

T E S T

软 件 测 试

课 程 讲 义

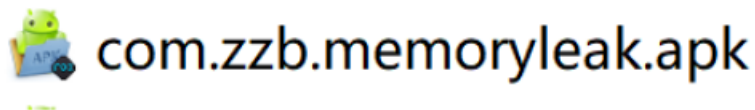
乐科@飞飞老师

1.实施分析日志（Crash）

如果 app 发生异常了，我们怎么去获取对应的信息

例子：

(1)把这个 APP 放入夜神模拟器里面



(2)双击运行这个 demo

首先模拟一下常识 crash 奔溃日志



(3)截取奔溃日志，输入命令

```
adb logcat -v time > d:\log_crash.txt
```

(4)分析日志

✧ 分析 crash 奔溃日志

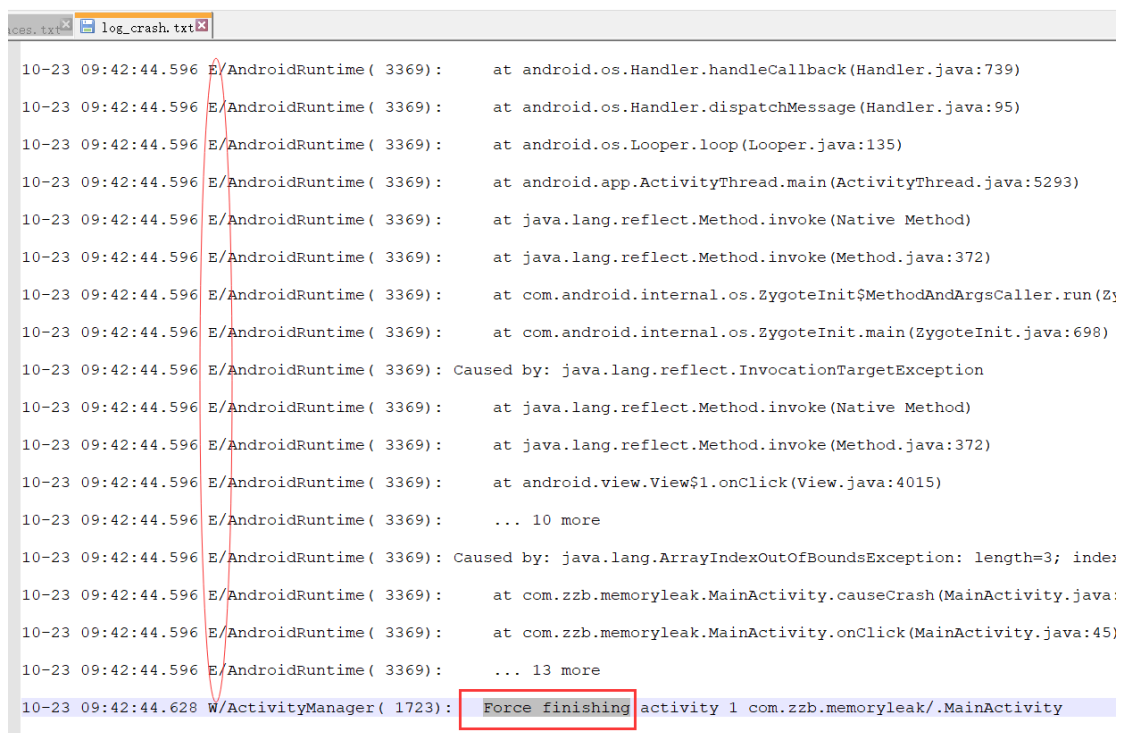
✧ 先根据包名 搜索，再根据 **force finishing** 强制结束 这个关键字

✧ 把force finishing 上面这些e(error级别) 日志全部提交给开发就可以了

✧ 今后在测试时候，遇到APP 发生崩溃，就用这种方式进行测试

✧ 就可以了，开发会认为你对Android 还是非常了解的

✧ 这里是：奔溃，闪退没有提示信息的。



```
10-23 09:42:44.596 E/AndroidRuntime( 3369): at android.os.Handler.handleCallback(Handler.java:739)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at android.os.Handler.dispatchMessage(Handler.java:95)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at android.os.Looper.loop(Looper.java:135)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at android.app.ActivityThread.main(ActivityThread.java:5293)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at java.lang.reflect.Method.invoke(Native Method)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at java.lang.reflect.Method.invoke(Method.java:372)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:698)
10-23 09:42:44.596 E/AndroidRuntime( 3369): Caused by: java.lang.reflect.InvocationTargetException
10-23 09:42:44.596 E/AndroidRuntime( 3369): at java.lang.reflect.Method.invoke(Native Method)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at java.lang.reflect.Method.invoke(Method.java:372)
10-23 09:42:44.596 E/AndroidRuntime( 3369): at android.view.View$1.onClick(View.java:4015)
10-23 09:42:44.596 E/AndroidRuntime( 3369): ... 10 more
10-23 09:42:44.596 E/AndroidRuntime( 3369): Caused by: java.lang.ArrayIndexOutOfBoundsException: length=3; index=
10-23 09:42:44.596 E/AndroidRuntime( 3369): at com.zzb.memoryleak.MainActivity.causeCrash(MainActivity.java:
10-23 09:42:44.596 E/AndroidRuntime( 3369): at com.zzb.memoryleak.MainActivity.onClick(MainActivity.java:45)
10-23 09:42:44.596 E/AndroidRuntime( 3369): ... 13 more
10-23 09:42:44.628 W/ActivityManager( 1723): Force finishing activity 1 com.zzb.memoryleak/.MainActivity
```

2.实施分析日志（ANR）

(1)什么是 ANR？

在android上，如果你的应用程序有一段时间不够灵敏，系统会自动弹出一个对话框，这个对话框称作：应用程序无响应（ANR：application not responding）对话框。用户可以选择“强制关闭”。所以一个流畅合理的应用程序不会出现ANR。

(2)简单理解 ANR

- 俗称 卡死
- 如果应用卡死了（四到五秒），就会出现一个对话框
- 导致卡死原因：**程序优化不到位。**

(3)例子

3.1 产生 ANR 事件

（1）点击产生 ANR

（2）在空白处 点击， 点击， 随意点击

（3）等待 4-5 秒系统抛出异常，卡死就是无响应



3.2 分析 ANR 日志

系统在发生ANR时，会通过三种方式记录信息：

- Logcat日志
- Trace文件
- DropBox服务

首先获取ANR 的logcat 日志

adb logcat -v time > D:\test\logcat_anr.txt

同样等10秒钟左右 ctrl+c 中断

搜索关键字 “ANR in”

```
10-22 21:52:20.629 I/art      ( 1816): Thread[2,tid=1823,WaitingInMainSignalCatcherLoop,Thread*=0xb3c27c00,peer=0x12
10-22 21:52:20.636 I/art      ( 1880): Wrote stack traces to '/data/anr/traces.txt'
10-22 21:52:20.669 I/art      ( 1816): Wrote stack traces to '/data/anr/traces.txt'
10-22 21:52:20.835 I/art      ( 1706): Explicit concurrent mark sweep GC freed 88886(3MB) AllocSpace objects, 7(388K
10-22 21:52:21.346 E/ActivityManager( 1706): ANR in com.zzb.memoryleak (com.zzb.memoryleak/.MainActivity)
10-22 21:52:21.346 E/ActivityManager( 1706): PID: 3022
10-22 21:52:21.346 E/ActivityManager( 1706): Reason: Input dispatching timed out (Waiting to send non-key event bec
10-22 21:52:21.346 E/ActivityManager( 1706): Load: 1.36 / 0.38 / 0.25
10-22 21:52:21.346 E/ActivityManager( 1706): CPU usage from 6674ms to 0ms ago:
10-22 21:52:21.346 E/ActivityManager( 1706): 2.8% 1430/mediaserver: 0% user + 2.8% kernel
10-22 21:52:21.346 E/ActivityManager( 1706): 0.4% 2329/com.bignox.app.store.hd: 0% user + 0.4% kernel
10-22 21:52:21.346 E/ActivityManager( 1706): 0.7% 1894/com.vphone.launcher: 0.1% user + 0.5% kernel / faults: 210
10-22 21:52:21.346 E/ActivityManager( 1706): 0% 3/ksoftirqd/0: 0% user + 0% kernel
10-22 21:52:21.346 E/ActivityManager( 1706): 11% TOTAL: 0.4% user + 10% kernel + 0.3% softirq
10-22 21:52:21.346 E/ActivityManager( 1706): CPU usage from 555ms to 1060ms later:
10-22 21:52:21.346 E/ActivityManager( 1706): 8.5% TOTAL: 0% user + 8.5% kernel
10-22 21:52:21.351 W/WindowManager( 1706): updateFocusedWindowLocked, focusedAppToken is null.
```

通过Logcat日志进行分析：

日志格式

ActivityManager : ANR in [进程名]([部件名])

ActivityManager : PID:[应用进程ID]

ActivityManager : Reason:[原因提示]

[Process CPU state]

根据以上分析此次产生ANR的原因，不是CPU过高的问题。

但是根据这些信息，开发人员暂时还定位不到问题，需要更多的信息。这时候，需要查看trace文件

3.3Trace 文件分析

这个文件对于我们测试来说不需要过多的关注，我们只需要把这个文件下载到本地，发给开发，让开发人员进行分析即可。

1.进入shell，找到traces文件

```
C:\Users\tangfeifei>adb shell 1
root@shamu:/ # cd /data/anr 2
root@shamu:/data/anr # ll 3
-rw-rw-rw- system system 172594 2019-10-22 21:52 traces.txt
```

2.退出shell，下载traces

```
root@shamu:/ # exit 1
C:\Users\tangfeifei>adb pull /data/anr/traces.txt d:\test 2
/data/anr/traces.txt: 1 file pulled. 3.0 MB/s (172594 bytes in 0.054s)
```

注意：pull，一定要选中文件夹才能导出。【前提这个文件已经存在】

思考：traces文件只能记录最后一次发生的信息，如果想要记录多个ANR信息，怎么办？

放在DropBox服务里。

3.4DropBox 服务

例子：构造多次产生ANR和Crash事件

ExceptionDemo

点击产生内存泄漏

点击产生ANR

点击产生crash

点击 三次crash
两次 ANR

1. 查找anr 和crash的文件

```
C:\Users\tangfeifei>adb shell
root@shamu:/ # cd /data/system/dropbox
root@shamu:/data/system/dropbox # ll
```

权限	所有者	组	大小	日期	时间	文件名称
-rw-r--r--	system	system	243	2019-10-20	22:58	SYSTEM_BOOT@1571583491929.txt
-rw-r--r--	system	system	246	2019-10-21	23:09	SYSTEM_BOOT@1571670589166.txt
-rw-r--r--	system	system	246	2019-10-22	14:42	SYSTEM_BOOT@1571726574838.txt
-rw-r--r--	system	system	246	2019-10-22	14:48	SYSTEM_BOOT@1571726933135.txt
-rw-r--r--	system	system	246	2019-10-22	21:34	SYSTEM_BOOT@1571751263411.txt
-rw-r--r--	system	system	22196	2019-10-21	23:28	data_app_anr@1571671714341.txt.gz
-rw-r--r--	system	system	22362	2019-10-22	21:52	data_app_anr@1571752341369.txt.gz
-rw-r--r--	system	system	21704	2019-10-22	22:29	data_app_anr@1571754594118.txt.gz
-rw-r--r--	system	system	15516	2019-10-22	22:30	data_app_anr@1571754618811.txt.gz
-rw-r--r--	system	system	1290	2019-10-22	21:34	data_app_crash@1571751273072.txt
-rw-r--r--	system	system	1290	2019-10-22	22:30	data_app_crash@1571754632258.txt

root@shamu:/data/system/dropbox #

主要关注data_app_anr
和data_app_crash开头日志

注意，data/system文件指的开机启动日志，不需要关注。

我们主要关注：

data_app_anr

data_app_crash 这2个开头的日志

2. 下载整个dropbox目录


```

root@shamu:/data/system/dropbox # exit
C:\Users\tangfeifei>adb pull /data/system/dropbox d:\test
/data/system/dropbox/: 11 files pulled. 1.3 MB/s (85585 bytes in 0.061s)

```

3. 查看电脑中的dropbox文件

本地磁盘 (D:) > test > dropbox

名称	修改日期	类型	大小
data_app_anr@1571671714341.txt.gz	2019-10-22 22:40	好压 GZ 压缩文件	22 KB
data_app_anr@1571752341369.txt.gz	2019-10-22 22:40	好压 GZ 压缩文件	22 KB
data_app_anr@1571754594118.txt.gz	2019-10-22 22:40	好压 GZ 压缩文件	22 KB
data_app_anr@1571754618811.txt.gz	2019-10-22 22:40	好压 GZ 压缩文件	16 KB
data_app_crash@1571751273072.txt	2019-10-22 22:40	文本文件	2 KB
data_app_crash@1571754632258.txt	2019-10-22 22:40	文本文件	2 KB
SYSTEM_BOOT@1571583491929.txt	2019-10-22 22:40	文本文件	1 KB
SYSTEM_BOOT@1571670589166.txt	2019-10-22 22:40	文本文件	1 KB
SYSTEM_BOOT@1571726574838.txt	2019-10-22 22:40	文本文件	1 KB
SYSTEM_BOOT@1571726933135.txt	2019-10-22 22:40	文本文件	1 KB
SYSTEM_BOOT@1571751263411.txt	2019-10-22 22:40	文本文件	1 KB

3.5 总结

- 可以通过logcat日志去定位问题，也可以通过这种方式去定位问题。
2个方法是一样的。
- 以后提取anr和crash日志，下载dropbox日志。
- Dropbox是一款免费网络文件同步工具，是Dropbox公司运行的在线存储服务，通过云计算实现因特网上的文件同步，用户可以存储并共享文件和文件夹。也分收费和不收费2个版本。