder> getService(const String16& name) const
{
    unsigned n;
    for (n = 0; n < 5; n++){
        sp<IBinder> svc = checkService(name); //【见流程3】
        if (svc != NULL) return svc;
        sleep(1);
    }
    return NULL;
}
```

检索服务是否存在，当服务存在则返回相应的服务，当服务不存在则休眠1s再继续检索服务。该循环进行5次。为什么是循环5次呢，这估计跟Android的ANR时间为5s相关。如果每次都无法获取服务，循环5次，每次循环休眠1s，忽略 checkService() 的时间，差不多就是5s的时间

[3] checkService

==> /framework/native/libs/binder/IServiceManager.cpp

检索指定服务是否存在

```
virtual sp<IBinder> checkService( const String16& name) const
{
    Parcel data, reply;
    data.writeInterfaceToken(IServiceManager::getInterfaceDescriptor()); //写入RPC头
    data.writeString16(name); //写入服务名
    remote()->transact(CHECK_SERVICE_TRANSACTION, data, &reply); //remote为BpBinder
    return reply.readStrongBinder();
}
```

这里调用BpBinder->transact()，再调用到IPCThreadState->transact()，再调用到IPCThreadState->waitForResponse，再调用。这个流程与Binder系列4——注册服务(addService) (<http://www.yuanhh.com/2015/11/14/binder-add-service/>)中的【流程4到流程10】基本一致。此处不再重复,最后reply 里面会返回IBinder对象。

[4]死亡通知

死亡通知是为了让Bp端能知道Bn端的生死情况。

- 定义：DeathNotifier是继承IBinder::DeathRecipient类，主要需要实现其binderDied()来进行死亡通告。
- 注册：binder->linkToDeath(sDeathNotifier)是为了将sDeathNotifier死亡通知注册到Binder上。

Bp端只需要覆写binderDied()方法，实现一些后尾清除类的工作，则在Bn端死掉后，会回调binderDied()进行相应处理。

4.1 注册

注册用该方法：

```
binder->linkToDeath(sDeathNotifier);
```

覆写binderDied方法，主要是实现收尾清除类的工作，比如

```
void IMediaDeathNotifier::DeathNotifier::binderDied(const wp<IBinder>&
who __unused) {
    SortedVector< wp<IMediaDeathNotifier> > list;
    {
        Mutex::Autolock _l(sServiceLock);
        sMediaPlayerService.clear();    //把Bp端的MediaPlayerService清除
掉
        list = sObitRecipients;
    }

    size_t count = list.size();
    for (size_t iter = 0; iter < count; ++iter) {
        sp<IMediaDeathNotifier> notifier = list[iter].promote();
        if (notifier != 0) {
            notifier->died();    //当media server挂了，通知应用程序。应用程
序回调该方法
        }
    }
}
```

4.2 取消注册

当Bp在收到服务端的死亡通知之前先挂了，那么需要在对象的销毁方法内，调用 unlinkToDeath() 来取消死亡通知；

```
IMediaDeathNotifier::DeathNotifier::~~DeathNotifier()
{
    Mutex::Autolock _l(sServiceLock);
    sObitRecipients.clear();
    if (sMediaPlayerService != 0) {
        IInterface::asBinder(sMediaPlayerService)->unlinkToDeath(thi
s);
    }
}
```

4.3 调用机制

每当service进程退出时，service manager会收到来自Binder设备的死亡通知。这项工作是在Service Manager创建的时候Binder系列2 —— 启动Service Manager (<http://www.yuanhh.com/2015/11/07/binder-start-sm/>)，通过 `binder_link_to_death(bs, ptr, &si->death)` 完成。

另外，每个Bp端，也可以自己注册死亡通知，能获取Binder的死亡消息，比如前面的 `IMediaDeathNotifier`。

0条评论

最新 最早 最热

还没有评论，沙发等你来抢


社交帐号登录: 微信 微博 QQ 人人 更多»



说点什么吧...

发布

多说 (<http://duoshuo.com>)

✉ gityuan@gmail.com (<mailto:gityuan@gmail.com>) ·  Github

(<https://github.com/yuanhuihui>) · 天道酬勤 · © 2015 Yuanhh · Jekyll

(<https://github.com/jekyll/jekyll>) theme by HyG (<https://github.com/Gaohaoyang>)