

`gdb`, `Valgrind`, `Apache bench`, `Siege` 使用介绍

```
docker commit id new-image
```

将容器打包成新的镜像

```
Docker rm -f id
```

删除容器

```
docker run --privileged -it -p 8888:9999 -v D:/2.MyFile/workspace/:/home/workspace -  
-name Liso new-image /bin/bash
```

用刚才打包成新的镜像，创建新的容器

`gdb`

安装gdb

```
apt-get install gdb
```

编译.c文件

```
gcc -g file.c -o exec_file
```

-g选项的作用是：在可执行文件中加入源码信息

比如：可执行文件中第几条机器指令对应源代码的第几行，但并不是把整个源文件都嵌入到可执行文件中，而是在调试时必须保证gdb能找到源文件。

gdb调试

```
gdb exec_file
```

```
root@75f74a718ed7:/home/tools/LisoTools# gdb binary_search
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from binary_search...done.
(gdb) _
```

命令	缩写	描述
run	r	从头开始运行程序
start		停在main函数第一行语句前面等待执行
quit	q	退出调试
next	n	执行下一行语句
step	s	进入函数内部单步执行（finish：结束当前函数并返回到调用点）
continue	c	从当前位置开始运行程序
list	l	查看源码，接着上次的位置向下显示10行 （list n：查看第n行源码；list 函数名：查看函数源码）
print	p	打印
break n	b	在第n行设置断点 （break 函数名：在函数开头设置断点）
info breakpoints num		显示num号断点信息
delete breakpoints num		删除num号断点
display var		跟踪查看变量，每次停下都显示其值
undisplay num		取消跟踪显示
set var		设置变量的值
watch var		设置观察点
info watchpoints		查看当前设置的观察点信息

Valgrind

安装valgrind

<https://www.valgrind.org/downloads/current.html>

mkdir download

cd download

wget <https://sourceware.org/pub/valgrind/valgrind-3.17.0.tar.bz2> 从网络上自动下载

tar -xjvf [valgrind-3.17.0.tar.bz2](#) 解压

cd valgrind-3.17.0

./configure 是源代码安装的第一步，主要的作用是对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系

make ./configure之后就有makefile出现，接着就是make编译

make install 安装

valgrind --version 查看版本，检查是否安装成功

valgrind --leak-check=yes ./exec_file

```
root@f2cd965ea222:/home/workspace/LisoTools# valgrind --leak-check=yes ./bi
==29== Memcheck, a memory error detector
==29== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==29== Using Valgrind-3.17.0 and LibVEX; rerun with -h for copyright info
==29== Command: ./binary_search
==29==
no
yes
==29==
==29== HEAP SUMMARY:
==29==   in use at exit: 0 bytes in 0 blocks
==29==   total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==29==
==29== All heap blocks were freed -- no leaks are possible
==29==
==29== For lists of detected and suppressed errors, rerun with: -s
==29== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

```
root@f2cd965ea222:/home/workspace/LisoTools# valgrind --leak-check=yes ./me
==72== Memcheck, a memory error detector
==72== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==72== Using Valgrind-3.17.0 and LibVEX; rerun with -h for copyright info
==72== Command: ./memcheck
==72==
==72== Invalid write of size 4
==72==   at 0x108668: f (memcheck.c:4)
==72==   by 0x108679: main (memcheck.c:9)
==72== Address 0x5235068 is 0 bytes after a block of size 40 alloc'd
==72==   at 0x4C32FB5: malloc (vg_replace_malloc.c:380)
==72==   by 0x10865B: f (memcheck.c:3)
==72==   by 0x108679: main (memcheck.c:9)
==72==
==72== HEAP SUMMARY:
==72==   in use at exit: 40 bytes in 1 blocks
==72==   total heap usage: 1 allocs, 0 frees, 40 bytes allocated
==72==
==72== 40 bytes in 1 blocks are definitely lost in loss record 1 of 1
==72==   at 0x4C32FB5: malloc (vg_replace_malloc.c:380)
==72==   by 0x10865B: f (memcheck.c:3)
==72==   by 0x108679: main (memcheck.c:9)
==72==
==72== LEAK SUMMARY:
==72==   definitely lost: 40 bytes in 1 blocks
==72==   indirectly lost: 0 bytes in 0 blocks
==72==   possibly lost: 0 bytes in 0 blocks
==72==   still reachable: 0 bytes in 0 blocks
==72==     suppressed: 0 bytes in 0 blocks
==72==
==72== For lists of detected and suppressed errors, rerun with: -s
==72== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```


Apache bench

ab V

```
root@dd7bac4fe823:/home# apt-get install apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2-utils is already the newest version (2.4.29-1ubuntu4.16).
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
root@dd7bac4fe823:/home# ab V
ab: invalid URL
Usage: ab [options] [http[s]://]hostname[:port]/path
Options are:
  -n requests      Number of requests to perform
  -c concurrency   Number of multiple requests to make at a time
  -t timelimit      Seconds to max. to spend on benchmarking
                   This implies -n 50000
  -s timeout        Seconds to max. wait for each response
                   Default is 30 seconds
  -b window size    Size of TCP send/receive buffer, in bytes
  -B address        Address to bind to when making outgoing connections
  -p postfile       File containing data to POST. Remember also to set -T
  -u putfile        File containing data to PUT. Remember also to set -T
  -T content-type   Content-type header to use for POST/PUT data, eg.
                   'application/x-www-form-urlencoded'
                   Default is 'text/plain'
  -v verbosity      How much troubleshooting info to print
  -w               Print out results in HTML tables
  -i               Use HEAD instead of GET
  -x attributes     String to insert as table attributes
```

Options are:

-n requests	Number of requests to perform
-c concurrency	Number of multiple requests to make at a time
-t timelimit	Seconds to max. to spend on benchmarking This implies -n 50000
-s timeout	Seconds to max. wait for each response Default is 30 seconds
-b window size	Size of TCP send/receive buffer, in bytes
-B address	Address to bind to when making outgoing connections
-p postfile	File containing data to POST. Remember also to set -T
-u putfile	File containing data to PUT. Remember also to set -T
-T content-type	Content-type header to use for POST/PUT data, eg. 'application/x-www-form-urlencoded' Default is 'text/plain'
-v verbosity	How much troubleshooting info to print
-w	Print out results in HTML tables
-i	Use HEAD instead of GET
-x attributes	String to insert as table attributes
-y attributes	String to insert as tr attributes
-z attributes	String to insert as td or th attributes

在测试会话中所执行的请求个数
并发数

-C attribute	Add cookie, eg. 'Apache=1234'. (repeatable)
-H attribute	Add Arbitrary header line, eg. 'Accept-Encoding: gzip' Inserted after all normal header lines. (repeatable)
-A attribute	Add Basic WWW Authentication, the attributes are a colon separated username and password.
-P attribute	Add Basic Proxy Authentication, the attributes are a colon separated username and password.
-X proxy:port	Proxyserver and port number to use
-V	Print version number and exit
-k	Use HTTP KeepAlive feature 启用HTTP KeepAlive功能，即在一个HTTP会话中执行多个请求。默认不启用KeepAlive
-d	Do not show percentiles served table.
-S	Do not show confidence estimators and warnings.
-q	Do not show progress when doing more than 150 requests
-l	Accept variable document length (use this for dynamic pages)
-g filename	Output collected data to gnuplot format file.
-e filename	Output CSV file with percentages served
-r	Don't exit on socket receive errors.
-m method	Method name
-h	Display usage information (this message)
-I	Disable TLS Server Name Indication (SNI) extension
-Z ciphersuite	Specify SSL/TLS cipher suite (See openssl ciphers)
-f protocol	Specify SSL/TLS protocol (SSL2, TLS1, TLS1.1, TLS1.2 or ALL)

```
root@dd7bac4fe823:/home# ab V
ab: invalid URL
Usage: ab [options] [http[s]://]hostname[:port]/path
```

ab -n 500 -c 10 -k https://www.baidu.com/

```
Benchmarking baidu.com (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests

Server Software:      bfe/1.0.8.18
Server Hostname:      baidu.com
Server Port:          443
SSL/TLS Protocol:     TLSv1.2, ECDHE-RSA-AES128-GCM-SHA256, 2048, 128
TLS Server Name:      baidu.com

Document Path:        /
Document Length:      161 bytes

Concurrency Level:     10
Time taken for tests:  0.473 seconds
Complete requests:     500
Failed requests:       0
Non-2xx responses:     500
Keep-Alive requests:   500

Total transferred:    178500 bytes
HTML transferred:    80500 bytes
Requests per second:  1057.20 [#/sec] (mean)
Time per request:     9.459 [ms] (mean)
Time per request:     0.946 [ms] (mean, across all concurrent requests)
Transfer rate:        368.57 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd]  median    max
Connect:      0        1   3.8        0     33
Processing:   6        7   9.3        7    214
Waiting:      6        7   9.3        7    213
Total:        6        8  10.0        7    214

Percentage of the requests served within a certain time (ms)
 50%    7
 66%    7
 75%    7
 80%    7
 90%    8
 95%    8
 98%   30
 99%   34
100%  214 (longest request)
```

ab -n 100 -c 10 -k http://127.0.0.1:9999/

Siege

```
root@dd7bac4fe823:/home/download/siege-4.1.1# siege
[alert] Zip encoding disabled; siege requires zlib support to enable it
SIEGE 4.1.1
Usage: siege [options]
       siege [options] URL
       siege -g URL

Options:
-V, --version          VERSION, prints the version number.
-h, --help             HELP, prints this section.
-C, --config           CONFIGURATION, show the current config.
-v, --verbose          VERBOSE, prints notification to screen.
-q, --quiet            QUIET turns verbose off and suppresses output.
-g, --get              GET, pull down HTTP headers and display the
                       transaction. Great for application debugging.
-p, --print            PRINT, like GET only it prints the entire page.
-c, --concurrent=NUM   CONCURRENT users, default is 10
-r, --reps=NUM         REPS, number of times to run the test.
-t, --time=NUMm        TIMED testing where "m" is modifier S, M, or H
                       ex: --time=1H, one hour test.
-d, --delay=NUM        Time DELAY, random delay before each request
-b, --benchmark        BENCHMARK: no delays between requests.
-i, --internet          INTERNET user simulation, hits URLs randomly.
-f, --file=FILE         FILE, select a specific URLS FILE.
-R, --rc=FILE           RC, specify an siegerc file
-l, --log[=FILE]        LOG to FILE. If FILE is not specified, the
                       default is used: PREFIX/var/siege.log
-m, --mark="text"       MARK, mark the log file with a string.
                       between .001 and NUM. (NOT COUNTED IN STATS)
-H, --header="text"     Add a header to request (can be many)
-A, --user-agent="text" Sets User-Agent in request
-T, --content-type="text" Sets Content-Type in request
-j, --json-output        JSON OUTPUT, print final stats to stdout as JSON
    --no-parser          NO PARSER, turn off the HTML page parser
    --no-follow           NO FOLLOW, do not follow HTTP redirects

Copyright (C) 2021 by Jeffrey Fulmer, et al.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE.

root@dd7bac4fe823:/home/download/siege-4.1.1#
```

SIEGE 4.1.1

Usage: siege [options]

siege [options] URL

siege -g URL

Options:

-V, --version

VERSION, prints the version number.

-h, --help

HELP, prints this section.

-C, --config

CONFIGURATION, show the current config.

-v, --verbose

VERBOSE, prints notification to screen.

-q, --quiet

QUIET turns verbose off and suppresses output.

-g, --get

GET, pull down HTTP headers and display the

transaction. Great for application debugging.

-p, --print

PRINT, like GET only it prints the entire page.

-c, --concurrent=NUM

CONCURRENT users, default is 10

-r, --reps=NUM

REPS, number of times to run the test.

-t, --time=NUMm

TIMED testing where "m" is modifier S, M, or H

ex: --time=1H, one hour test.

-d, --delay=NUM	Time DELAY, random delay before each request
-b, --benchmark	BENCHMARK: no delays between requests.
-i, --internet	INTERNET user simulation, hits URLs randomly.
-f, --file=FILE	FILE, select a specific URLS FILE.
-R, --rc=FILE	RC, specify an siegerc file
-l, --log[=FILE]	LOG to FILE. If FILE is not specified, the default is used: PREFIX/var/siege.log
-m, --mark="text"	MARK, mark the log file with a string. between .001 and NUM. (NOT COUNTED IN STATS)
-H, --header="text"	Add a header to request (can be many)
-A, --user-agent="text"	Sets User-Agent in request
-T, --content-type="text"	Sets Content-Type in request
-j, --json-output	JSON OUTPUT, print final stats to stdout as JSON
--no-parser	NO PARSER, turn off the HTML page parser
--no-follow	NO FOLLOW, do not follow HTTP redirects

siege -c 10 -r 1 https://www.baidu.com/

```
root@f2cd965ea222:/home# siege -c 10 -r 1 https://www.baidu.com/
** SIEGE 4.0.4
** Preparing 10 concurrent users for battle.
The server is now under siege...
Transactions:          290 hits
Availability:          100.00 %
Elapsed time:           2.03 secs
Data transferred:       3.74 MB
Response time:          0.07 secs
Transaction rate:       142.86 trans/sec
Throughput:             1.84 MB/sec
Concurrency:            9.85
Successful transactions: 290
Failed transactions:     0
Longest transaction:    1.06
Shortest transaction:    0.01
```

Transactions:	访问次数
Availability:	成功次数百分比
Elapsed time:	测试用时
Data transferred:	测试传输数据量
Response time:	平均响应时间
Transaction rate:	平均每秒处理请求数
Throughput:	吞吐量
Concurrency:	最高并发
Successful transactions:	成功的请求数
Failed transactions:	失败的请求数
Longest transaction	最长响应时间
Shortest transaction	最短响应时间

