strace
from upstream PoV

Dmitry Levin
ldv@altlinux.org

29.06.2013

strace: trace system calls and signals

Synopsis

strace [-CdffhiqrtttTvVxxy] [-I*n*] [-b*execve*] [-e*expr*]... [-a*column*] [-o*file*] [-s*strsize*] [-P*path*]... -p*pid*... / [-D] [-E*var*[=*val*]]... [-u*username*] *command* [*args*]
strace -c[df] [-I*n*] [-b*execve*] [-e*expr*]... [-O*overhead*] [-S*sortby*] -p*pid*... / [-D] [-E*var*[=*val*]]... [-u*username*] *command* [*args*]

Description

strace is a useful diagnostic, instructional, and debugging tool. In the simplest case strace runs the specified command until it exits. It intercepts and records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or to the file specified with the -o option.

Before revision control, 1991 – 1999

1991 – 1992, Paul Kranenburg

1.0 wrote strace for SunOS

1993, Branko Lankester

1.5 ported to Linux x86

1993 – 1996, Richard Sladkey

3.0 merged 2.5 for SunOS and 2nd release for Linux

3.1 ported to SVR4, Solaris, Irix;
Linux 2.0 (x86, alpha, m68k)

1996 – 1999, Wichert Akkerman

Debian packages: 3.1-1 – 3.1.0.1-12

Since revision control, 1999 – 2002, Wichert Akkerman

3.99 – 4.4

19.02.1999 introduced revision control (CVS)

18.03.1999 **3.99**

09.06.1999 **3.99.1**

09.07.1999 **4.0**: Linux powerpc, sparc, arm

26.11.1999 **4.1**: Linux mips

24.12.1999 **4.2**: Linux s390

01.03.2001 **4.3**: Linux ia64 and hppa, FreeBSD i386

19.08.2001 **4.4**: Linux ioctl parser

4.4.90 – 4.5.18

10.01.2003 – 17.07.2003 **4.4.90 – 4.4.99**

24.09.2003 **4.5**: Linux x86-64 biarch, Linux s390x, sh and sh64

13.11.2003 **4.5.1**: display multiple ioctl name matches on Linux

01.03.2004 **4.5.2**: Linux syscalls enhancements

16.04.2004 **4.5.3**: Linux syscalls: mq_*

03.06.2004 **4.5.4**: Linux ioctl update

27.06.2004 **4.5.5**: bug fixes

12.07.2004 **4.5.6**: Linux sparc64, Linux ioctl updates

31.08.2004 **4.5.7**: Linux *xattr and clock_* enhancements

19.10.2004 **4.5.8**: Linux syscalls: fadvise64, fadvise64_64, epoll_*, mbind, set_mempolicy, get_mempolicy, waitid

...

Since revision control, 2002 – 2009, Roland McGrath

4.4.90 – 4.5.18

04.02.2005	**4.5.9**: Linux ioctl and syscalls enhancements
14.03.2005	**4.5.10**: Linux signal decoding enhancements
22.03.2005	**4.5.11**: build fix
09.06.2005	**4.5.12**: mm fixes, x86-64 biarch enhancements, ppc updates, Linux aio enhancements
03.08.2005	**4.5.13**: "-e trace=desc" option
16.01.2006	**4.5.14**: accept numeric system calls in -e
11.01.2007	**4.5.15**: Linux syscalls: *at, inotify*, pselect6, ppoll, unshare
03.08.2007	**4.5.16**: Linux syscalls: move_pages, utimensat, signalfd, timerfd, eventfd, getcpu, epoll_pwait
21.07.2008	**4.5.17**: Linux arm improvements
28.08.2008	**4.5.18**: Linux syscalls enhancements
02.06.2009	CVS → GIT

strace 4.5.19, released 21.10.2009 by Roland McGrath

Noteworthy changes

- exit status transparency
- new architectures: Blackfin, AVR32, Cris
- new syscalls
- arm improvements
- syscalls enhancements
- socket type flags decoding

git commit count summary

```
$ git log v4.5.18..v4.5.19 |git shortlog -s |sort -k1,1nr |pr -t3w80
    64 Denys Vlasenko      4 Andreas Schwab          1 Frederik Schüler
    42 Dmitry V. Levin     2 Carlos O'Donell         1 Jakub Bogusz
    15 Roland McGrath      2 Paolo Bonzini           1 Jan Kratochvil
    12 Mike Frysinger      1 Edgar E. Iglesias       1 Kirill A. Shutemov
```

strace 4.5.20, released 13.04.2010

Noteworthy changes

- new option: -C
- new architecture: Tile
- new syscalls
- syscalls enhancements
- ioctls update

git commit count summary

```
$ git log v4.5.19..v4.5.20 |git shortlog -s |sort -k1,1nr |pr -t3w80
    19 Dmitry V. Levin     2 Frederik Schüler         1 Heiko Carstens
    11 Andreas Schwab      1 Bernhard Reutner-Fisher  1 Mark Wielaard
     3 Chris Metcalf       1 David Daney              1 Mike Frysinger
```

strace 4.6, released 15.03.2011

Noteworthy changes

- PTRACE_O_TRACECLONE et al support
- new architecture: MicroBlaze
- new syscalls
- syscalls enhancements
- ioctls enhancements
- biarch enhancements
- signal notification enhancements
- test suite

git commit count summary

```
$ git log v4.5.20..v4.6 |git shortlog -s |sort -k1,1nr |pr -t3w80
    67 Dmitry V. Levin      4 Carmelo Amoroso          1 David Daney
    12 Mike Frysinger       3 Holger Hans Peter        1 Edgar E. Iglesias
     9 Wang Chao            3 Sebastian Pipping        1 Frederik Schüler
     8 Andreas Schwab       2 Roland McGrath           1 Neil Campbell
```

strace 4.7, released 02.05.2012

Noteworthy changes

- new options: -y, -P, and -I
- process monitoring enhancements
- new architectures: x86-32, x86-64 multiarch
- multiarch enhancements
- new syscalls
- syscalls enhancements
- ioctls enhancements
- speed improvements
- non-Linux code finally removed

git commit count summary

```
$ git log v4.6..v4.7 |git shortlog -s |sort -k1,1nr |pr -t3w80
   251 Denys Vlasenko      3 Andreas Schwab          1 Grant Edwards
   116 Dmitry V. Levin     2 Andi Kleen              1 Heiko Carstens
    12 Mike Frysinger      1 Anton Blanchard         1 Sergei Trofimovich
    11 H.J. Lu             1 Damir Shayhutdinov
```

strace 4.8, released 03.06.3013

Noteworthy changes

- PTRACE_SEIZE support
- PTRACE_GETREGSET support
- new option: "-e trace=memory"
- new architectures: AArch64, Meta, OpenRISC 1000, TileGx, Xtensa
- multiarch enhancements
- new syscalls
- syscalls enhancements
- ioctls enhancements

git commit count summary

```
$ git log v4.7..v4.8 |git shortlog -s |sort -k1,1nr |pr -t3w80
    99 Denys Vlasenko      2 Andreas Schwab          1 Daniel P. Berrange
    85 Dmitry V. Levin     2 Ben Noordhuis           1 John Spencer
    21 Mike Frysinger      2 Bernhard Reutner-F      1 Maxin B. John
     7 Chris Metcalf       2 Chris Zankel            1 Namhyung Kim
     6 James Hogan         2 Stanislav Brabec        1 Zev Weiss
     3 Steve McIntyre      1 Christian Svensson
```

strace usage examples: -P, -e trace=file

```
$ strace -e file ls /var/empty
execve("/bin/ls", ["ls", "/var/empty"], [/* 32 vars */]) = 0
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
open("/lib64/libtinfo.so.5", O_RDONLY)  = 3
open("/lib64/libselinux.so.1", O_RDONLY) = 3
open("/lib64/librt.so.1", O_RDONLY)     = 3
open("/lib64/libcap.so.2", O_RDONLY)    = 3
open("/lib64/libacl.so.1", O_RDONLY)    = 3
open("/lib64/libc.so.6", O_RDONLY)      = 3
open("/lib64/libdl.so.2", O_RDONLY)     = 3
open("/lib64/libpthread.so.0", O_RDONLY) = 3
open("/lib64/libattr.so.1", O_RDONLY)   = 3
stat("/var/empty", {st_mode=S_IFDIR|0555, st_size=4096, ...}) = 0
open("/var/empty", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
+++ exited with 0 +++

$ strace -P /etc/ld.so.cache ls /var/empty
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=22446, ...}) = 0
mmap(NULL, 22446, PROT_READ, MAP_PRIVATE, 3, 0) = 0x2b7ac2ba9000
close(3)                                = 0
+++ exited with 0 +++
```

strace usage examples: -P, -v

```
$ strace -P /var/empty ls /var/empty
stat("/var/empty", {st_mode=S_IFDIR|0555, st_size=4096, ...}) = 0
open("/var/empty", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
fcntl(3, F_GETFD)                       = 0x1 (flags FD_CLOEXEC)
getdents(3, /* 2 entries */, 32768)     = 48
getdents(3, /* 0 entries */, 32768)     = 0
close(3)                                = 0
+++ exited with 0 +++
```

```
$ strace -P /var/empty -v ls /var/empty
stat("/var/empty", {st_dev=makedev(0, 30), st_ino=1020461,
  st_mode=S_IFDIR|0555, st_nlink=2, st_uid=0, st_gid=0, st_blksize=4096,
  st_blocks=8, st_size=4096, st_atime=2012/07/22-14:21:04,
  st_mtime=2012/05/17-19:22:58, st_ctime=2012/05/17-19:24:07}) = 0
open("/var/empty", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
fcntl(3, F_GETFD)                       = 0x1 (flags FD_CLOEXEC)
getdents(3, {{d_ino=799280, d_off=1551270678, d_reclen=24, d_name=".."}
  {d_ino=1020461, d_off=2147483647, d_reclen=24, d_name="."}}, 32768) = 48
getdents(3, {}, 32768)                  = 0
close(3)                                = 0
+++ exited with 0 +++
```

strace usage examples: -y, -e trace=

```
$ strace -e fstat,close -y ls /var/empty >/dev/null
fstat(3</etc/ld.so.cache>, {st_mode=S_IFREG|0644, st_size=22446, ...}) = 0
close(3</etc/ld.so.cache>)                = 0
fstat(3</lib/libtinfo.so.5.7>, {st_mode=S_IFREG|0644, st_size=135352, ...}) = 0
close(3</lib/libtinfo.so.5.7>)            = 0
fstat(3</lib/libselinux.so.1>, {st_mode=S_IFREG|0644, st_size=121992, ...}) = 0
close(3</lib/libselinux.so.1>)            = 0
fstat(3</lib/librt-2.11.3.so>, {st_mode=S_IFREG|0755, st_size=31792, ...}) = 0
close(3</lib/librt-2.11.3.so>)            = 0
fstat(3</lib/libcap.so.2.16>, {st_mode=S_IFREG|0644, st_size=23048, ...}) = 0
close(3</lib/libcap.so.2.16>)             = 0
fstat(3</lib/libacl.so.1.1.0>, {st_mode=S_IFREG|0644, st_size=35376, ...}) = 0
close(3</lib/libacl.so.1.1.0>)            = 0
fstat(3</lib/libc-2.11.3.so>, {st_mode=S_IFREG|0755, st_size=1452024, ...}) = 0
close(3</lib/libc-2.11.3.so>)             = 0
fstat(3</lib/libdl-2.11.3.so>, {st_mode=S_IFREG|0755, st_size=14776, ...}) = 0
close(3</lib/libdl-2.11.3.so>)            = 0
fstat(3</lib/libpthread-2.11.3.so>, {st_mode=S_IFREG|0755, st_size=138064, ...}
close(3</lib/libpthread-2.11.3.so>)       = 0
fstat(3</lib/libattr.so.1.1.0>, {st_mode=S_IFREG|0644, st_size=18704, ...}) = 0
close(3</lib/libattr.so.1.1.0>)           = 0
close(3</var/empty>)                      = 0
close(1</dev/null>)                       = 0
close(2</dev/pts/0>)                      = 0
+++ exited with 0 +++
```

strace usage examples: -y, -e trace=, -e read=

```
$ strace -e trace=read -e read=3 -y ls /var/empty
read(3</lib64/libtinfo.so.5.7>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\315\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  c0 cd 00 00 00 00 00 00  ..>..... ........ |
 | 00320  00 00 00 00 00 00 00 00  00 00 00 00 4d 00 00 00  ........ ....M... |
 | 00330  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........ |
read(3</lib64/libselinux.so.1>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\240W\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  a0 57 00 00 00 00 00 00  ..>..... .W...... |
 | 00320  40 20 00 00 20 00 00 80  c8 84 e2 00 00 12 00 03  @ .. ... ........ |
 | 00330  20 40 02 21 80 50 02 21  70 00 00 00 71 00 00 00   @.!.P.! p...q... |
read(3</lib64/librt-2.11.3.so>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200!\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  80 21 00 00 00 00 00 00  ..>..... .!...... |
 | 00320  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........ |
 | 00330  00 00 00 00 48 00 00 00  00 00 00 00 49 00 00 00  ....H... ....I... |
read(3</lib64/libcap.so.2.16>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\30\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  40 18 00 00 00 00 00 00  ..>..... @....... |
 | 00320  89 71 ee b2 ee 3e 3c d4  dd e7 a8 99 18 bf 5b 17  .q...><. ......[. |
 | 00330  7d dd 81 63 ed 16 0b 88  4d a7 3a ea f5 3e 3c d4  }..c.... M.:..><. |
read(3</lib64/libacl.so.1.1.0>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\240\37\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  a0 1f 00 00 00 00 00 00  ..>..... ........ |
 | 00320  47 00 00 00 00 00 00 00  48 00 00 00 4a 00 00 00  G....... H...J... |
 | 00330  00 00 00 00 4b 00 00 00  00 00 00 00 00 00 00 00  ....K... ........ |
read(3</lib64/libc-2.11.3.so>, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360\354\1\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 03  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  f0 ec 01 00 00 00 00 00  ..>..... ........ |
 | 00320  80 ca 44 42 28 00 06 80  10 18 42 00 20 40 80 00  ..DB(... ..B. @.. |
 | 00330  09 50 00 51 8a 40 10 00  00 00 00 08 00 00 11 10  .P.Q.@.. ........ |
read(3</lib64/libdl-2.11.3.so>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\340\r\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  e0 0d 00 00 00 00 00 00  ..>..... ........ |
 | 00320  91 21 fc f8 06 02 04 f9  fb 33 fb 0f f9 19 73 42  .!...... .3....sB |
 | 00330  fa 19 73 42 95 b3 5f 19  7f 9e d0 18 61 a2 92 06  ..sB.._. ....a... |
read(3</lib64/libpthread-2.11.3.so>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360Y\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  f0 59 00 00 00 00 00 00  ..>..... .Y...... |
 | 00320  01 05 00 50 20 a9 02 07  28 00 00 82 04 98 40 04  ...P ... (.....@. |
 | 00330  00 10 e0 54 00 02 40 02  02 10 c1 30 44 02 80 00  ...T..@. ...0D... |
read(3</lib64/libattr.so.1.1.0>, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\23\0\0\0\0\0\0"..., 832) = 832
 | 00000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00  .ELF.... ........ |
 | 00010  03 00 3e 00 01 00 00 00  b0 13 00 00 00 00 00 00  ..>..... ........ |
 | 00320  bf a8 e3 f8 db 0c 16 89  bb e3 92 7c c5 e8 1b 9b  ........ ...|.... |
 | 00330  05 c1 58 15 4b 3d 47 f3  91 78 a9 dd eb d3 ef 0e  ..X.K=G. .x...... |
+++ exited with 0 +++
```

strace usage examples: -r, -e trace=process

```
$ strace -r /bin/true
     0.000000 execve("/bin/true", ["/bin/true"], [/* 32 vars */]) = 0
     0.000281 exit_group(0)                  = ?
     0.000063 +++ exited with 0 +++

# strace -r -e process unshare -i /bin/true
     0.000000 execve("/usr/bin/unshare",
       ["/usr/bin/unshare", "-i", "/bin/true"], [/* 32 vars */]) = 0
     0.000899 arch_prctl(ARCH_SET_FS, 0x7f4e537cd700) = 0
     0.000398 unshare(CLONE_NEWIPC)          = 0
     0.000190 execve("/bin/true", ["/bin/true"], [/* 32 vars */]) = 0
     0.000186 exit_group(0)                  = ?
     0.028931 +++ exited with 0 +++
```

strace usage examples: -r, -T, -f, -e trace=process

```
$ strace -e process -r -T sh -c 'kill $$'
     0.000000 execve("/bin/sh", ["sh", "-c", "kill $$"], [/* 32 vars */]) = 0 <0.000361>
     0.001185 arch_prctl(ARCH_SET_FS, 0x2b0c3236b020) = 0 <0.000008>
     0.002239 --- SIGTERM {si_signo=SIGTERM, si_code=SI_USER, si_pid=12345, si_uid=500} ---
     0.000218 +++ killed by SIGTERM +++
```

```
$ strace -e process -f -q sh -c 'sleep 1 & pid=$!; sleep 0.1; kill $pid; wait'
execve("/bin/sh", ["sh", "-c", "sleep 1 & pid=$!; sleep 0.1; kil"...], [/* 32 vars */]) = 0
arch_prctl(ARCH_SET_FS, 0x2ae37beef020) = 0
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2ae37beef2f0) = 10001
[pid 10000] clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x2ae37beef2f0) = 10002
[pid 10001] execve("/bin/sleep", ["sleep", "1"], [/* 32 vars */] <unfinished ...>
[pid 10002] execve("/bin/sleep", ["sleep", "0.1"], [/* 32 vars */] <unfinished ...>
[pid 10001] <... execve resumed> )      = 0
[pid 10002] <... execve resumed> )      = 0
[pid 10000] wait4(-1,  <unfinished ...>
[pid 10001] arch_prctl(ARCH_SET_FS, 0x2b8cf7d49b20) = 0
[pid 10002] arch_prctl(ARCH_SET_FS, 0x2ada74416b20) = 0
[pid 10002] exit_group(0)               = ?
[pid 10002] +++ exited with 0 +++
[pid 10000] <... wait4 resumed> [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 10002
[pid 10000] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=10002, si_status=0, si_utime=0, si_stime=0} ---
[pid 10000] wait4(-1, 0x7fff7560e53c, WNOHANG, NULL) = 0
[pid 10001] --- SIGTERM {si_signo=SIGTERM, si_code=SI_USER, si_pid=10000, si_uid=600} ---
[pid 10001] +++ killed by SIGTERM +++
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_KILLED, si_pid=10001, si_status=SIGTERM, si_utime=1, si_stime=0} ---
wait4(-1, [{WIFSIGNALED(s) && WTERMSIG(s) == SIGTERM}], WNOHANG, NULL) = 10001
wait4(-1, 0x7fff7560e58c, WNOHANG, NULL) = -1 ECHILD (No child processes)
exit_group(0)                           = ?
+++ exited with 0 +++
```

strace usage examples: -ff, -ttt, -o, strace-log-merge

```
$ strace -e process -ff -ttt -o log sh -c 'sleep 1 & pid=$!; sleep 0.1; kill $pid; wait'
sh: line 1: 10001 Terminated              sleep 1
```

```
$ head -1 log.*
==> log.10000 <==
1342993484.722384 execve("/bin/sh", ["sh", "-c", "sleep 1 & pid=$!; sleep 0.1; kil"...], [/* 32 vars */]) = 0
==> log.10001 <==
1342993484.727498 execve("/bin/sleep", ["sleep", "1"], [/* 32 vars */]) = 0
==> log.10002 <==
1342993484.727422 execve("/bin/sleep", ["sleep", "0.1"], [/* 32 vars */]) = 0
```

```
$ strace-log-merge log
10000 1342993484.722384 execve("/bin/sh", ["sh", "-c", "sleep 1 & pid=$!; sleep 0.1; kil"...], [/* 33 vars */]) = 0
10000 1342993484.723369 arch_prctl(ARCH_SET_FS, 0x2ad5cc1fa020) = 0
10000 1342993484.725378 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
  child_tidptr=0x2ad5cc1fa2f0) = 10001
10000 1342993484.726783 clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
  child_tidptr=0x2ad5cc1fa2f0) = 10002
10000 1342993484.727188 wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 10002
10002 1342993484.727422 execve("/bin/sleep", ["sleep", "0.1"], [/* 32 vars */]) = 0
10001 1342993484.727498 execve("/bin/sleep", ["sleep", "1"], [/* 32 vars */]) = 0
10002 1342993484.769744 arch_prctl(ARCH_SET_FS, 0x2acee796db20) = 0
10001 1342993484.769845 arch_prctl(ARCH_SET_FS, 0x2b2bd019cb20) = 0
10002 1342993484.872233 exit_group(0)          = ?
10002 1342993484.872389 +++ exited with 0 +++
10000 1342993484.872492 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=10002, si_status=0, si_utime=0,
  si_stime=0} ---
10000 1342993484.872519 wait4(-1, 0x7fffe27a860c, WNOHANG, NULL) = 0
10001 1342993484.872666 --- SIGTERM {si_signo=SIGTERM, si_code=SI_USER, si_pid=10000, si_uid=600} ---
10000 1342993484.872795 wait4(-1, [{WIFSIGNALED(s) && WTERMSIG(s) == SIGTERM}], 0, NULL) = 10001
10001 1342993484.872849 +++ killed by SIGTERM +++
10000 1342993484.873117 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_KILLED, si_pid=10001, si_status=SIGTERM,
  si_utime=0, si_stime=0} ---
10000 1342993484.873140 wait4(-1, 0x7fffe27a81bc, WNOHANG, NULL) = -1 ECHILD (No child processes)
10000 1342993484.873339 exit_group(0)          = ?
10000 1342993484.873599 +++ exited with 0 +++
```

strace usage examples: -o pipeline

```
$ strace -e open -o '|grep /lib' ls /var/empty
open("/lib64/libtinfo.so.5", O_RDONLY)  = 3
open("/lib64/libselinux.so.1", O_RDONLY) = 3
open("/lib64/librt.so.1", O_RDONLY)     = 3
open("/lib64/libcap.so.2", O_RDONLY)    = 3
open("/lib64/libacl.so.1", O_RDONLY)    = 3
open("/lib64/libc.so.6", O_RDONLY)      = 3
open("/lib64/libdl.so.2", O_RDONLY)     = 3
open("/lib64/libpthread.so.0", O_RDONLY) = 3
open("/lib64/libattr.so.1", O_RDONLY)   = 3
```

```
$ strace -e desc -y -o "|grep '</[^l]'" ls /var/empty
fstat(3</etc/ld.so.cache>, {st_mode=S_IFREG|0644, st_size=22446, ...}) = 0
mmap(NULL, 22446, PROT_READ, MAP_PRIVATE, 3</etc/ld.so.cache>, 0) = 0x2ab097dfb
close(3</etc/ld.so.cache>)              = 0
ioctl(1</dev/pts/0>, SNDCTL_TMR_TIMEBASE or SNDRV_TIMER_IOCTL_NEXT_DEVICE or TC
ioctl(1</dev/pts/0>, TIOCGWINSZ, {ws_row=46, ws_col=128, ws_xpixel=1408, ws_ypi
fcntl(3</var/empty>, F_GETFD)           = 0x1 (flags FD_CLOEXEC)
getdents(3</var/empty>, /* 2 entries */, 32768) = 48
getdents(3</var/empty>, /* 0 entries */, 32768) = 0
close(3</var/empty>)                    = 0
close(1</dev/pts/0>)                    = 0
close(2</dev/pts/0>)                    = 0
```

strace usage examples: -p

```
$ sleep 1 & sleep 1 & sleep 0.1 &&
  strace -e process -p "$(pidof sleep)"
[1] 10000
[2] 10001
Process 10001 attached
Process 10000 attached
[pid 10000] exit_group(0)                       = ?
[pid 10001] exit_group(0)                       = ?
[pid 10001] +++ exited with 0 +++
[2]+  Done                    sleep 1
+++ exited with 0 +++
[1]-  Done                    sleep 1
```

strace usage examples: -c, -S

```
$ strace -c -S calls find /usr/share/doc/ > /dev/null
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  1.77    0.000023           0      6417         1 fcntl
  1.85    0.000024           0      1992           close
 93.83    0.001216           1       982           getdents
  1.70    0.000022           0       982           newfstatat
  0.00    0.000000           0       520           fstat
  0.85    0.000011           0       511           openat
  0.00    0.000000           0        60           write
  0.00    0.000000           0        23           mmap
  0.00    0.000000           0        14           mprotect
  0.00    0.000000           0         9           open
  0.00    0.000000           0         8           read
  0.00    0.000000           0         6           brk
  0.00    0.000000           0         6           fadvise64
  0.00    0.000000           0         3           munmap
  0.00    0.000000           0         3         2 ioctl
  0.00    0.000000           0         2           rt_sigaction
  0.00    0.000000           0         2         1 futex
  0.00    0.000000           0         1           rt_sigprocmask
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           uname
  0.00    0.000000           0         1           fchdir
  0.00    0.000000           0         1           gettimeofday
  0.00    0.000000           0         1           getrlimit
  0.00    0.000000           0         1           statfs
  0.00    0.000000           0         1           arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         1           set_robust_list
------ ----------- ----------- --------- --------- ----------------
100.00    0.001296                 11551         5 total
```

Linux ptrace API: tracer

- attachment and subsequent commands are per thread
- starts tracing a program:
  `fork` + `PTRACE_TRACEME` + `execve`
- attaches to existing threads:
  `PTRACE_ATTACH` or
  `PTRACE_SEIZE` + `PTRACE_INTERRUPT`
- receives ptrace events: `waitpid`
- commands tracees: `tgkill`,
  `PTRACE_SYSCALL` or `PTRACE_LISTEN`
- inspects tracees: `PTRACE_GETREGSET` or `PTRACE_GETREGS` or
  `PTRACE_PEEKUSER`,
  `PTRACE_GETSIGINFO`, `PTRACE_PEEKTEXT`, `PTRACE_PEEKDATA`
- detaches if necessary:
  `PTRACE_DETACH`

Plans and ideas for the future

plans: perpetual catching up with Linux kernel

- new ptrace extensions
- new architectures
- new decoders for new syscalls
- enhancing already existing decoders
- updating kernel constants

ideas

- test suite: cover more use cases
- more reliable decoders: don't trust the kernel
- more reliable multiarch

Contacts

homepage

http://sourceforge.net/projects/strace/

git repository

git://git.code.sf.net/p/strace/code.git

mailing list

strace-devel@lists.sourceforge.net