strace/truss

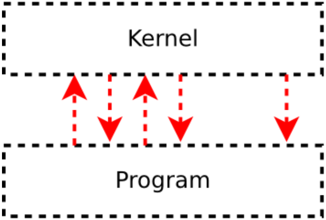
Angelo Haller

2013-05-06

1. What is strace/truss?
   - History
2. Why profile with strace/truss?
   - Collected data
3. Examples
4. Under the hood
5. ltrace
   - Examples
6. Sources

What is strace/truss?

`strace - trace system calls and signals`



```
open, read, write, close, ...
SIGKILL, SIGPIPE, SIGILL, ...
```

History

truss

- System V Release 4 (1989)
- Solaris
- FreeBSD

strace

- SunOS (1991)
- Linux

Implemented via ptrace in Linux and FreeBSD

Why profile with strace/truss?

- User space utility
- Light weight: `strace [options] command`
- No source code required!
- Ideal for quick debugging (e.g. environment for system administrators)
- Attachable to running processes (e.g. daemons)

Collected data

Input/Output

- System calls (name, parameters, return value)
- Signals
- Call and error count
- Time spent
- Environment variables
- Filters for file, network, ipc, . . .

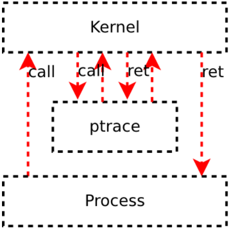
```
$ strace ./hello_world
execve("./hello_world", ["./hello_world"], [/* 53 vars */]) = 0
brk(0)                                  = 0x18f6000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or director
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 4
fstat(4, {st_mode=S_IFREG|0644, st_size=187142, ...}) = 0
mmap(NULL, 187142, PROT_READ, MAP_PRIVATE, 4, 0) = 0x7f508c659000
close(4)                                = 0
open("/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 4
read(4, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0␣\33\2\0\0\0\0\0"...
fstat(4, {st_mode=S_IFREG|0755, st_size=2035213, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0
mmap(NULL, 3852848, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 4, 0) =
mprotect(0x7f508c25c000, 2097152, PROT_NONE) = 0
mmap(0x7f508c45c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_
mmap(0x7f508c462000, 14896, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_
close(4)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0
arch_prctl(ARCH_SET_FS, 0x7f508c657700) = 0
mprotect(0x7f508c45c000, 16384, PROT_READ) = 0
mprotect(0x7f508c687000, 4096, PROT_READ) = 0
munmap(0x7f508c659000, 187142)          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 3), ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0
write(1, "Hello␣World!\n", 13)          = 13
exit_group(0)                           = ?
+++ exited with 0 +++
strace -c ./hello world
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  -nan    0.000000           0         1           read
  -nan    0.000000           0         1           write
  -nan    0.000000           0         2           open
  -nan    0.000000           0         2           close
  -nan    0.000000           0         3           fstat
  -nan    0.000000           0         8           mmap
  -nan    0.000000           0         3           mprotect
  -nan    0.000000           0         1           munmap
  -nan    0.000000           0         1           brk
  -nan    0.000000           0         1         1 access
  -nan    0.000000           0         1           execve
  -nan    0.000000           0         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                    25         1 total
```

```
strace -c ruby --help
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 74.18    0.005570           1     10691      9980 stat
 22.87    0.001717           1      1691      1244 open
  1.57    0.000118           0      2453           rt_sigprocmask
  0.48    0.000036           0       107           getegid
  0.47    0.000035           0      1102           lstat
  0.44    0.000033           0      1069           fstat
  0.00    0.000000           0       677           read
  0.00    0.000000           0        63           write
  0.00    0.000000           0       449           close
  0.00    0.000000           0       195           lseek
  0.00    0.000000           0        69           mmap
  0.00    0.000000           0        37           mprotect
  0.00    0.000000           0         5           munmap
  0.00    0.000000           0        96           brk
  0.00    0.000000           0        17           rt_sigaction
  0.00    0.000000           0       303       300 ioctl
  0.00    0.000000           0         2         2 access
  0.00    0.000000           0         1           pipe
  0.00    0.000000           0         1           clone
  0.00    0.000000           0         4         2 execve
  0.00    0.000000           0         6           fcntl
  0.00    0.000000           0         4           getdents
  0.00    0.000000           0         5           getcwd
  0.00    0.000000           0         6           getrlimit
  0.00    0.000000           0         1           getrusage
  0.00    0.000000           0       106           getuid
  0.00    0.000000           0       106           getgid
  0.00    0.000000           0       107           geteuid
  0.00    0.000000           0         1           sigaltstack
  0.00    0.000000           0         2           arch_prctl
  0.00    0.000000           0         1           futex
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         2           openat
  0.00    0.000000           0         1           set_robust_list
------ ----------- ----------- --------- --------- ----------------
100.00    0.007509                 19381     11528 total
```

Under the hood

ptrace - process trace

- system call: `ptrace(PTRACE_x, pid, ...)`
- process is run as a child of ptrace
- every system call is interrupted



ltrace

ltrace - A library call tracer

- trace dynamically linked library calls
- trace system calls and signals
- filters for library calls and libraries
- hooks into the library preloading mechanism
- no runtime linked library tracing

Available for Linux and FreeBSD

```
ltrace ./hello_world
__libc_start_main(0x400410, 1, 0x7fff90b1e398, 0x400520 <unfinished ..
puts("Hello␣World!") = 13
+++ exited (status 0) +++
ltrace slideextract -g test.mp4 out
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 44.58  224.948906        7259     30988 cvQueryFrame
 16.88   85.184030        2749     30987 cvCloneImage
 11.07   55.834884        1801     30987 cvSetImageROI
  8.22   41.454323        1337     30986 cvMatchTemplate
  6.00   30.261083         976     30986 cvCreateMat
  4.90   24.721057         797     30987 cvReleaseImage
  4.19   21.162885         682     30986 cvMinMaxLoc
  3.84   19.363947         624     30986 cvReleaseMat
  0.30    1.522337      101489        15 cvSaveImage
  0.02    0.117349      117349         1 cvCreateFileCapture
  0.00    0.009701         646        15 snprintf
  0.00    0.008979        8979         1 cvReleaseCapture
  0.00    0.008788         585        15 cvResetImageROI
  0.00    0.000400         200         2 __posix_getopt
  0.00    0.000224         224         1 __isoc99_sscanf
------ ----------- ----------- --------- --------------------
100.00  504.598893                247943 total
```

Sources

- https://www.freebsd.org/cgi/man.cgi?query=truss&manpath=FreeBSD+9.1-RELEASE
- https://www.freebsd.org/cgi/man.cgi?query=truss&manpath=SunOS+5.5.1
- http://docs.oracle.com/cd/E19082-01/819-2239/truss-1/index.html
- git://strace.git.sourceforge.net/gitroot/strace/strace tag: v4.7
  man strace
- https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.8.11.tar.xz
  man ptrace
- git://git.debian.org/git/collab-maint/ltrace.git tag: 0.7.2
  man ltrace
- http://www.linuxjournal.com/article/6100
- http://www.kernel.org/doc/ols/2007/ols2007v1-pages-41-52.pdf
- https://en.wikipedia.org/w/index.php?title=Strace&oldid=541397332
- https://en.wikipedia.org/w/index.php?title=Ptrace&oldid=551860101
- https://en.wikipedia.org/w/index.php?title=Ltrace&oldid=553254706
- http://www.unix.org/what_is_unix/history_timeline.html