大唐高鸿CP-ARM调试工具说明书

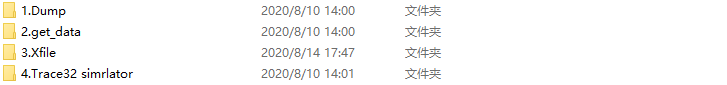
# 1 简介

本文描述了针对CP-ARM死机原因分析的工具使用方法。CP-ARM死机后，使用本工具对死机现场的信息进行、恢复、分析，给出可能的死机原因，以及程序运行过程的函数调用流程，在一定程度上协助开发者定位bug。

# 2 工具介绍

本章主要针对四部分工具的使用方法进行详细介绍。

2.1目录介绍，首先针对工具包中各子目录简要介绍，如下：

1.<Dump>:dump memory的AP侧工具，放入/bin

2.<get\_data>:PC侧内存数据重建工具

3.<Xfile>Xfile生成工具

4.<Trace32 simrlator> 内存查看，axf汇编工具

## 2.2 Dump工具使用

本工具运行环境：AP

Dump是AP侧死机信息导出工具。当CP-ARM死机后，在AP侧使用此工具可将 /dev/modem 的200Mb数据全部导出。为了节省片内空间，可以通过特定指令将导出的数据进行分片以及压缩。 dump工具可以放置在AP侧**任何路径**下，可以通过Xftp等工具将其传输进板内，第一次使用时应使用如下命令，给予其运行权限：

chmod 777 dump

该工具参数说明如表1所示。

表格 1 dump 工具参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数格式 | 可选/必选 | 说明 |
| -l | **十进制** | **必选** | 需要导出的数据的大小，单位为byte，该参数必选为4096的整数倍。 |
| -s | 十进制 | 可选 | 目标数据分片压缩数 |
| -n | 不限 | 可选 | 对导出的自定义数据命名 |
| -a | 十六进制 | 可选 | 设定导出数据的偏移地址，默认为0 |
| -d | 有效路径名 | 可选 | 设定导出数据的目录 |

### 2.2.1 参数-l使用例程

参数-l(length)是必选的，表示要导出的数据大小，将dump.bin拷入AP侧，运行指令如下：

./dump -l 209715200

本指令将会在当前目录下导出 /dev/modem中200MB数据，导出后将其压缩为1个包。(209715200 = 1024 \* 1024 \* 200 byte),实际运行效果如图2-1所示。

图 2-1 参数l使用例程

### 2.2.2 参数s使用例程

参数s(split)是可选的，表示此次会将导出的数据分成几片进行压缩。当片内目标存储路径空间不足时，单次不能完整的放下200Mb的导出数据，使用本参数，将数据分片导出，每一分片的数据被导出后会，立马进行压缩操作，可以在较小的空间内完成数据的获取。使用方法如下：

./dump -l 209715200 -s 5

运行该指令，会将导出的200MB数据分成5片进行压缩，得到5个压缩包，这5个压缩包数据是连续的，必须放置在一起，为后续的数据解压缩合并工作做准备。运行效果如图2-2所示。

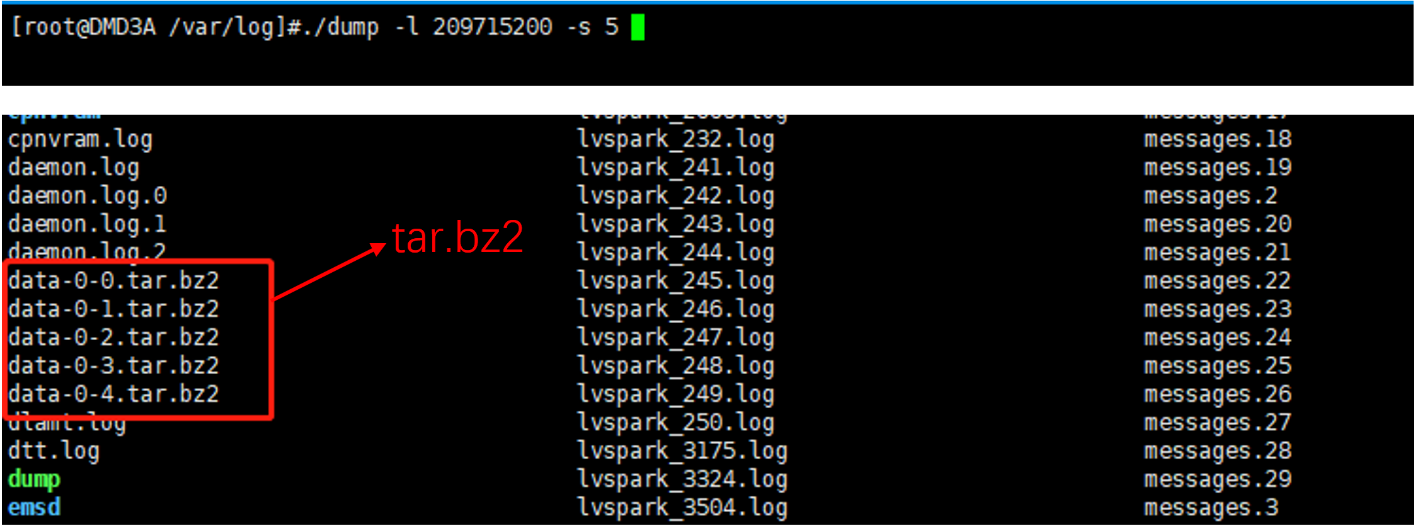


图 2-2参数s使用例程

### 2.2.3 参数-n使用例程

参数-n(name)用来对导出的文件设定名称。Dump工具导出的数据命名规则如图2-3所示。

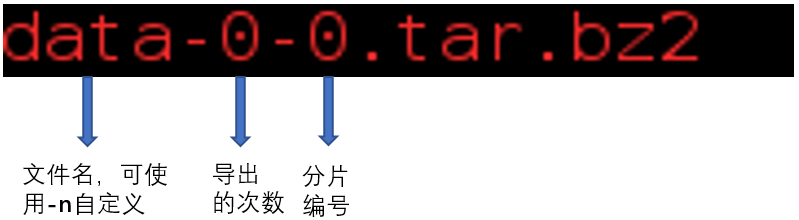


图 2-3 dump数据命名格式

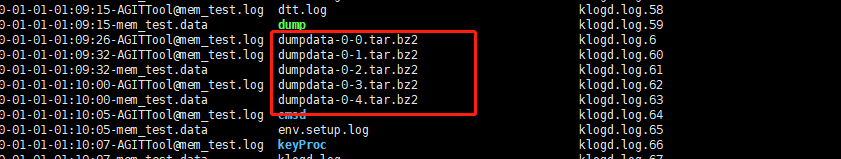
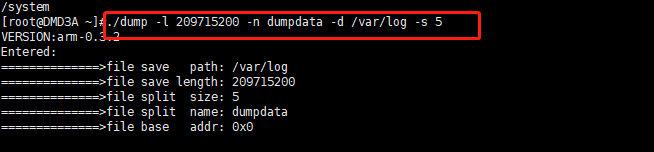
其中 **data** 表示导出数据的名称，其后的第一个**0**表示导出的次数，即使用该名称在当前路径下数据导出次数。**注意：本程序仅会保留最新的五次数据导出。**当文件名称未发生变化时，每导出导出一次数据，新文件中导出的次数会自动加1，该参数从0变化至4，循环往复。第二个**0**表示，分片的编号，该编号从0增长至s参数给定的数值。

通过使用**-n**参数可以人为设定导出数据的名称，使用方法如图2-4所示。

### 2.2.4 参数-d使用例程

参数-d用来指定导出数据的保存路径，若当前路径的空间不足以保存导出数据时，可使用该参数进行数据导出路径的设定。图2-4使用-n命名其为**dumpdata**, -d指定其保存路径为：**/var/log**

图 2-4 dump参数-n和-d使用演示



### 2.2.6 参数-a使用例程

参数-a(十六进制)可以设定导出数据的偏移量，默认为0，当改变该值时，程序将从该参数指定的偏移处开始进行数据导出。**一般不设置，取默认即可。**

### 2.2.7 导出死机文件

在AP侧运行dump工具后，将得到的压缩文件使用xftp等工具导入到PC侧，为数据恢复、分析做准备。

## 2.3 xFile工具使用

本工具运行环境：windows，此工具通过分析由AP侧导出的数据，给出可能发生死机的原因以及栈回溯的过程，通过这些信息协助开发者快速定位bug。

### 2.3.1 准备

运行此工具所必须的文件如图2-5。addr2line为地址转换工具，readelf是源码读取文件，xFileGen是栈回溯及错误信息分析工具，需要配合L1860-MODEM.axf源码文件，此源文件需与系统跑死时所烧录的源码文件一致。

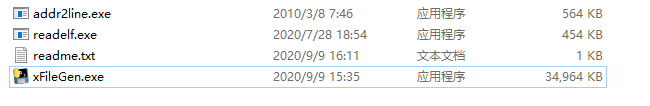


图 2-5 get\_data.exe运行效果

### 2.3.2 xFile的使用

(a) 双击xFileGen.exe文件

(b) 生成sym.xlsx，如图2-9所示，点击ASF，选择.axf文件。

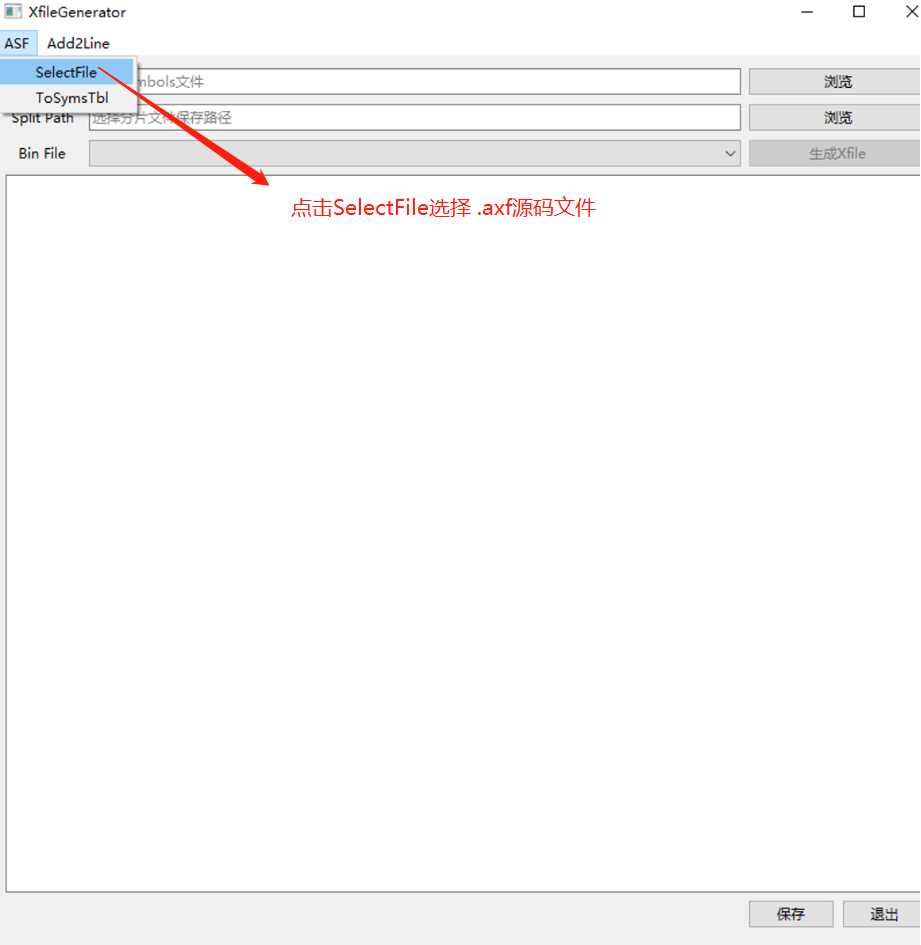


图 2-9 添加.axf文件

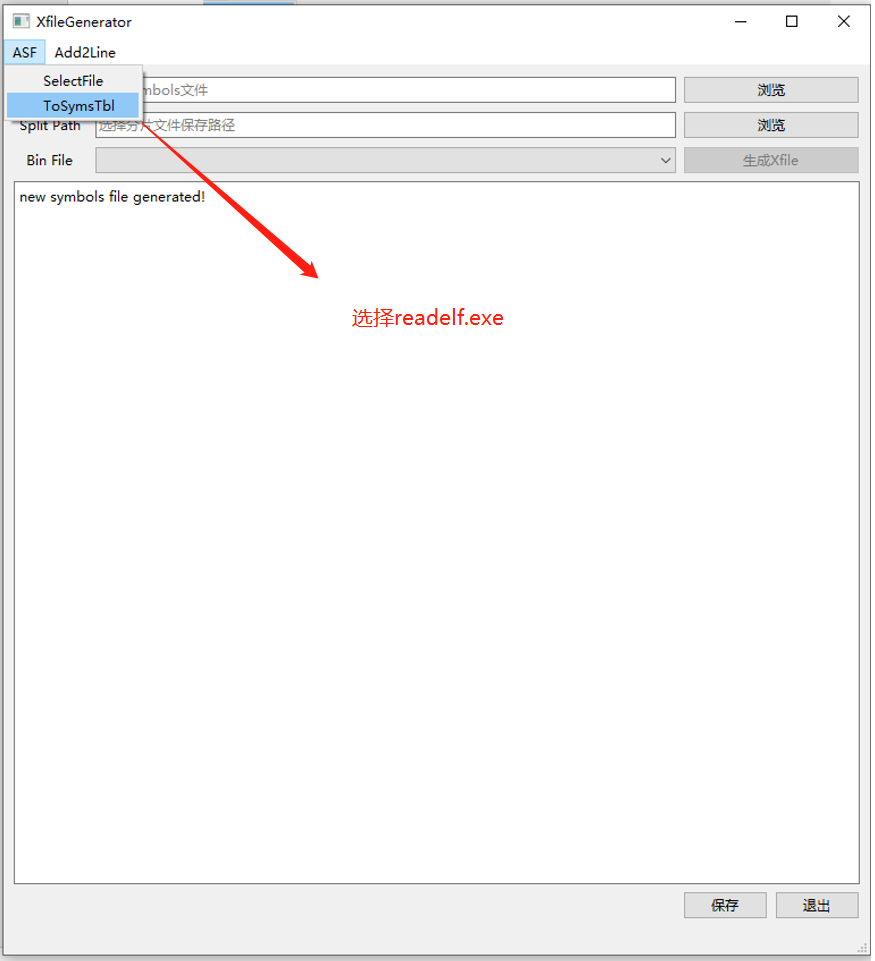


图 2-10 生成.symble file文件

(c) 点击ASF, 选择readelf.exe 当输出“new symbols file generated!”，应当生成sym.xlsx文件，如图2-10所示。

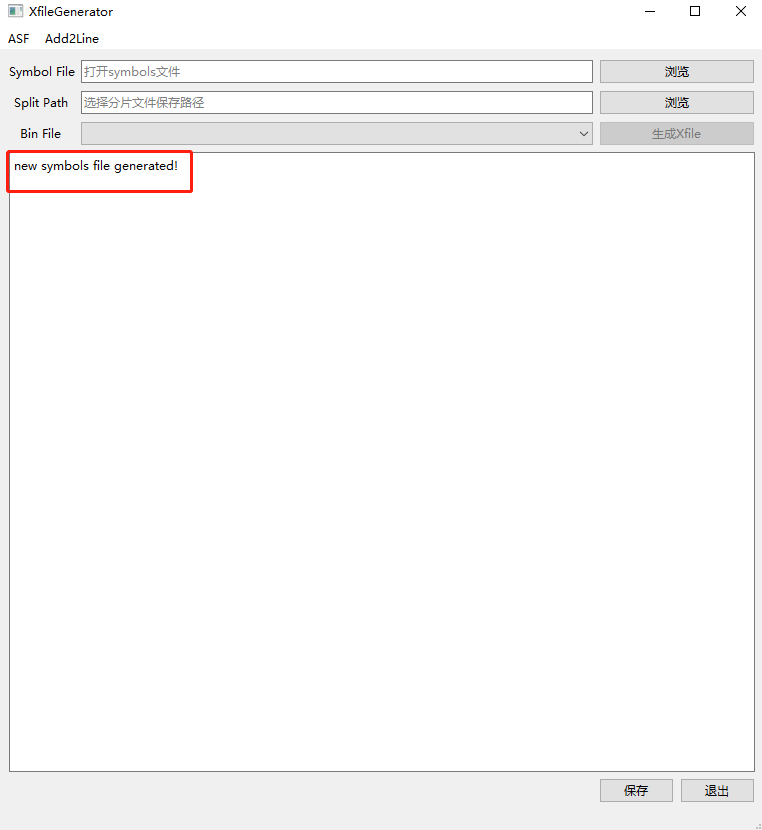


图 2-11 生成.symble file文件

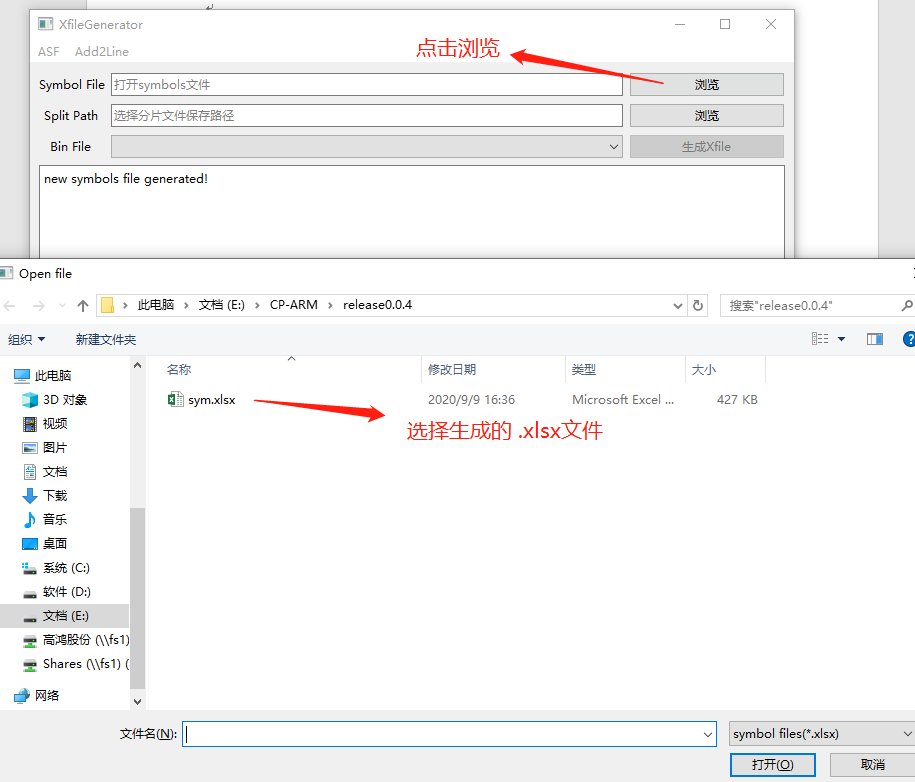
(d) 载入刚生成的symbol file,点击浏览，载入sym.xlsx，如图2-11所示。

图 2-11 添加生成的.symble file文件

(e) 点击Add2line 选择addr2line.exe工具，如图2-12所示

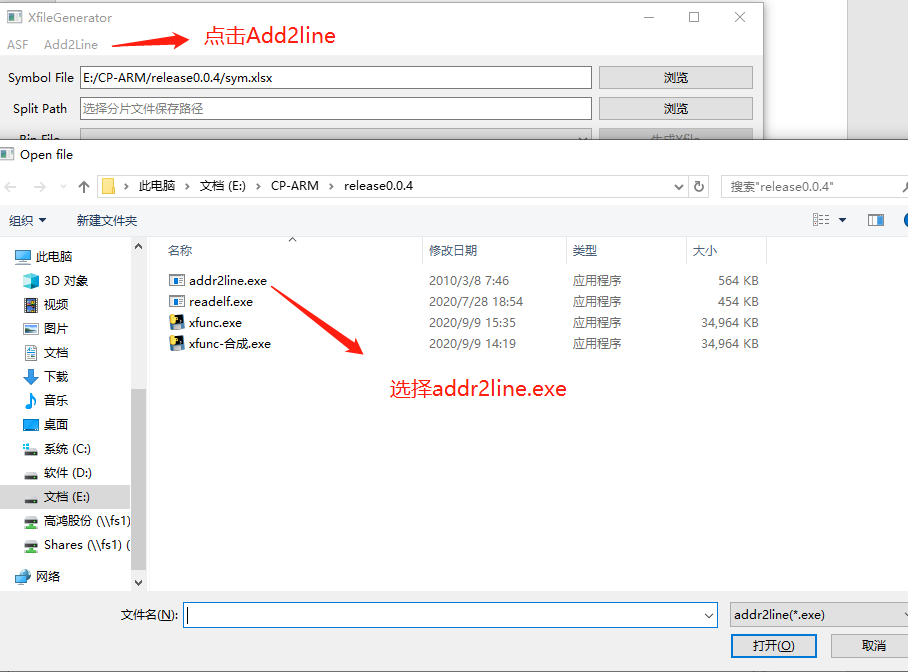


图2-12使用addr2line工具

(f) 载入dump file，浏览，载入dump出来的bin文件，如图2-13所示。

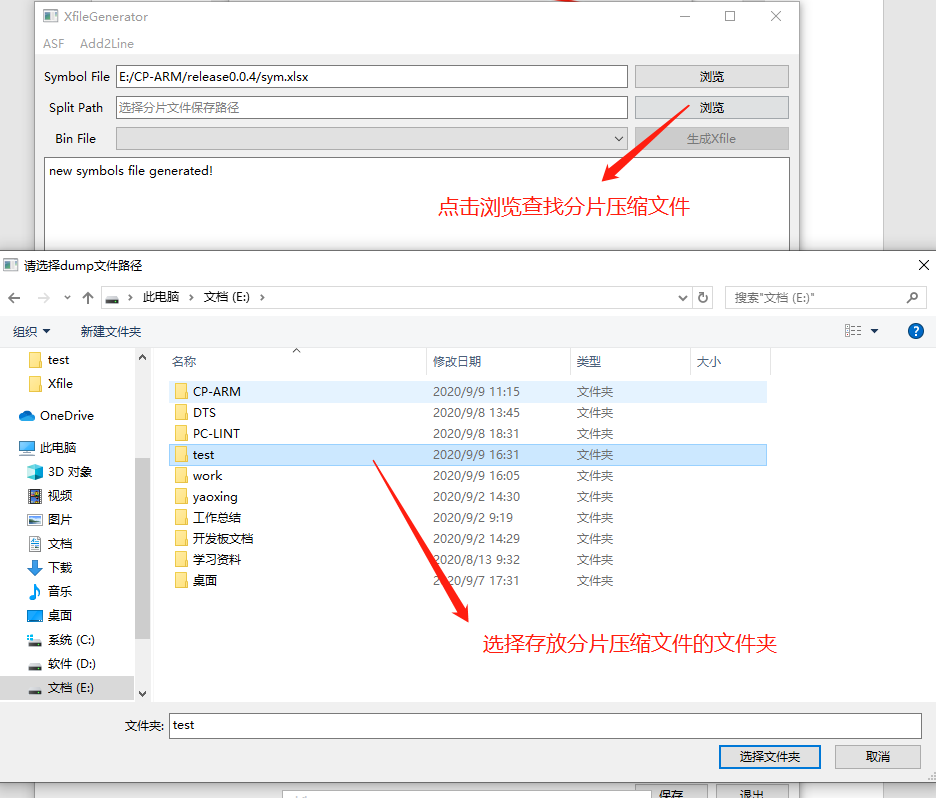


图 2-13 选择.存放分片压缩文件的目录

(g) 选中目录后，若该目录下存放的有AP侧导出的分片压缩文件以**tar.bz2结尾**，会自动加载把所有的文件按其命名加载进来，如图2-13所示。若无分片压缩文件，（**生成Xfile**按钮不可用）选择此次导入的文件名，点击生成Xfile按钮，应当输出分析的结果，如图2-14所示。点击下方保存，可以保存此次生成结果。

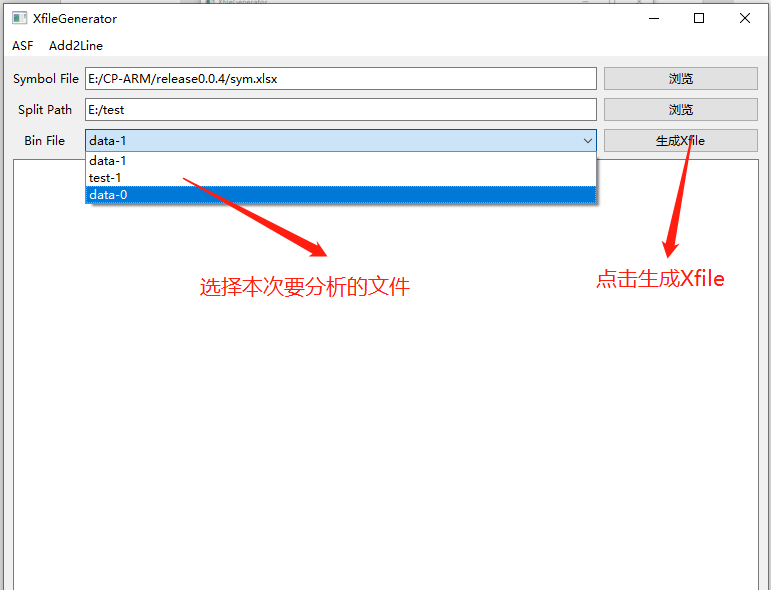
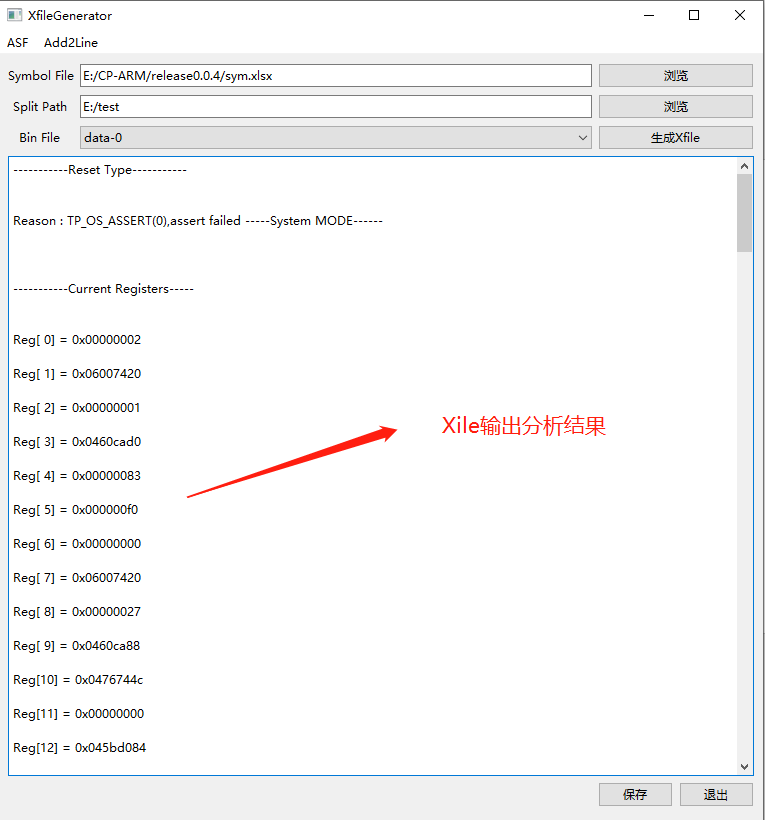


图 2-15 加载分片压缩文件



图Xfile 给出分析结果

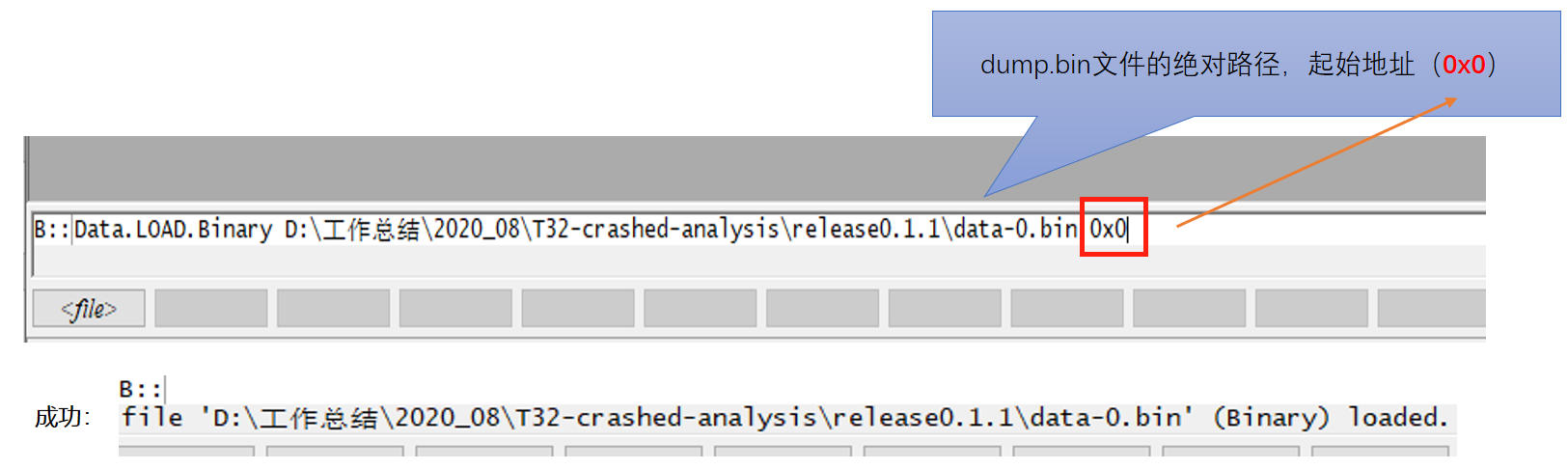
## 2.4 trace32工具

### 2.4.1 trace32工具说明

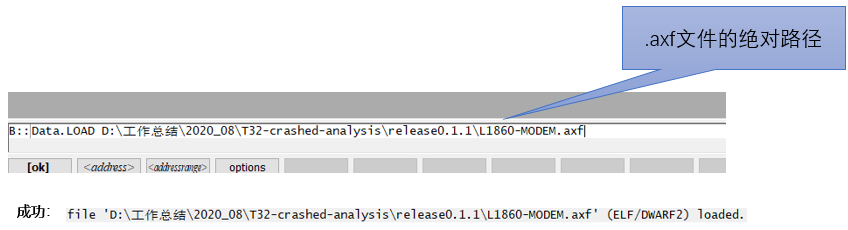
Trace32s是德国Lauterbach公司产品，一种对硬件的仿真模拟，类似于虚拟机，我们可以不用连接实体目标板，就可以对我们的软件进行一些debug以及内存分析。

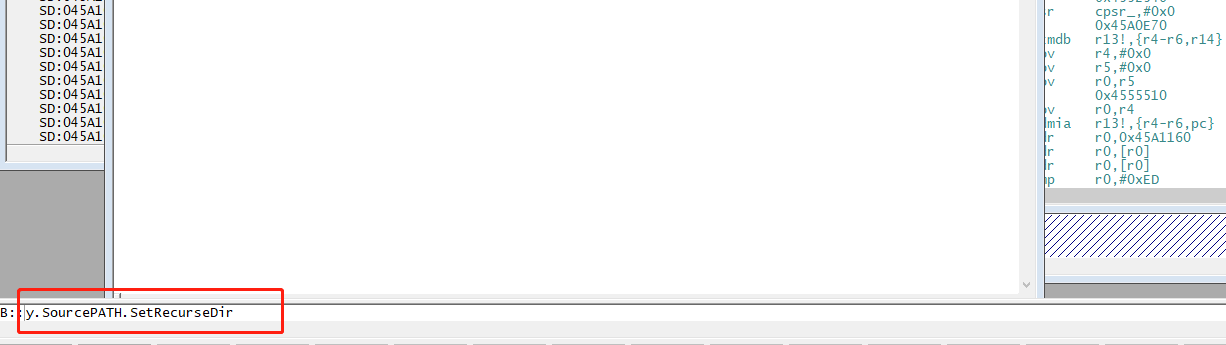
### 2.4.2 trace32工具使用

(a) Load bin(Memory)添加导出的bin文件

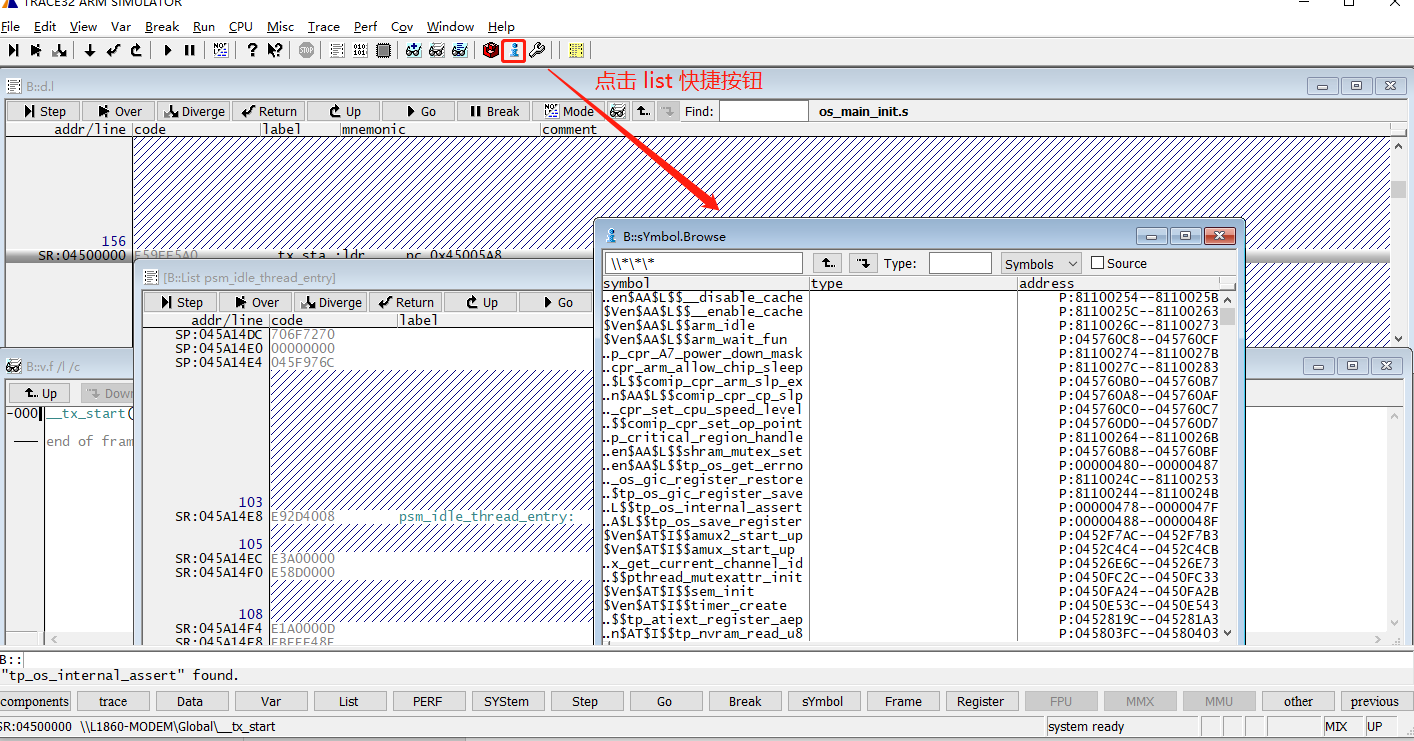


(b) Load axf/elf (code) 加载源码文件

(c) 导入数据后，设置y.sourcepath.SetRecurseDir 递归遍历源码目录

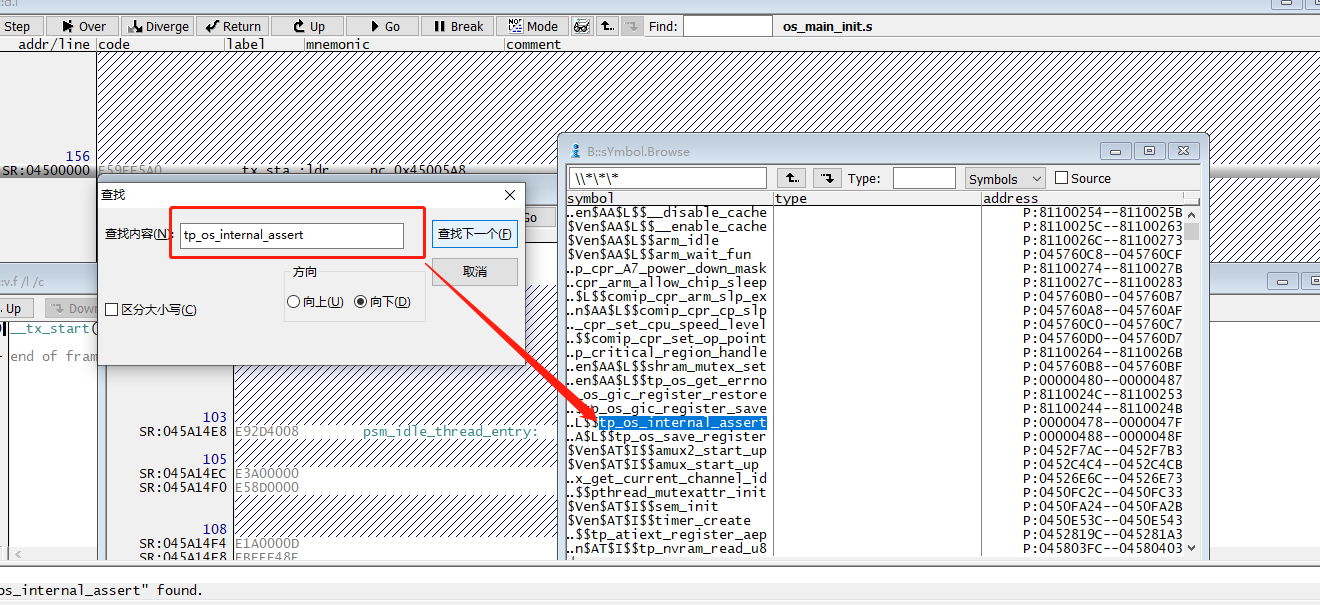
****

(d) 点击list Symbles按钮，将会列出源码所有的symbles, 包括函数、变量等

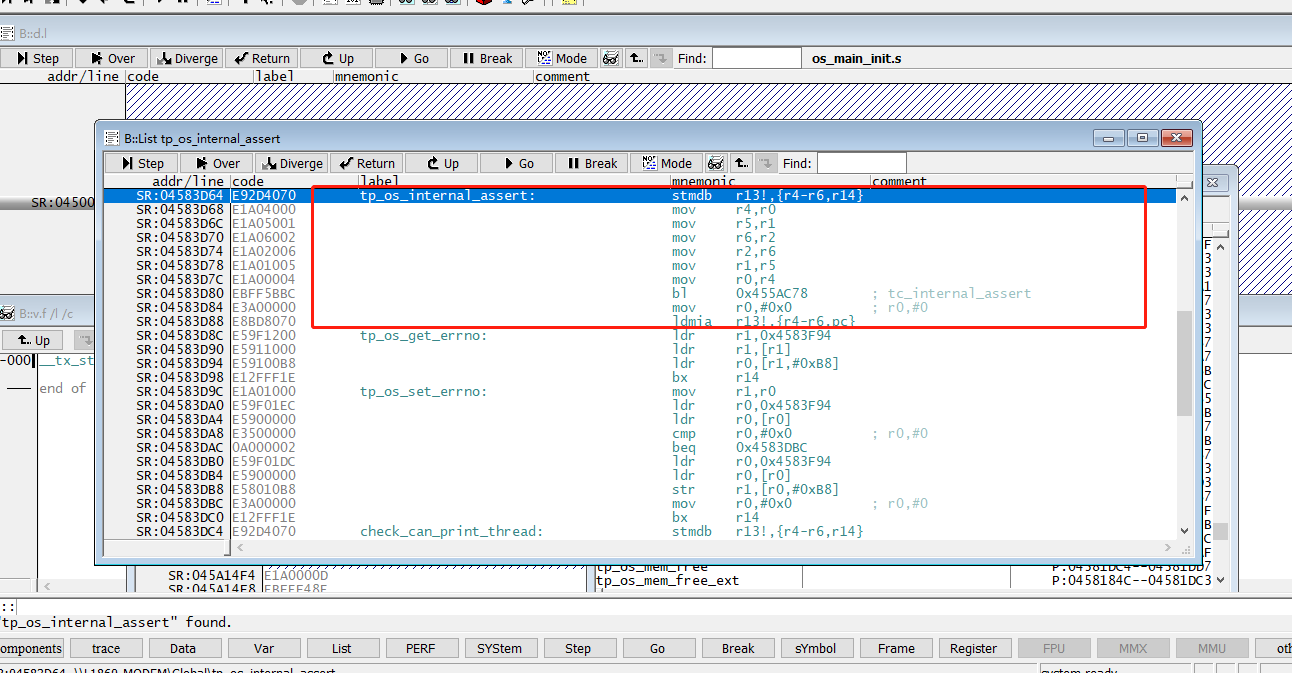


(e)使用ctrl + f查找自己想要查看的函数

**重要：**导入dump.bin 以及 1860.exl文件后，点击list symble后，找到自己想要查看的symble（变量、函数等），双击symble 名并记录。**需要在此处再次导入dump.bin文件**，输入记录的地址即可查看死机现场的值。

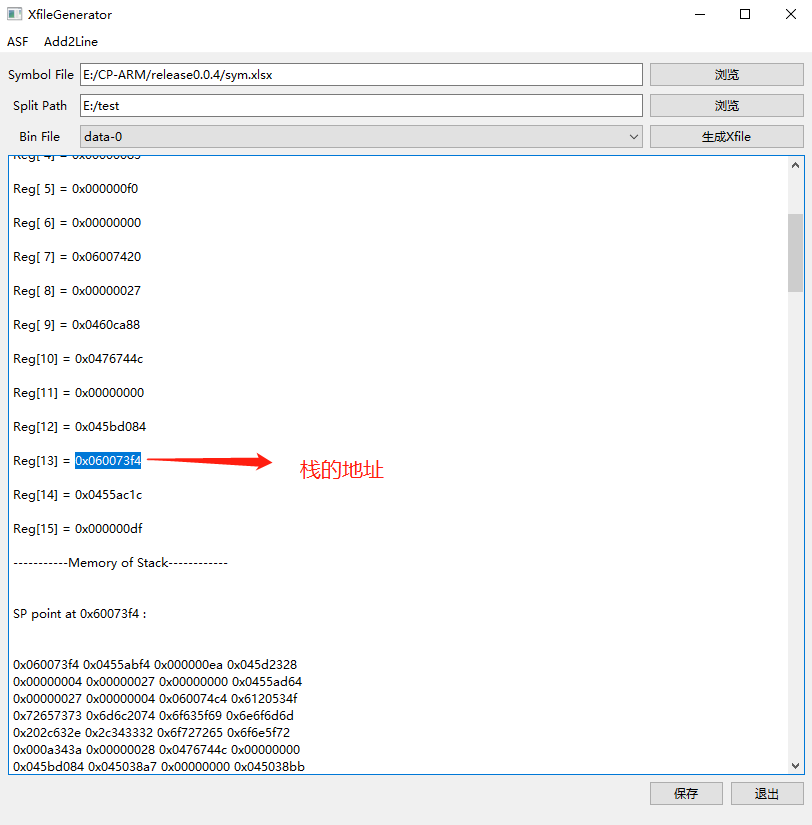


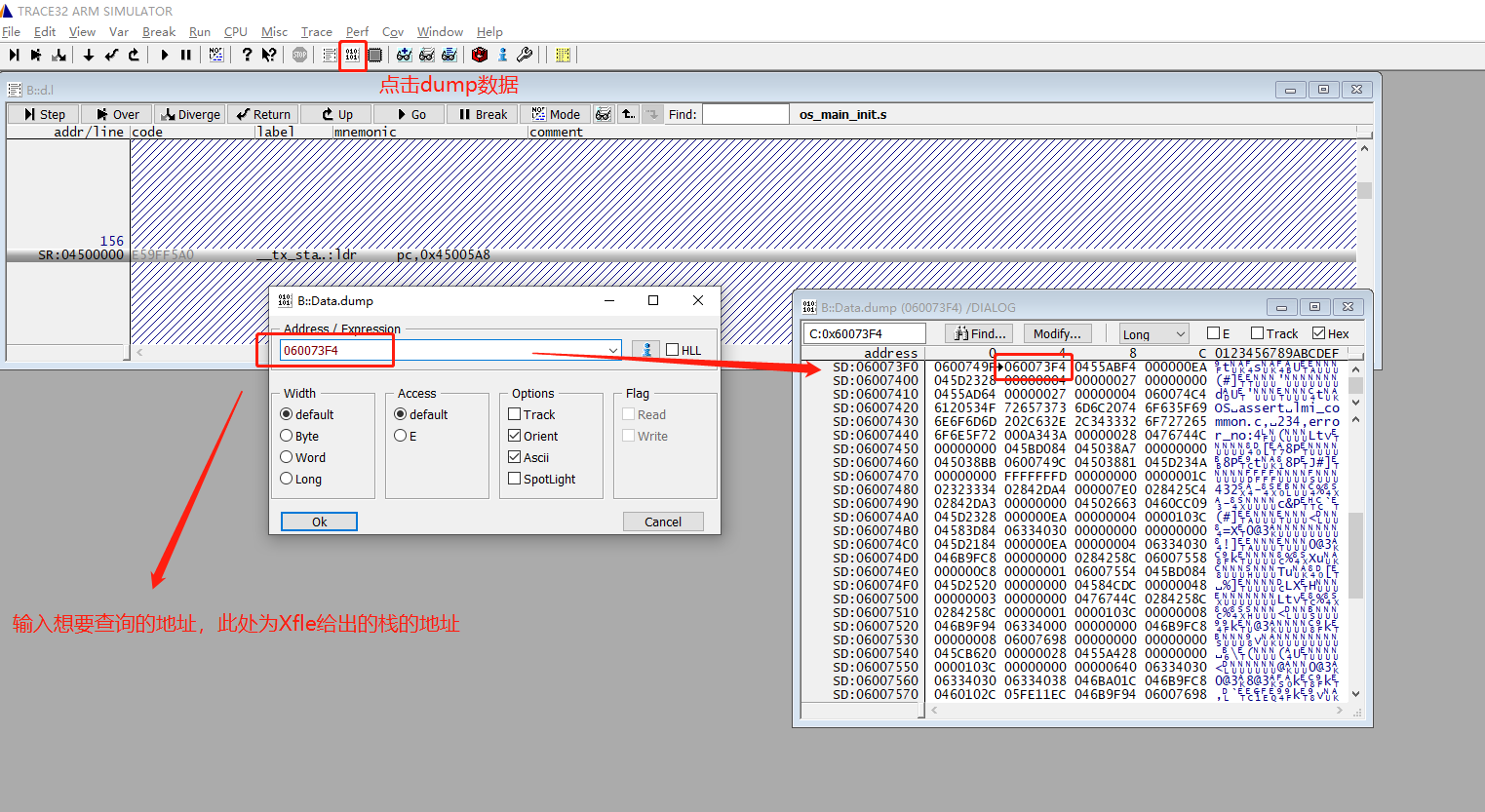
(f)双击函数名即可得到该函数的汇编代码。



**Trce32栈回溯示例：**

栈中保存了SP压栈记录，可以根据该值来进行系统运行过程的回溯，找到死机处。





找到栈顶后，以此往回追溯便可得到程序的运行过程，可以结合该处的汇编代码进行分析，协助发现问题。

**注：**

T32高级更多高级调试用法请参照附件t32\_demo.pdf

可以初步判定问题,找到分析问题的切入点

通过综合统计，得到最常出现的问题（Block issue,优先解决）

结合代码和logfile，加快问题定位和解决