

Core行号定位

</>

Plain Text | 收起 ^

```
1 CAR_ID: JIDUL6T79T2E8PP002404
2 崩溃时间: 2023-09-16 14:50:01
3 问题节点: master
4 ANP版本号: 8.4.32.592
5 产出分支: RB_8-4-32-35_base59
6 问题场景: 运行非预期退出 | ['dreamview'] | highway
7 日志下载链接: http://anp-
  toolchain.bj.bcebos.com/data_recycle_daemon/20230916/anp_run_error.1694847001
  /master/JIDUL6T79T2E8PP002404/anp_run_error.1694847001_JIDUL6T79T2E8PP002404_
  20230916145002_master.tar.gz?authorization=bce-auth-
  v1%2Fcfcac25d495c4f1ca238329f51759552%2F2023-09-16T06%3A50%3A10Z%2F-
  1%2Fhost%2F718210c8e4c8c3f63392aa57a2d6f8053223f350c09e3e5ae4e674a167e716fc
8 trace info :*** SIGSEGV (@0x2cef09120a500a) received by PID 258996 (TID 25940
  3) from PID 302665738; stack trace: ***
9 @    0xfffffaafd37c0    __kernel_rt_sigreturn+0
10 @    0xfffffa45bf584    apollo::dreamview::MapService::RetrieveMapElements()+0
  x804
11 @    0xffff9370b21c    +0x4f621c
12 @    0xfffffa45b9c20    apollo::dreamview::WebSocketHandler::handleJsonData()+
  0x820
13 @    0xfffffa45ba658    apollo::dreamview::WebSocketHandler::handleData()+0x31
  8
14 @    0xfffffa5eab9dc    CivetServer::websocketDataHandler()+0x84
15 @    0xfffffa5ea4528    +0x39528
链接: https://console.cloud.baidu-
int.com/devops/icafe/issue/ANP-3505286/show
晓凯, 这个core康哥说你能反解出来
core在哪一行, 这个有文档看一下怎么操作吗?
```

堆栈如上, 用以下方法定位:

1、确定core堆栈所在的动态库

在上述堆栈中看到core所在堆栈函数是

apollo::dreamview::MapService::RetrieveMapElements()+0x804, 看代码该函数所在动态库是 libdreamview-app.so (也可以直接在产出lib目录grep一下RetrieveMapElements, 看哪个库包含这个符号)

2、确定偏移地址, 堆栈位置: apollo::dreamview::MapService::RetrieveMapElements()+0x804, 表示crash的函数是RetrieveMapElements, 0x804表示crash位置相对于该函数的偏移地址, 我们要确定

的是相对于动态库的偏移地址，所以首先要拿到该函数相对于动态库的偏移地址，然后加上0x804即可

3、确定函数偏移地址使用nm命令，如下：

</>

Plain Text | 收起 ^

```
1 nm -C -D ./output/cybertron/lib/libdreamview-app.so | grep
  RetrieveMapElements
2 00000000000078d80 T
  apollo::dreamview::MapService::RetrieveMapElements(apollo::dreamview_new::Map
  ElementIds const&)
```

可以看到前面的这个地址是00000000000078d80，这个就是该函数相对于动态库的地址

4、计算总偏移地址：00000000000078d80 + 0x804 = 0x79584

5、获取debug版本产出，获取行号需要debug符号，如果产出是release，或者是debug然后strip后的，因为没有调试信息，获取不到具体行号，此时编译一个对应的debug版本即可，上述分支是RB_8-4-32-35_base59，在icode上找到这个分支，触发debug版本的编译，编译出一个debug版本（如果本身有debug版本，可以不用再编译）

6、确认debug版本和release版本的符号一致性

如果符号不一致，获取到的行号就不准了，分别使用nm命令看一下debug和release版本的RetrieveMapElements这个函数对应地址的符号是否一致：

</>

Plain Text | 收起 ^

```
1 nm -C -D ./output/cybertron/lib/libdreamview-app.so | grep
  RetrieveMapElements
```

7、使用addr2line命令确定行号，命令如下：

</>

Plain Text | 收起 ^

```
1 addr2line -e ./output/cybertron/lib/libdreamview-app.so 0x79584 -Cf
```

输出结果：

</>

Plain Text | 收起 ^

```
1 /home/bcloud/bcloud_data/EE/BCLLOUD_PROTobuf/CompileServer/Task/c2121b9d449a4d
c04393b1c0632415a8/baidu/adu-
lab/andes/modules/dreamview/backend/map/map_service.cc:207
```