crashed处理机制

## 背景：

在我们项目经常出现一些程序异常退出，却没有可以用的log去跟踪和定位异常点，我们需要一种机制来处理这样的问题，方便我们从根源上解决问题。

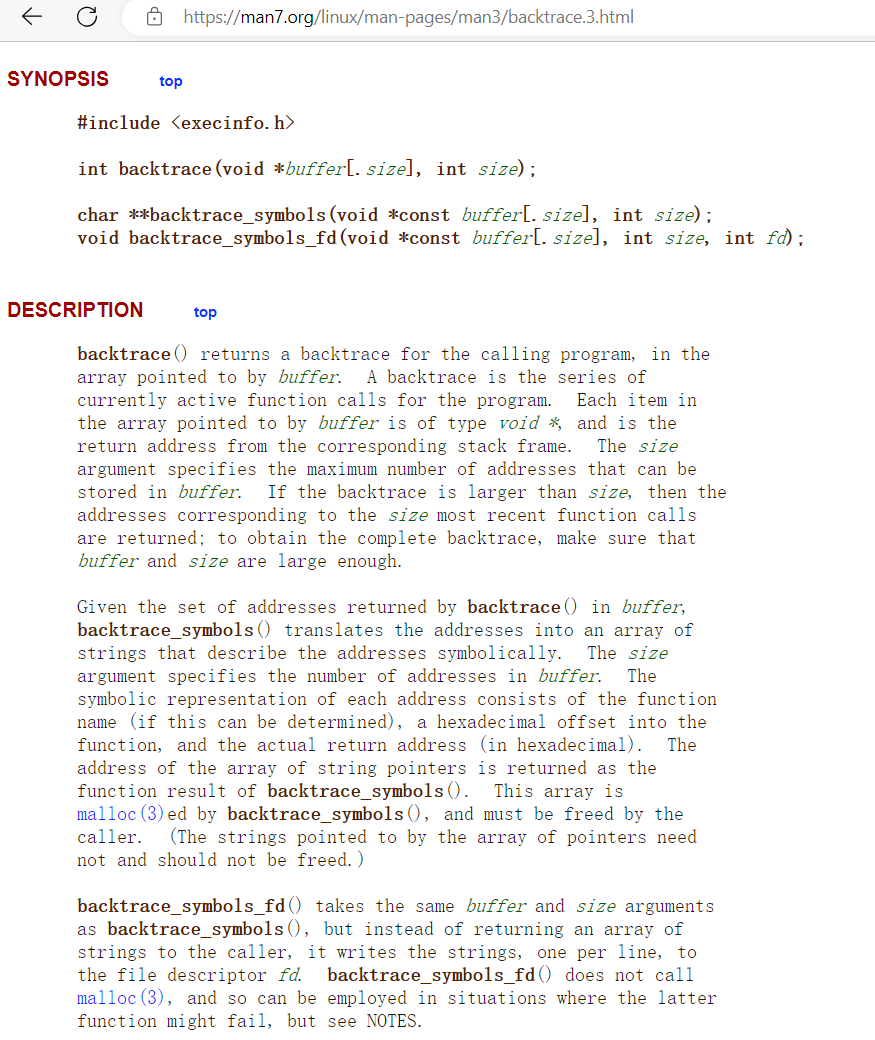
## 进程crashed一般调试方法：

1. **Gdb调试**
2. **Coredump + gdb**
3. **使用backtrace API**

前两种我们不做过多的解释，本方案的重点是使用用第三种方法。

## 理论依据：

1. **什么是backtrace**



backtrace就是回溯堆栈，简单的说就是可以列出当前函数调用关系。跟多理论基础看参考：

<https://blog.csdn.net/maimang1001/article/details/121253473>

<https://kernel.googlesource.com/pub/scm/java/jato/jato/+/master/sys/linux-arm/backtrace.c>

https://blog.csdn.net/jxgz\_leo/article/details/53458366?ydreferer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8%3D

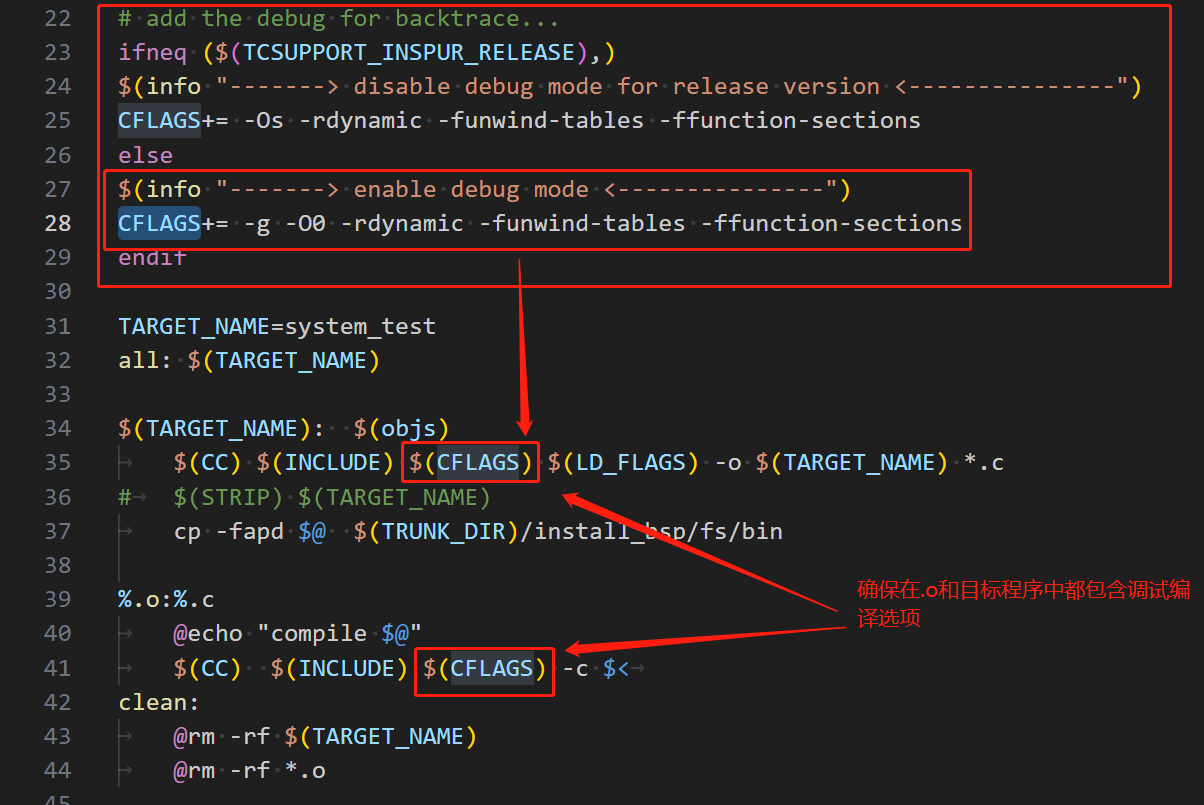
1. **使用的前提条件**

需要使用一些特殊的编译选项，方式gcc编译时过多优化导致backtrace和gdb没法获得更多的调试信息。

* + backtrace()的实现依赖于栈指针（fp寄存器），在gcc编译过程中任何非零的优化等级（-On参数）或加入了栈指针优化参数-fomit-frame-pointer后多将不能正确得到程序栈信息；
  + backtrace\_symbols的实现需要符号名称的支持，在gcc编译过程中需要加入-rdynamic参数；
  + 内联函数没有栈帧，它在编译过程中被展开在调用的位置；
  + 尾调用优化（Tail-call Optimization）将复用当前函数栈，而不再生成新的函数栈，这将导致栈信息不能正确被获取

所以我们需要在自己对应的项目makefile里面增加相关编译选项，具体参考如下所示：

（-g –O0 -rdynamic -funwind-tables -ffunction-sections）



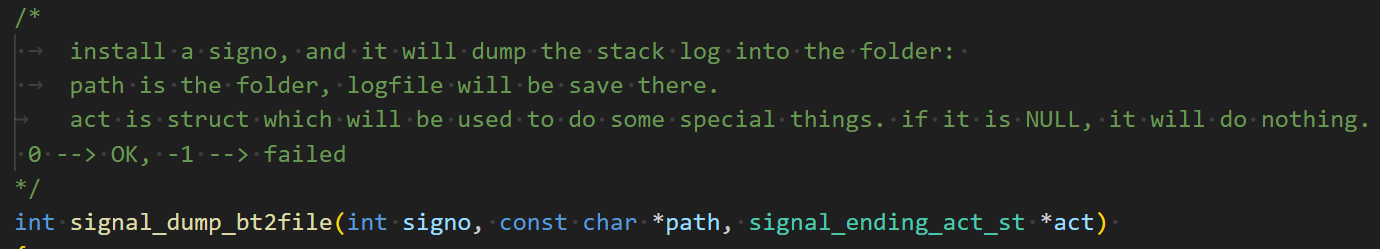
1. **如何处理crashed问题**

根据backtrace的作用，我们可以在收到异常信号的callback函数里面使用backtrace相关APIs保存异常时的堆栈信息，然后根据堆栈信息定位具体出问题的代码。

## 光AP中的实现：

1. **代码流程：**
   1. **实现信号处理和backtrace信息保存**

目前我们实现了一个API用于使用者在自己的程序里面安装信号处理的自定义行为。



用户只需要出入需要处理的signo和backtrace信息保存的路径和一个结构体指针(具体解释见下文)。当系统收到signo的信号时，就会触发异常信息保存流程。之后我们根据保存的异常信息分析问题点。

* 1. **实现额外处理机制(可选)**

如果用户想要做一些额外的事情，比如清理程序遗留资源等动作。用户可以传入act这个指针，其中包含一个callback和一个callback使用的入参。如果没有这个需求，可以直接传入NULL。

1. **具体测试和分析方法：**
   1. **测试代码：**

|  |  |
| --- | --- |
|  |  |

上面的代码模拟内存异常访问导致SIGSEGV的异常信号。

* 1. **测试用例分析：**

|  |  |  |
| --- | --- | --- |
| **编译选项** | **具体log** | **分析方法** |
| 包含-g –O0，不做strip | Slogd.log文件记录了backtrace log保存的地方，具体说明如下： | 主要分析的是log的栈调用关系，如左图log所示，我们可以看到问题是出在test3()这个函数里面，偏移地址为0x30，具体地址为0x10e0c。根据我们的测试代码，因为test3()没有几行代码很快就可以看出问题点，但是如果代码比较复杂，我们可以使用addr2line工具来把具体地址转换为具体的代码行号。如下所示：    异常地址定位到的代码为system\_helper.c:81，这个位置和上面的测试代码异常点完全一致。 |
| 包含-g –Os，不做strip |  | 编译使用了 –Os的优化编译，导致无法获得完整的堆栈信息。无法获得异常问题的有效信息。  但是这样的编译选项gdb还是可以定位到问题：    **Note： 不是所有的程序在优化编译后都可以获得完整的栈调用信息。只是我们这个测试程序比较简单的原因** |
| 其他strip或者优化的情况 |  | 对于一下特殊情况据需要使用更多工具辅助定位问题。 |