

[CSDN首页 \(http://www.csdn.net?ref=toolbar\)](http://www.csdn.net?ref=toolbar)[学院 \(http://edu.csdn.net?ref=toolbar\)](http://edu.csdn.net?ref=toolbar)[下载 \(http://download.csdn.net?ref=toolbar\)](http://download.csdn.net?ref=toolbar)

更多 ▼

[下载 CSDN APP \(http://www.csdn.net/app/?ref=toolbar\)](http://www.csdn.net/app/?ref=toolbar)[写博客 \(http://write.blog.csdn.net/postedit?ref=toolbar\)](http://write.blog.csdn.net/postedit?ref=toolbar)[登录 \(https://passport.csdn.net/account/login?ref=toolbar\)](http://blog.csdn.net/) | [注册 \(http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister\)](http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)[首页 \(http://blog.csdn.net/\)](http://blog.csdn.net/)[移动开发 \(http://blog.csdn.net/mobile/newarticle.html\)](http://blog.csdn.net/mobile/newarticle.html)

全部 ▼

CSDN (<http://www.csdn.net>)

喜欢

Android 中input event的分析



收藏

原创 2011年03月10日 11:42:00

24526

14

1



评论

文章将分析Android 的Input Event 子系统的来龙去脉。



分享

Android 系统里面有很多小工具，运行这些工具，我们对它们有一个感性的认识，进而阅读和分析这些小工具源代码，再顺藤摸瓜，就可以把整个子系统的来龙去脉弄清楚。

1.运行toolbox的getevent 工具。

```
# getevent -help
getevent -help
Usage: getevent [-t] [-n] [-s switchmask] [-S] [-v [mask]] [-p] [-q] [-c count] [-r] [device]
  -t: show time stamps
  -n: don't print newlines
  -s: print switch states for given bits
  -S: print all switch states
  -v: verbosity mask (errs=1, dev=2, name=4, info=8, vers=16, pos. events=32)
  -p: show possible events (errs, dev, name, pos. events)
  -q: quiet (clear verbosity mask)
  -c: print given number of events then exit
```

**learnrose** (<http://blog.csdn.net/learnrose>)[+ 关注](#)<http://blog.csdn.net/learnrose>

码云

原创

粉丝

喜欢

未开通

11

15

0

<https://github.com/learnrose>

他的最新文章

[更多文章 \(http://blog.csdn.net/learnrose\)](http://blog.csdn.net/learnrose)

bash shell中数组的用法

[\(/learnrose/article/details/6254249\)](/learnrose/article/details/6254249)

ls 命令中支持*的用法

[\(/learnrose/article/details/6253948\)](/learnrose/article/details/6253948)

android 中利用串口来传输文件

[\(/learnrose/article/details/6237082\)](/learnrose/article/details/6237082)

编辑推荐

最热专栏

Android 4.0 事件输入(Event Input)系统...



返回顶部

```

-r: print rate events are received
# getevent -c 20
getevent -c 20
add device 1: /dev/input/event4
  name:  "sensor-input"
add device 2: /dev/input/event3
  name:  "88pm860x_hook"
add device 3: /dev/input/event2
  name:  "88pm860x_on"
add device 4: /dev/input/event1
  name:  "88pm860x-touch"
add device 5: /dev/input/event0
  name:  "pxa27x-keypad"
/dev/input/event0: 0001 0066 00000001
/dev/input/event0: 0000 0000 00000000
/dev/input/event0: 0001 0066 00000000
/dev/input/event0: 0000 0000 00000000
/dev/input/event1: 0003 0000 00000c48
/dev/input/event1: 0003 0001 00000751
/dev/input/event1: 0001 014a 00000001
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0003 0000 00000c67
/dev/input/event1: 0003 0001 000006f9
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0003 0000 00000c9e
/dev/input/event1: 0003 0001 0000069e
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0003 0000 00000cc4
/dev/input/event1: 0003 0001 00000620
/dev/input/event1: 0000 0000 00000000

```



喜欢



收藏



评论



分享

[Linux如何查看与/dev/input目录下的eve...](#)
[Android Framework-----之Input子系统 ...](#)
[Android InputEvent\(Motion/Key/Sleep....](#)
[struct input_event详解 \(/bingqingsuime...](#)

在线课程



Presto的服务治理与架构
在京东的实践与应用
([http://edu.csdn.net/course/series_detail/64?](http://edu.csdn.net/course/series_detail/64?utm_source=blog9)
([http://edu.csdn.net/course/series_detail/64?](http://edu.csdn.net/course/series_detail/64?utm_source=blog9)
Course/series_detail/64?
utm_source=blog9)



深入掌握Kubernetes应用
实践
([http://edu.csdn.net/course/detail/6080?](http://edu.csdn.net/course/detail/6080?utm_source=blog9)
([http://edu.csdn.net/course/detail/6080?](http://edu.csdn.net/course/detail/6080?utm_source=blog9)
e/detail/6080?
utm_source=blog9)



返回顶部

```
/dev/input/event1: 0003 0000 00000ce8
```

```
/dev/input/event1: 0003 0001 000005ba
```

```
/dev/input/event1: 0000 0000 00000000
```

运行这个工具，然后按键或者滑动触摸屏，会看到程序会实时打印event。从上面的输出来看，系统有5个input 子系统。它们分别是



add device 1: /dev/input/event4

name: "sensor-input"

喜欢 #Sensor input 子系统



add device 2: /dev/input/event3

name: "88pm860x_hook"

收藏 #耳机Hook键子系统。可支持接电话挂电话的耳机上面有一个按键，对应的就是这个input 子系统。



评论



add device 3: /dev/input/event2

name: "88pm860x_on"

分享

#开机键 input 子系统

add device 4: /dev/input/event1

name: "88pm860x-touch"

#Touch Screen input 子系统

add device 5: /dev/input/event0

name: "pxa27x-keypad"

#按键子系统，包括Home/Menu/Back等按键。

可以尝试多种event，实际感觉一下出来的log。

2.阅读getevent的代码。代码为./core/toolbox/getevent.c



返回顶部

从代码中，我们知道，程序在while(1)的一个死循环里，不断地在读取 (select 操作)/**dev/input** 下面的文件，检查是否Kernel往里面更新内容，如果有内容更新，就把它打印出来。并且从代码中，我们还知道，任何一个event都有三种属性，type,code,value.

```
while(1) {  
    pollres = poll(ufds, nfd, -1);  
    //printf("poll %d, returned %d/n", nfd, pollres);  
    if(ufds[0].revents & POLLIN) {  
        read_notify(device_path, ufds[0].fd, print_flags);  
    }  
    for(i = 1; i < nfd; i++) {  
        if(ufds[i].revents) {  
            if(ufds[i].revents & POLLIN) {  
                res = read(ufds[i].fd, &event, sizeof(event));  
                if(res < (int)sizeof(event)) {  
                    fprintf(stderr, "could not get event/n");  
                    return 1;  
                }  
                if(get_time) {  
                    printf("%ld-%ld: ", event.time.tv_sec, event.time.tv_usec);  
                }  
                if(print_device)  
                    printf("%s: ", device_names[i]);  
                printf("%04x %04x %08x", event.type, event.code, event.value);  
                if(sync_rate && event.type == 0 && event.code == 0) {  
                    int64_t now = event.time.tv_sec * 1000000LL + event.time.tv_usec;  
                    if(last_sync_time)  
                        printf(" rate %lld", 1000000LL / (now - last_sync_time));  
                    last_sync_time = now;  
                }  
            }  
        }  
    }  
}
```



喜欢



收藏



评论



分享



返回顶部

```
        printf("%s", newline);
        if(event_count && --event_count == 0)
            return 0;
    }
}
```



喜欢

3.问题来了，Android Framework是否也是一样的原理呢？？猜测应该是一样的才对，不然这个工具就没有调试的价值了。



收藏



我们来阅读和分析framework中input event的相关代码。

评论

我们从Kernel层往上看，先看看Framework中，直接操纵/dev/input设备的代码。



分享

在.frameworks/base/libs/ui/EventHub.cpp 中，我们看到跟getevent工具类似的代码。

```
bool EventHub::getEvent(int32_t* outDeviceId, int32_t* outType,
    int32_t* outScancode, int32_t* outKeycode, uint32_t *outFlags,
    int32_t* outValue, nsecs_t* outWhen)
{
    ....

    while(1) {
        ....

        release_wake_lock(WAKE_LOCK_ID);

        pollres = poll(mFDs, mFDCount, -1);

        acquire_wake_lock(PARTIAL_WAKE_LOCK, WAKE_LOCK_ID);
```



返回顶部

```

if (pollres <= 0) {
    if (errno != EINTR) {
        LOGW("select failed (errno=%d)/n", errno);
        usleep(100000);
    }
    continue;
}

// mFDs[0] is used for inotify, so process regular events starting at mFDs[1]
for(i = 1; i < mFDCount; i++) {
    if(mFDs[i].revents) {
        LOGV("revents for %d = 0x%08x", i, mFDs[i].revents);
        if(mFDs[i].revents & POLLIN) {
            res = read(mFDs[i].fd, &iev, sizeof(iev));
            if (res == sizeof(iev)) {
                LOGV("%s got: t0=%d, t1=%d, type=%d, code=%d, v=%d",
                    mDevices[i]->path.string(),
                    ....
            }
        }
    }
}

```

4.那么framework中那个模块再调用EventHub呢，接着往下查。

在framework目录中，输入下面的命令查找

```

# find . -name "*.cpp" |grep -v EventHub | xargs grep EventHub
./base/services/jni/com_android_server_KeyInputQueue.cpp:#include <ui/EventHub.h>
./base/services/jni/com_android_server_KeyInputQueue.cpp:static sp<EventHub> gHub;
./base/services/jni/com_android_server_KeyInputQueue.cpp:    sp<EventHub> hub = gHub;

```



喜欢



收藏



评论



分享



返回顶部

```
./base/services/jni/com_android_server_KeyInputQueue.cpp:    hub = new EventHub;
./base/services/jni/com_android_server_KeyInputQueue.cpp:    sp<EventHub> hub = gHub;
./base/services/jni/com_android_server_KeyInputQueue.cpp:    hub = new EventHub;
```

5.从查找结果中得知，在jni文件 com_android_server_KeyInputQueue.cpp文件中有对EventHub进行调用。



喜欢



打开并阅读com_android_server_KeyInputQueue.cpp文件得知，在下面的函数中调用了EventHub的getEvent函数



收藏



评论



分享

```
static jboolean
android_server_KeyInputQueue_readEvent(JNIEnv* env, jobject clazz,
                                         jobject event)
{
    gLock.lock();
    sp<EventHub> hub = gHub;
    if (hub == NULL) {
        hub = new EventHub;
        gHub = hub;
    }
    gLock.unlock();

    int32_t deviceId;
    int32_t type;
    int32_t scancode, keycode;
    uint32_t flags;
    int32_t value;
    nsecs_t when;
    bool res = hub->getEvent(&deviceId, &type, &scancode, &keycode,
        &flags, &value, &when);
```



返回顶部

```

env->SetIntField(event, gInputOffsets.mDeviceId, (jint)deviceId);
env->SetIntField(event, gInputOffsets.mType, (jint)type);
env->SetIntField(event, gInputOffsets.mScancode, (jint)scancode);
env->SetIntField(event, gInputOffsets.mKeycode, (jint)keycode);
env->SetIntField(event, gInputOffsets.mFlags, (jint)flags);
env->SetIntField(event, gInputOffsets.mValue, value);
env->SetLongField(event, gInputOffsets.mWhen,
                (jlong)(nanoseconds_to_milliseconds(when)));

```



喜欢



收藏



评论



分享

```
return res;
```

6.根据jni的调用规则，在本文件中查找对于的java函数。

```

static JNINativeMethod gInputMethods[] = {
    /* name, signature, funcPtr */
    { "readEvent",      "(Landroid/view/RawInputEvent;)Z",
      (void*) android_server_KeyInputQueue_readEvent },
    ....

```

7. 接着顺藤摸瓜，找到对应的java文件,base/services/java/com/android/server/KeyInputQueue.java

```
private static native boolean readEvent(RawInputEvent outEvent);
```

在一个线程中会调用readEvent函数。

```

Thread mThread = new Thread("InputDeviceReader") {
    public void run() {
        if (DEBUG) Slog.v(TAG, "InputDeviceReader.run()");
        android.os.Process.setThreadPriority(
            android.os.Process.THREAD_PRIORITY_URGENT_DISPLAY);
    }
}

```



返回顶部



喜欢



收藏



评论



分享

```
RawInputEvent ev = new RawInputEvent();
while (true) {
    try {
        InputDevice di;

        // block, doesn't release the monitor
        readEvent(ev);

        boolean send = false;
        boolean configChanged = false;

        if (false) {
            Slog.i(TAG, "Input event: dev=0x"
                + Integer.toHexString(ev.deviceId)
                + " type=0x" + Integer.toHexString(ev.type)
                + " scancode=" + ev.scancode
                + " keycode=" + ev.keycode
                + " value=" + ev.value);
        }
    }
}
```

8.那是谁启动这个线程呢??? 查找mThread变量,得知在KeyInputQueue的构造函数中会启动这个线程。

```
KeyInputQueue(Context context, HapticFeedbackCallback hapticFeedbackCallback) {
    if (MEASURE_LATENCY) {
        lt = new LatencyTimer(100, 1000);
    }

    Resources r = context.getResources();
    BAD_TOUCH_HACK = r.getBoolean(com.android.internal.R.bool.config_filterTouchEvent);
}
```



返回顶部

```
JUMPY_TOUCH_HACK =  
r.getBoolean(com.android.internal.R.bool.config_filterJumpyTouchEvents);  
  
mHapticFeedbackCallback = hapticFeedbackCallback;  
  
readExcludedDevices();  
  
PowerManager pm = (PowerManager)context.getSystemService(  
    Context.POWER_SERVICE);  
mWakeLock = pm.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,  
    "KeyInputQueue");  
mWakeLock.setReferenceCounted(false);  
  
mFirst = new QueuedEvent();  
mLast = new QueuedEvent();  
mFirst.next = mLast;  
  
mThread.start();  
}
```



喜欢



收藏



评论



分享

9.那这个KeyInputQueue是在哪里被实例化呢？

而且查看KeyInputQueue类的声明，得知它是一个abstract class.

```
public abstract class KeyInputQueue  
{  
    .....  
}
```

说明它肯定会被某个类继承.接着查找。



返回顶部

```

/frameworks$ find . -name "*.java" |grep -v KeyInputQueue | xargs grep KeyInputQueue
./policies/base/phone/com/android/internal/policy/impl/KeyguardViewMediator.java: * {@link
com.android.server.KeyInputQueue}'s and {@link android.view.WindowManager}'s.
./base/services/java/com/android/server/PowerManagerService.java:                &&
!"KeyInputQueue".equals(tag))) {
./base/services/java/com/android/server/WindowManagerService.java:import
com.android.server.KeyInputQueue.QueuedEvent;
./base/services/java/com/android/server/WindowManagerService.java:    implements
Watchdog.Monitor, KeyInputQueue.HapticFeedbackCallback {
./base/services/java/com/android/server/WindowManagerService.java:    return
KeyInputQueue.getSwitchState(sw);
./base/services/java/com/android/server/WindowManagerService.java:    return
KeyInputQueue.getSwitchState(devid, sw);
./base/services/java/com/android/server/WindowManagerService.java:    return
KeyInputQueue.hasKeys(keycodes, keyExists);
./base/services/java/com/android/server/WindowManagerService.java:    private class KeyQ extends
KeyInputQueue
./base/services/java/com/android/server/WindowManagerService.java:    implements
KeyInputQueue.FilterCallback {
./base/services/java/com/android/server/InputDevice.java:    // For use by KeyInputQueue for keeping
track of the current touch
./base/services/java/com/android/server/InputDevice.java:        if
(KeyInputQueue.BAD_TOUCH_HACK) {
./base/services/java/com/android/server/InputDevice.java:        Slog.i("KeyInputQueue",
"Updating: " + currentMove);
./base/services/java/com/android/server/InputDevice.java:        Slog.i("KeyInputQueue",
"Updating: " + currentMove);

```

10.从上面的查找结果得知，会在WindowManagerService.java中有一个KeyQ类继承KeyInputQueue类，再在这个文件中查找KeyQ类在哪里定义并实例化的，找到在其构造函数里实例化的。

```
private WindowManagerService(Context context, PowerManagerService pm,
    boolean haveInputMethods) {
    if (MEASURE_LATENCY) {
        lt = new LatencyTimer(100, 1000);
    }
    ....

```

```
    mQueue = new KeyQ();

```

```
    mInputThread = new InputDispatcherThread();

```

```
    PolicyThread thr = new PolicyThread(mPolicy, this, context, pm);

```

```
    ...
}

```



喜欢



收藏



评论



分享

至此，基本上把Input event的Framework的流程全部走完了。WindowManagerService是属于System server进程里面起的一个Service.一开机就会运行，当然其构造函数一开机就能会运行。

至此，整个流程如下：

WindowManagerService

|

|

//

KeyQ

|

|

//

KeyInputQueue

|

|



返回顶部

```
//
EventHub
|
|
//
Kernel device (/dev/input)
```



后续的文章将介绍/dev/input在Kernel中的实现。

喜欢



收藏



评论



版权声明：本文为博主原创文章，未经博主允许不得转载。

举报

分享

标签：android (<http://so.csdn.net/so/search/s.do?q=android&t=blog>) /

input (<http://so.csdn.net/so/search/s.do?q=input&t=blog>) / server (<http://so.csdn.net/so/search/s.do?q=server&t=blog>) /

events (<http://so.csdn.net/so/search/s.do?q=events&t=blog>) / 工具 (<http://so.csdn.net/so/search/s.do?q=工具&t=blog>) /

newline (<http://so.csdn.net/so/search/s.do?q=newline&t=blog>)



tusson (/tusson) 2017-06-21 10:44

12楼

(/tusson) 通过 getevent, sendevent 就可以实现操作录制和播放，自动化测试很方便 <https://github.com/TUSSON/android-event-recorder>



返回顶部

回复



devwang_com (/devwang_com) 2016-05-17 18:41

11楼

(/devwang_com), 最近遇到/dev/input/event11:0000 0000 00000000问题, 前来看看后面打印的三个参数是什么~~~

回复



喜欢



m3225029 (/m3225029) 2014-06-03 10:06

10楼

(/m3225029) 请问楼主, 如果我想读取屏幕输入的原始信息, 即/dev/input/event3文件中内容怎么读取出来? 假设我需要读取这些文件并显示在屏幕上

回复



收藏



评论



分享

查看 14 条热评 ∨

相关文章推荐

Android 4.0 事件输入(Event Input)系统 (/myarrow/article/details/7091061)

1. TouchScreen功能在Android4.0下不工作 原来在Android2.3.5下能正常工作的TouchScreen功能, 移植到Android 4.0就不能正常工作了。凭...



MyArrow (<http://blog.csdn.net/MyArrow>) 2011-12-21 13:54 46131

Linux如何查看与/dev/input目录下的event对应的设备 (/myarrow/article/details/10582667)

1. /dev/input目录 /dev/input目录下的事件都是在驱动中调用input_register_device(struct input_dev *dev)产生的。如我的盒子的/...



MyArrow (<http://blog.csdn.net/MyArrow>) 2013-08-30 11:03 24670

返回顶部



《程序员看天下》实战：揭秘携程大数据的应用处理

一直以来，携程拥有海量数据，如何存储、分析和应用这些数据一直是部门痛点所在！携程大数据团队将会给出什么样的解决方案呢？开源产品的选型和运维又该如何抉择呢....

(http://www.baidu.com/cb.php?c=lgF_pyfqhHmsrHTYrHcOIz0qnfK9ujYzP1D4P1m40Aw-5hc3rHnYnHb0TAq15HfLPWRznjb0T1YzPhcsPWR1nAP9m1czPAm10AwY5HDdnjmLPWf3nWT0lgF_5y9YIZ0IQzq-UZR8mLPbUB48ugfEpZNGXy-jULNzTvRETVnzpyN1gVw-IA7GUatvPjqdAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqhHRLPjnvnfKEpyfqhHc4rj6kP0KWpyfqP1cVrHnz0AqLUWYs0ZK45HcsP6KWThnqPW04njm)



分享

Android Framework-----之Input子系统 (/wangkaiblog/article/details/12085183)

下面这是基于Android4.2代码的关于Input子系统的笔记。在这篇笔记中，只涉及Android相...

wangkaiblog (<http://blog.csdn.net/wangkaiblog>) 2013-09-27 11:25 10150

Android InputEvent(Motion/Key/Sleep....) 流程跟踪 (/fyfcauc/article/details/49948301)

ViewRootImpl的setView()函数中（这个函数被调用代表着Activity的界面基本建立）会建立一群InputStage并以职责链模式链接起来进行协同工作：syntheticInputS...

fyfcauc (<http://blog.csdn.net/fyfcauc>) 2015-11-20 16:18 733

struct input_event详解 (/bingqingsuimeng/article/details/8178122)

查看/dev/input/eventX是什么类型的事件， cat /proc/bus/input/devices 设备有着自己特殊的按键键码，我需要将一些标准的按键，比如0 - 9，X - Z等模拟...


bingqingsuimeng (<http://blog.csdn.net/bingqingsuimeng>) 2012-11-13 12:57 7687



Android平台读写i2c设备开发笔记二 (/rickbeyond/article/details/7839245)

二、使用JNI在应用程序框架层添加服务访问接口 APP应用不能直接访问HAL层，需要JNI层访问HAL模块并向上提供API接口。可以直接提供接口，但建议最好使用服务的方式提供访问。 ...

 Rickbeyond (<http://blog.csdn.net/Rickbeyond>) 2012-08-07 16:43  4381


Android App层通过JNI从驱动获取Input Event (/jack_a8/article/details/43113831)

 1 概述 尝试在App层直接读取驱动的Input Event，获取触屏事件（本文获取的是电磁笔触屏事件），而不通过Android的Input Framework. ...

 jack_a8 (http://blog.csdn.net/jack_a8) 2015-01-25 13:23  1258

收藏



Input event 分析 (/hello_yj/article/details/45058099)

评论  以gsensor器件为例：1、Gsensor上报ABS_DISTANCE后，接着上报sync事件，当event core 检测到sync事件时，认为一包event数据完成【参考下文代码得出】。 ...

分享  hello_yj (http://blog.csdn.net/hello_yj) 2015-04-15 13:30  991

Linux/Android——输入子系统input_event传递 (二) (/jscese/article/details/42099381)

在前文 Linux/Android——usb触摸屏驱动 - usbtouchscreen 中记录了如何在kernel中添加input device 类型为touchscreen的驱动，这在整个输...

 jscese (<http://blog.csdn.net/jscese>) 2014-12-23 20:13  8623

android input子系统详解 (/stephen8341/article/details/48416763)

Input子系统是一个庞大的系统，为了简单明了的介绍该系统，我们采用分模块以及倒序追踪的方式来一步步解开它的面纱。
一、java层事件传递过程 &...

 stephen8341 (<http://blog.csdn.net/stephen8341>) 2015-09-13 16:59  1860


返回顶部

Android 中input event的分析 (/mcu_hong/article/details/7084271)

文章将分析Android 的Input Event 子系统的来龙去脉。 Android 系统里面有很多小工具，运行这些工具，我们对它们有一个感性的认识，进而阅读和分析这些小工具源代码，再顺藤摸...



mcu_hong (http://blog.csdn.net/mcu_hong) 2011-12-19 15:06 429



getevent 工具和 Android 中input event的分析 (/u013478557/article/details/26063671)

喜欢

getevent -help getevent -help Usage: getevent [-t] [-n] [-s switchmask] [-S] [-v [mask]] [-p] [-...



u013478557 (<http://blog.csdn.net/u013478557>) 2014-05-17 13:17 378

收藏



Android 中input event的分析 (/wave_1102/article/details/39345723)

评论



文章将分析Android 的Input Event 子系统的来龙去脉。 Android 系统里面有很多小工具，运行这些工具，我们对它们有一个感性的认识，进而阅读和分析这些小工具源代码，...

分享



wave_1102 (http://blog.csdn.net/wave_1102) 2014-09-17 17:38 520

Android 中input event的分析 (/wangzirui0001/article/details/7818736)

文章将分析Android 的Input Event 子系统的来龙去脉。 Android 系统里面有很多小工具，运行这些工具，我们对它们有一个感性的认识，进而阅读和分析这些小工具源代码，再顺藤摸...



wangzirui0001 (<http://blog.csdn.net/wangzirui0001>) 2012-08-01 15:35 1202



(http://download.csdn.net/detail/meta_cpp/9788746)

Android Input Event Analyzer (http://download.csdn.net/detail/meta_cpp/9788746)



返回顶部

2017-03-21 16:52 3.17MB

[下载](#)[\(/http://download.csdn.net/detail/kilody1/961\)](#)

Android 4.0 事件输入(Event Input)系统 (<http://download.csdn.net/detail/kilody1/961>)

6022

喜欢

2016-08-29 16:32 654KB

[下载](#)

收藏 Android的Event Input System (</doupi2008/article/details/50548184>)

转自:<http://www.cnblogs.com/samchen2009/p/3368158.html> 今天看到一个讲解android 事件分发很好的文章,转载过来,备忘.

评论..

[doupi2008 \(http://blog.csdn.net/doupi2008\)](http://blog.csdn.net/doupi2008) 2016-01-20 14:16 307

分享

Android Input Event Dispatching (</myspor/article/details/6280034>)

The input event dispatch engine is in WindowManagerService.java. WindowManagerService.java creates a...

[myspor \(http://blog.csdn.net/myspor\)](http://blog.csdn.net/myspor) 2011-03-26 15:37 370

Android 4.0 事件输入(Event Input)系统 (</a363175481/article/details/7331766>)

这篇文章太好了, 害怕博主删掉, 备个份: <http://blog.csdn.net/myarrow/article/details/7091061> 1. TouchScreen功能在A...[a363175481 \(http://blog.csdn.net/a363175481\)](http://blog.csdn.net/a363175481) 2012-03-08 10:37 760

Android 4.0 事件输入(Event Input)系统 (/ksjay_1943/article/details/64904444)

返回顶部