## 高通平台下解析ramdump

[具体操作参考第5点，其他都是过程记录](#Xea32ca4912a196619236642ab39a9d16975bb20)

### 1.下载crash工具

github官网：<https://github.com/crash-utility/crash>

解析64位Android kernel使用如下指令编译：

make target=ARM64

make install

### 2.解压ramdump文件

安装：

apt-get install p7zip-full

解压7z：

7z x file.7z

解压出来就是文件夹.

### 3.gdb依赖包

texinfo bison libncurses-dev

### 4.高通ramdump内存解析脚本适配【记录过程，不用care~】

#### （1）尝试使用github上python2版本（<https://github.com/emonti/qualcomm-opensource-tools>）

##### python2依赖包

enum pyelftools prettytable

##### python2没有collections.abc模块，修改为import collections

##### local\_settings报错

<https://blog.csdn.net/m0_46250244/article/details/112428261>

##### sched\_info.py

from utils.anomalies import Anomaly

修改为 from anomalies import anomaly

##### pip报错sys.stderr.write(f"ERROR: {exc}")

curl <https://bootstrap.pypa.io/pip/2.7/get-pip.py> --output get-pip.py

python get-pip.py

【参考：<https://www.cjavapy.com/article/1605/>】

##### python2不太行~~~

【参考：<https://www.cnblogs.com/rainey-forrest/p/12162216.html>】

报错 supportid

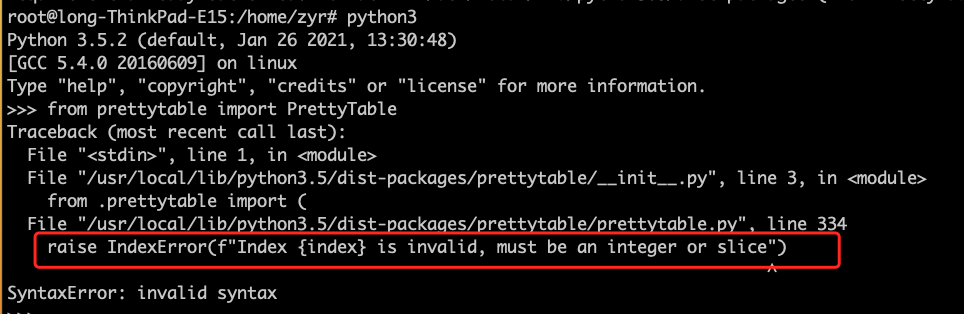
#### （2）尝试用源码中的tools来进行解析

sxr2130p\_repo/emdoor/LINUX/android/vendor/qcom/opensource/tools/linux-ramdump-parser-v2

##### python3依赖包

numpy anomalies prettytable

##### python3遇到问题

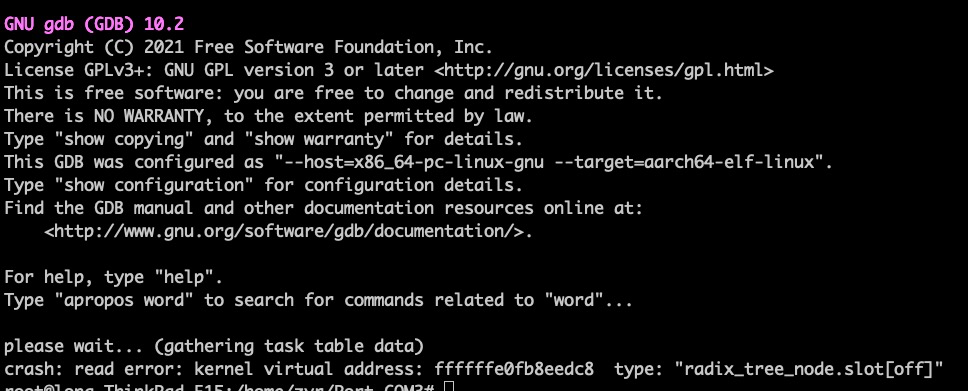


解决方法：重装prettytable即可

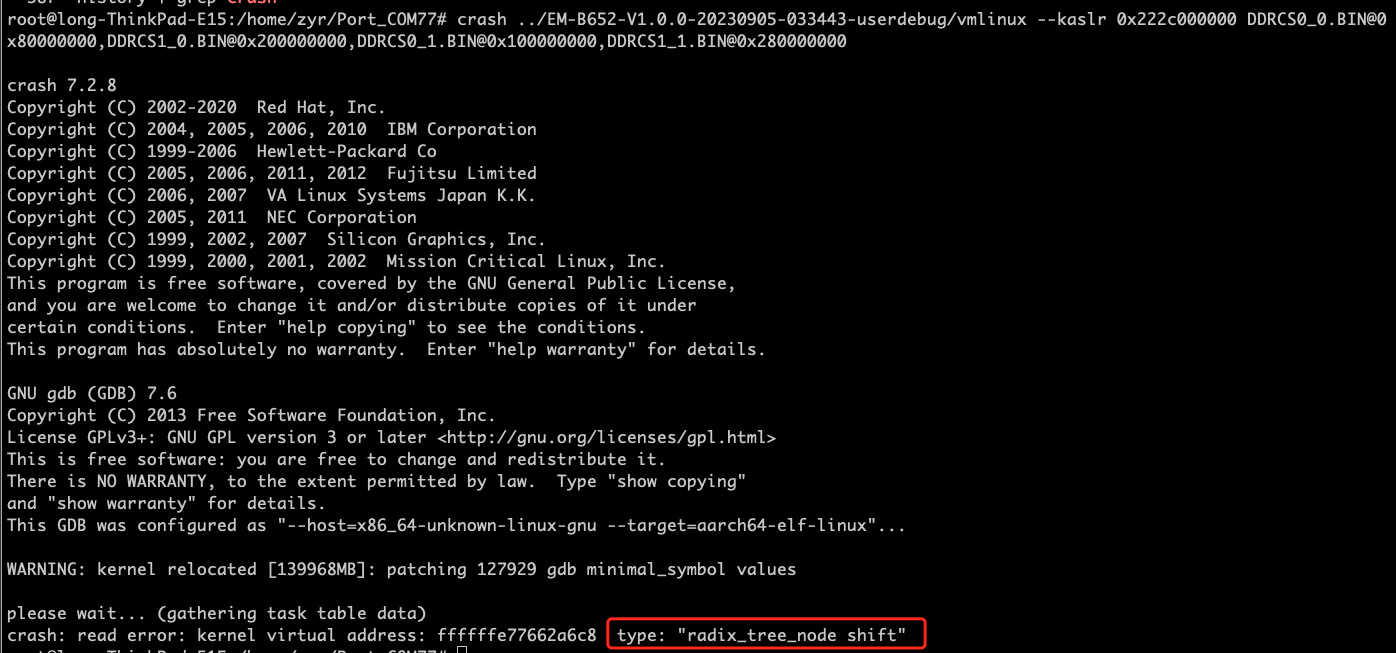
##### 直接使用crash解析还有bug😅

尝试了2个dump和vmlinux都有类似的问题...

crash ../vmlinux --kaslr 0x1b2d400000 DDRCS0\_0.BIN@0x80000000,DDRCS1\_0.BIN@0x200000000,DDRCS0\_1.BIN@0x100000000,DDRCS1\_1.BIN@0x280000000



crash ../EM-B652-V1.0.0-20230905-033443-userdebug/vmlinux --kaslr 0x222c000000 DDRCS0\_0.BIN@0x80000000,DDRCS1\_0.BIN@0x200000000,DDRCS0\_1.BIN@0x100000000,DDRCS1\_1.BIN@0x280000000



#crash/tools源码  
readmem(node + OFFSET(radix\_tree\_node\_slots) + sizeof(void \*) \* off, KVADDR, &slot, sizeof(void \*), "radix\_tree\_node.slot[off]", FAULT\_ON\_ERROR);

看上去是读取内存信息错误，暂时没思路，后面再看下【doing】

### 5.操作方法

#### （1）下载工具链

包括aarch64-linux-android-gdb，aarch64-linux-android-nm，aarch64-linux-android-objdump

<https://gitlab.com/TeeFirefly/prebuilts/-/tree/master/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/bin>

#### （2）编辑bash脚本

ramdump=$1  
vmlinux=$2  
ramparse\_dir=$3/ramparse.py  
outdir=$4  
  
gdb="/home/zyr/prebuilts-master-gcc-linux-x86-aarch64/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/bin/aarch64-linux-android-gdb"  
nm="/home/zyr/prebuilts-master-gcc-linux-x86-aarch64/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/bin/aarch64-linux-android-nm"  
objdump="/home/zyr/prebuilts-master-gcc-linux-x86-aarch64/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/bin/aarch64-linux-android-objdump"  
  
echo $1,$2,$ramparse\_dir,$4  
python3 $ramparse\_dir --vmlinux $vmlinux -g $gdb -n $nm -j $objdump -a $ramdump -o $outdir -x

#### （3）执行脚本，举例如下：

#参数1：内存dump文件目录  
#参数2：vmlinux文件目录  
#参数3：高通平台自带内存解析脚本  
#参数4：输出文件夹名（存放解析结果）  
./dumpparse.sh Port\_COM77 EM-B652-V1.0.0-20230905-033443-userdebug/vmlinux linux-ramdump-parser-v2 ./out77/

#### （4）解析结果

输出文件夹里最重要的是dmesg\_TZ.txt文件，可以根据调用栈快速定界是哪个模块的问题；

除此之外，还可以看到其他信息，如：

tasks.txt 所有进程调用栈信息；

pagetypeinfo.txt 页块信息；

memory.txt 各进程内存占用信息

mem\_stat.txt 整体内存占用信息

### 6.总结

解析出的dmesg\_TZ.txt日志可用于定界，但目前使用crash还存在bug未解决，无法像在linux系统下解core那样使用

### 7.参考链接

1.基于crash工具搭建分析ramdump的平台

<https://blog.csdn.net/u014175785/article/details/112868957>

2.高通平台RAMDUMP分析

<https://www.tsz.wiki/tools/optimize/crash/qcom-ramdump/qcom-ramdump.html#_3-ramdump>