一、Linux内存检测

1、valgrind 检测内存泄漏

1. 、安装valgrind

1：tar -xf valgrind-3.9.0.tar.bz2

2：./configure

3：make

4：make install（如果失败，改用root权限）

5：valgrind ls -l（检查是否正常工作，如果出错，按照提示安装相应文件

1. 、Valgrind的最新版是3.11.0，它一般包含下列工具：

Memcheck：用于检测内存错误。它帮助c和c++的程序更正确。

Cachegrind：用于分析cache和分支预测。它帮助程序执行得更快。

Callgrind：用于函数调用的分析。

Helgrind：用于分析多线程。

DRD：也用于分析多线程。与Helgrind类似，但是用不同的分析技术，所以可以检测不同的问题。

Massif：用于分析堆。它帮助程序精简内存的使用。

SGcheck：检测栈和全局数组溢出的实验性工具，它和Memcheck互补使用。

--leak-check=yes打开内存泄露检测细节

--tool=helgrind使用Helgrind工具检测线程错误

假设hello是可执行程序：可能的使用方法如下

valgrind --leak-check=yes ./hello

或者valgrind ./hello

终端上显示检测信息。

1.Memcheck

最常用的工具，用来检测程序中出现的内存问题，所有对内存的读写都会被检测到，一切对malloc()/free()/new/delete的调用都会被捕获。所以，它能检测以下问题：

对未初始化内存的使用；

读/写释放后的内存块；

读/写超出malloc分配的内存块；

读/写不适当的栈中内存块；

内存泄漏，指向一块内存的指针永远丢失；

不正确的malloc/free或new/delete匹配；

memcpy()相关函数中的dst和src指针重叠。

2.Callgrind

和gprof类似的分析工具，但它对程序的运行观察更是入微，能给我们提供更多的信息。和gprof不同，它不需要在编译源代码时附加特殊选项，但加上调试选项是推荐的。Callgrind收集程序运行时的一些数据，建立函数调用关系图，还可以有选择地进行cache模拟。在运行结束时，它会把分析数据写入一个文件。callgrind\_annotate可以把这个文件的内容转化成可读的形式。

3.Cachegrind

Cache分析器，它模拟CPU中的一级缓存I1，Dl和二级缓存，能够精确地指出程序中cache的丢失和命中。如果需要，它还能够为我们提供cache丢失次数，内存引用次数，以及每行代码，每个函数，每个模块，整个程序产生的指令数。这对优化程序有很大的帮助。

4.Helgrind

它主要用来检查多线程程序中出现的竞争问题。Helgrind寻找内存中被多个线程访问，而又没有一贯加锁的区域，这些区域往往是线程之间失去同步的地方，而且会导致难以发掘的错误。Helgrind实现了名为“Eraser”的竞争检测算法，并做了进一步改进，减少了报告错误的次数。不过，Helgrind仍然处于实验阶段。

5.Massif

堆栈分析器，它能测量程序在堆栈中使用了多少内存，告诉我们堆块，堆管理块和栈的大小。Massif能帮助我们减少内存的使用，在带有虚拟内存的现代系统中，它还能够加速我们程序的运行，减少程序停留在交换区中的几率。

<https://jingyan.baidu.com/article/948f59247be1c8980ef5f96b.html>

Valgrind 调用上述工具的说明

valgrind在ARM64平台移植说明(以ubuntu为例)

<https://blog.csdn.net/m0_37874806/article/details/105590314>

1、安装说明

(1)、先安装automake sudo apt-get install automake

如果没有将交叉编译工具链添加到环境变量，可以不使用下面的export,

1. #直接在confgure使用CC指定交叉编译工具链的位置也可以

(2)、export CC=aarch64-linux-gnu-gcc

(3)、export LD=aarch64-linux-gnu-ld

(4)、export AR=aarch64-linux-gnu-ar

1. ./autogen.sh
2. ./configure --prefix=/your/path/to/install/valgrind --host=aarch64-linux --enable-only64bi

2、

1. Linux网络检测工具
2. <https://blog.csdn.net/qiangcuo6087/article/details/79084648>

1、网络性能测试工具，Iperf 灌包工具

测试TCP吞吐量

iperf是以C/S模式工作，运行于服务端和客户端下。服务端主要用于监听到达的测试请求，而客户端主要用于发起连接会话。因此要使用iperf，需要两台服务器，一台运行在服务器模式下，一台运行在客户端模式下

是一个基于Client/Server的网络性能测试工具，可以测试TCP、UDP和SCTP带宽质量，能够提供网络吞吐率信息，以及震动、丢包率，最大段和最大传输单元大小等统计信息，帮助我们测试网络性能，定位网络瓶颈。其中抖动和丢包率适应于UDP测试，带宽测试适应于TCP和UDP。

1. 怎么用

iperf详解

<https://www.cnblogs.com/klb561/p/9215952.html>

1. 、安装Iperf

wget <https://iperf.fr/download/source/iperf-3.1.3-source.tar.gz>

tar xzvf iperf-3.1.3-source.tar.gz

cd iperf-3.1.3/

./configure

make && make install

<https://bugxia.com/256.html>

1. 、有哪些作用
2. 、

（1）TCP方面

q 测试网络带宽。

q 支持多线程，在客户端与服务端支持多重连接。

q 报告MSS/MTU值的大小。

q 支持TCP窗口值自定义并可通过套接字缓冲。

（2）UDP方面

q 可以设置指定带宽的UDP数据流

q 可以测试网络抖动值、丢包数

q 支持多播测试

q 支持多线程，在客户端与服务端支持多重连接。

2、抓包工具 tcpdump

3、磁盘检测工具Iotop

(1)、linux系统下测试硬盘IO读写的工具，简单的说,iotop是一个用来监视磁盘I/O使用状况的 top 类工具，可监测到哪一个程序使用的磁盘IO的信息

(2)、安装yum

<https://www.cnblogs.com/yinzhengjie/p/9934260.html>

4、网络监视工具 iftop

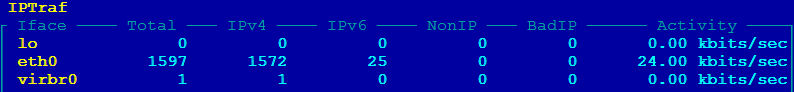
(1)、说明

类Unix系统中可以使用top查看系统资源、进程、内存占用等信息。查看网络状态可以使用netstat、nmap等工具。若要查看实时的网络流量，监控TCP/IP连接等，则可以使用iftop。

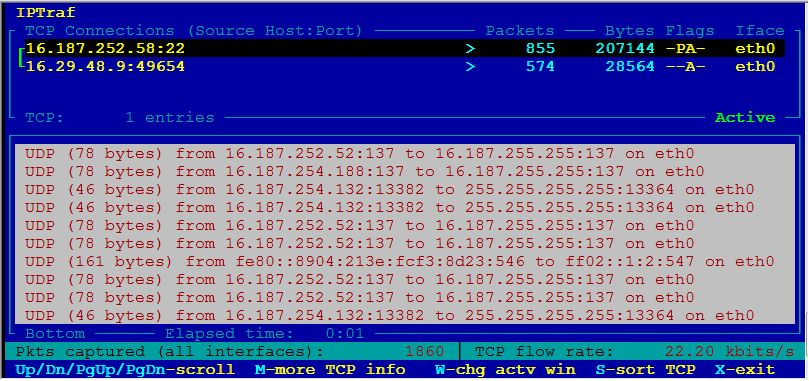
2、安装

# 5、iptraf:观察网络流量的工具<https://blog.csdn.net/qiangcuo6087/article/details/79084648>

(1)、看看总体状况，“iptraf -g” 显示每一个网卡上的流量



1. 、“iptraf -d eth0” 显示指定网卡上的流量统计，总体流量、流入量、流出量、以及按协议分类的流量统计：



1. 、进一步看看是哪些远程主机在跟我们的SSH端口通信，“iptraf -i eth0” ：

