May 30, 2012

Monitor file I/O on Android with inotify

👍 Like 8 | 🐦 ツイート | to see | 🔖 24 users

While reading the Android source code the other day, I wanted to check the flow of file I/O that occurs on the system in a given process. This is because I thought it would be useful for understanding the overall picture of processing. So I decided to use inotify.

The Android Linux kernel includes inotify , and the Android SDK provides a FileObserver class that uses inotify .

- Monitor Linux file system events with inotify-www.ibm.com
- FileObserver | Android Developers - developer.android.com
- Android FileObserver - The Underlying inotify Mechanism and an Example

Since Android limits the files and directories that can be accessed by general application processes, using FileObserver to provide tools in the form of an Android app would not be very useful for this purpose. In such cases, the CUI command line tool is more flexible. I decided to run the tool on a root-privileged console obtained by connecting to the adb shell to the Android emulator environment.

Use Android's notify command

Android includes a command line tool "notify" that monitors file I/O using the inotify API. [platform/system/core/toolbox/notify.c]

```
# notify
Usage: notify [-m eventmask] [-c count] [-p] [-v verbosity] path [path ...]
```

Here is an output example when the following command is executed from another process while monitoring the "/data/local/tmp" directory using this notify command.

```
# pwd
/data/local/tmp
# echo aaa > test.txt
# mv test.txt aaa.txt
```

```
# notify -c 10 /data/local/tmp
/data/local/tmp: 00000100 00000000 "test.txt"  ; 0x00000100:IN_CREATE
/data/local/tmp: 00000020 00000000 "test.txt"  ; 0x00000020:IN_OPEN
/data/local/tmp: 00000002 00000000 "test.txt"  ; 0x00000002:IN_MODIFY
/data/local/tmp: 00000008 00000000 "test.txt"  ; 0x00000008:IN_CLOSE_WRITE
/data/local/tmp: 00000040 00000033 "test.txt"  ; 0x00000040:IN_MOVED_FROM
/data/local/tmp: 00000080 00000033 "aaa.txt"   ; 0x00000080:IN_MOVED_TO
```

*The definition of I/O event is [platform/bionic/libc/kernel/common/linux/inotify.h].

Use the inotifywait command

Although Android's notify command is a compact and handy tool, it is inconvenient because it is difficult to read the output and it cannot recursively monitor the directory. It seems interesting to modify and make tools, but I found out that there is a package called inotify-tools in the Linux area, and decided to use the inotifywait command included in it.

- inotifywait(1) - Linux man page
- Monitor directories with inotify-practice now! Linux system administration / Vol.157

I already built it by hand because I couldn't find it if someone had already built and distributed inotifywait that works on Android, and I couldn't find it. The complete set of projects modified for inotifywait and NDK builds are shown below.

inotifywait_bin_android.zip

md5sum [55BF0FD8365A4139D679CE0D6A3A07B3]

Project set (*) -github

(*) The following works are included. When using, please comply with the license agreement of each software.

▪ **Inotify-tools**

Copyright: Rohan McGovern, Radu Voicilas

License: GPL version 2

Site: https://github.com/rvoicilas/inotify-tools/wiki/

▪ **GNU glibc POSIX regular expression functions (glibc-2.11.3 more)**

copyright owner: Free Software Foundation, Inc.

license: LGPL version 2.1 or later

site: Http://Www.Gnu.Org/software/libc/

The following is an example of the output when the self-made application "HelloApp" is started with inotifywait recursively monitoring each directory of /data/cache/system.

```
# ./inotifywait -r -m /data /cache /system
Setting up watches.  Beware: since -r was given, this may take a while!
Watches established.

/system/framework/ ACCESS framework-res.apk
/system/framework/ ACCESS framework-res.apk
/data/app/ OPEN com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ OPEN com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/dalvik-cache/ OPEN data@app@com.example.helloapp-1.apk@classes.dex
/data/dalvik-cache/ ACCESS data@app@com.example.helloapp-1.apk@classes.dex
/data/dalvik-cache/ ACCESS data@app@com.example.helloapp-1.apk@classes.dex
/data/app/ OPEN com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/system/framework/ ACCESS framework-res.apk
/data/app/ ACCESS com.example.helloapp-1.apk
/system/fonts/ OPEN DroidSans-Bold.ttf
/system/lib/hw/ OPEN gralloc.default.so
/system/lib/hw/ ACCESS gralloc.default.so
/system/lib/hw/ ACCESS gralloc.default.so
/system/framework/ ACCESS framework-res.apk
/system/framework/ ACCESS framework-res.apk
/data/app/ CLOSE_NOWRITE,CLOSE com.example.helloapp-1.apk
                    :
```

(tanabe)

klab_gijutsu2 at 16:51 | Comments(0) | TrackBack(0) | Android | win

To the top of this blog | Previous article | Next article

Trackback URL

quick

Comment on this article

name:

Email:

URL:

Remember information: ☐ Rating: -- ○ face ○ star

😊

To post