一种linux下检测内存泄露的方法

Table of Contents

1.	工具介绍	1
	1.1. valgring获取安装方法	
	1.2. 简单使用	
۷.	更多valgrind用法	4

1. 工具介绍

- · linux下用户态程序,经常会遇到动态分配内存,调用malloc之类的分配函数 有时候程序嵌套比较深或者结构比较复制,往往忘记free申请的内存,这样就造成 了内存泄露,排查起来颇为吃力。
 - · 工具 valgrind 可以帮我们预先检查函数是否存在内存泄露的风险

1.1. valgring获取安装方法

- · 获取工具
 - 。 valgrind官网地址: valgrind官网
 - 。 最新版本下载链接: valgring最新版
 - 下载后存放linux机器下,为valgrind-3.12.0.tar.bz2
 - tar jxf valgrind-3.12.0.tar.bz2 解压
 - 进入软件, ./autogen.sh;./configure;make;make install
 - 以上步骤安装在本机上,参照软件中的 READEME

1.2. 简单使用

· 写一个内存泄露的代码,如下:

```
#include <stdlib.h>
#include <stdlib.h>
#include <stdlib.h>
#include <stdlib.h>
#include <stdlib.h>
#include <stdlib.h>

#include <stdlib.h>

#include <stdlib.h>

#include <stdlib.h>

valgrind 3.12.0 (tar.bz2) [12MB] - 20 October 2016.

Fint8main(4) arm32, arm64,ppc32,ppc64le,ppc64be,s390x,mips32,mips64}-linux, {arm32, support for tilegx-linux.

md5: 6eh3f0c*pstl7013nlf22e,483d61bb

3.12.0 is a feature release with many improvements and the usual collection of bug fixes. Thi commopStlfpcie(charne), ma, 15.0c, (1.9.24) is many smaller refinements and new featurememcpy(pstr, "helloyou", sizeof("helloyou"));

Valkyrie 2.0.0

valkyrie printf(b2The66st) - 250%s\n 2,pstr);
md5: a-return 1543dae5f988de0160aeb5

Valkyrie is a Qt4-based GUI for the Valgrind 3.6.x and 3.7.x series, that works for the Memchavalgrind-3.6.0 and valgrind-3.7.0.
```

· gcc -o malloc malloc.c 编译代码,生成malloc程序

- · 执行检查 valgrind --leak-check=full ./malloc
- · 看到结果如下:

```
hdd@ubutu:~/my_test_progress/my_test_program $ valgrind --leak-check=full ./malloc
==26175== Memcheck, a memory error detector
==26175== Copyright (C) 2002-2015, and GNU GPL d, by Julian Seward et al.
==26175==uUsing bValgrind=3.12v0aand: LibVEX;narerun cwiths=hefonecopyrig<u>htuinfo:nts</u> mailing list.
==26175== Command: ./malloc
The str is helloyou
==26175==3.12.0 (tar.bz2) [12MB] - 20 October 2016.
==26175≌£aHEAParSUMMARY:ppc32,ppc64le,ppc64be,s390x,mips32,mips64}-linux, {arm32,arm64,x86,mips32}-android, {x86,amd6
==26175<sup>±±for tilegx</sup>inuxuse at exit: 1,024 bytes in 1 blocks
==26175==<sup>0.036</sup>101tal<sup>1</sup>heap<sup>7</sup>03age<sup>2,6</sup>2<sup>b</sup>allocs, 1 frees, 2,048 bytes allocated
=26175 1,024 bytes in 1 blocks are definitely lost in loss record 1 of 1 notes for details.
=26175==
                at 0x4C2DBB6: malloc (vg_replace_malloc.c:299)
 =26175==
                by 0x4005EF: main (in /home/hdd/my test progress/my test program/malloc)
=26175#i= 2.0.0
=26175== (LEAK ISUMMARY): - 21 October 2010.
 =26175==1dfb8definitelydlost:e01,024 bytes in 1 blocks
=26175== indirectly lost: 0 bytes in 0 blocks
=26175== is a Qtd-base sibly tost in 0 bytes in 0 blocks or the Memcheck and Helgrind tools. It also has an XN
=26175== still reachable: 0 bytes in 0 blocks
 =26175<del>aa</del>ion of Valkyrie <code>Suppressed</code> a <code>aoy</code> <code>bytesoin</code> itolocks 3.6.0. If you want to use Valkyrie with an older Valgrind vers
 =26175==
 =26175== For counts of detected and suppressed errors, rerun with: -v
==26175== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

可以看到其中检查到一处内存泄露

2. 更多valgrind用法

· 参考 valgrind用法