valgrind的使用

https://wenku.baidu.com/view/408cc444767f5acfa1c7cda0.html

Valgrind是一个GPL的软件，用于Linux（For x86, amd64 and ppc32）程序的内存调试和代码剖析。你可以在它的环境中运行你的程序来监视内存的使用情况，比如C 语言中的malloc和free或者 C++中的new和 delete。使用Valgrind的工具包，你可以自动的检测许多内存管理和线程的bug，避免花费太多的时间在bug寻找上，使得你的程序更加稳固。 Valgrind的主要功能 Valgrind工具包包含多个工具，如Memcheck,Cachegrind,Helgrind, Callgrind，Massif。下面分别介绍个工具的作用：

Memcheck 工具主要检查下面的程序错误：  使用未初始化的内存 (Use of uninitialised memory)  使用已经释放了的内存 (Reading/writing memory after it has been free’d)  使用超过 malloc分配的内存空间(Reading/writing off the end of malloc’d blocks)  对堆栈的非法访问 (Reading/writing inappropriate areas on the stack)  申请的空间是否有释放 (Memory leaks – where pointers to malloc’d blocks are lost forever)  malloc/free/new/delete申请和释放内存的匹配(Mismatched use of malloc/new/new [] vs free/delete/delete [])  src和dst的重叠(Overlapping src and dst pointers in memcpy() and related functions)

Callgrind

Callgrind收集程序运行时的一些数据，函数调用关系等信息，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。 Cachegrind 它模拟 CPU中的一级缓存I1,D1和L2二级缓存，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。

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。

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。 Valgrind 安装 1、 到www.valgrind.org下载最新版valgrind-3.2.3.tar.bz2 2、 解压安装包：tar –jxvf valgrind-3.2.3.tar.bz2 3、 解压后生成目录valgrind-3.2.3 4、 cd valgrind-3.2.3 5、 ./configure 6、 Make;make install

Valgrind 使用 用法: valgrind [options] prog-and-args [options]: 常用选项，适用于所有Valgrind工具 1. -tool=<name> 最常用的选项。运行 valgrind中名为toolname的工具。默认 memcheck。 2. h –help 显示帮助信息。 3. -version 显示valgrind内核的版本，每个工具都有各自的版本。 4. q –quiet 安静地运行，只打印错误信息。 5. v –verbose 更详细的信息, 增加错误数统计。 6. -trace-children=no|yes 跟踪子线程? [no] 7. -track-fds=no|yes 跟踪打开的文件描述？[no] 8. -time-stamp=no|yes 增加时间戳到LOG信息? [no] 9. -log-fd=<number> 输出LOG到描述符文件 [2=stderr] 10. -log-file=<file> 将输出的信息写入到filename.PID的文件里，PID是运行程序 的进行ID 11. -log-file-exactly=<file> 输出LOG信息到 file 12. -log-file-qualifier=<VAR> 取得环境变量的值来做为输出信息的文件名。 [none] 13. -log-socket=ipaddr:port 输出LOG到socket ，ipaddr:port

LOG信息输出

1. -xml=yes 将信息以xml格式输出，只有memcheck可用 2. -num-callers=<number> show <number> callers in stack traces [12] 3. -error-limit=no|yes 如果太多错误，则停止显示新错误? [yes] 4. -error-exitcode=<number> 如果发现错误则返回错误代码 [0=disable] 5. -db-attach=no|yes 当出现错误，valgrind会自动启动调试器gdb。[no] 6. -db-command=<command> 启动调试器的命令行选项[gdb -nw %f %p] 适用于Memcheck工具的相关选项： 1. -leak-check=no|summary|full 要求对leak给出详细信息? [summary] 2. -leak-resolution=low|med|high how much bt merging in leak check [low] 3. -show-reachable=no|yes show reachable blocks in leak check? [no]