Действительно структурированный вывод в strace

1 октября 2016 г.

strace - trace system calls and signals

— man 1 strace

```
$ strace cat
execve("/bin/cat", ["cat"], [/* 40 vars */]) = 0
brk(0)                                  = 0x155e000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8691ab4000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=135526, ...}) = 0
mmap(NULL, 135526, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8691a92000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\34\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1729984, ...}) = 0
mmap(NULL, 3836448, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f86914ed000
mprotect(0x7f869168c000, 2097152, PROT_NONE) = 0
mmap(0x7f869188c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19f000) = 0x7f869
mmap(0x7f8691892000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f869189200
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8691a91000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8691a90000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8691a8f000
arch_prctl(ARCH_SET_FS, 0x7f8691a90700) = 0
mprotect(0x7f869188c000, 16384, PROT_READ) = 0
mprotect(0x60b000, 4096, PROT_READ)     = 0
mprotect(0x7f8691ab6000, 4096, PROT_READ) = 0
munmap(0x7f8691a92000, 135526)          = 0
brk(0)                                  = 0x155e000
brk(0x157f000)                          = 0x157f000
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=2855152, ...}) = 0
mmap(NULL, 2855152, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8691233000
close(3)                                = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
fadvise64(0, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
mmap(NULL, 139264, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8691a92000
read(0, "", 131072)                     = 0
munmap(0x7fb604be3000, 139264)          = 0
```

```
case Q_XGETQSTAT:
{
struct xfs_dqstats dq;

if (entering(tcp))
return 0;
if (umove_or_printaddr(tcp, data, &dq))
break;
tprintf("{version=%d, ", dq.qs_version);
if (abbrev(tcp)) {
tprints("...}");
break;
}
tprints("flags=");
printflags(xfs_quota_flags,
   dq.qs_flags, "XFS_QUOTA_???");
tprintf(", incoredqs=%u, ", dq.qs_incoredqs);
tprintf("u_ino=%" PRIu64 ", ", dq.qs_uquota.qfs_ino);
tprintf("u_nblks=%" PRIu64 ", ", dq.qs_uquota.qfs_nblks);
tprintf("u_nextents=%u, ", dq.qs_uquota.qfs_nextents);
tprintf("g_ino=%" PRIu64 ", ", dq.qs_gquota.qfs_ino);
tprintf("g_nblks=%" PRIu64 ", ", dq.qs_gquota.qfs_nblks);
tprintf("g_nextents=%u, ", dq.qs_gquota.qfs_nextents);
tprintf("btimelimit=%d, ", dq.qs_btimelimit);
tprintf("itimelimit=%d, ", dq.qs_itimelimit);
tprintf("rtbtimelimit=%d, ", dq.qs_rtbtimelimit);
tprintf("bwarnlimit=%u, ", dq.qs_bwarnlimit);
tprintf("iwarnlimit=%u}", dq.qs_iwarnlimit);
```

```
accept4(3, {sa_family=AF_UNIX, sun_path="accept4.socket.connect"},
[110->25], SOCK_NONBLOCK|SOCK_CLOEXEC) = 5

setitimer(ITIMER_REAL, {it_interval={0, 222222}, it_value={0, 111111}},
NULL) = 0

execve("./execve", ["./execve"], [/* 41 vars */]) = 0

listen(3<TCP:[4490622]>, 1) = 0

rt_sigprocmask(SIG_SETMASK, NULL, [HUP INT QUIT ALRM TERM], 8) = 0
```

```
global re_extract_unfinished
re_extract_unfinished \
= re.compile(r"\s*(\d+\.\d+ .*) <unfinished \.\.\.>$")

global re_extract_resumed
re_extract_resumed \
= re.compile(r"\s*(\d+\.\d+) <\.\.\. [\a-zA-Z\d]+ resumed>(.*)$")

global re_extract_signal
re_extract_signal \
= re.compile(r"\s*(\d+\.\d+) --- (\w+) \(([\w ]+)\) @ (\d)+ \((\d+)\) ---$")

global re_extract_arguments_and_return_value_none
re_extract_arguments_and_return_value_none \
= re.compile(r"\((.*)\)[ \t]*= (\?)$")

global re_extract_arguments_and_return_value_ok
re_extract_arguments_and_return_value_ok \
= re.compile(r"\((.*)\)[ \t]*= (-?\d+)$")

global re_extract_arguments_and_return_value_ok_hex
re_extract_arguments_and_return_value_ok_hex \
= re.compile(r"\((.*)\)[ \t]*= (-?0[xX][a-fA-F\d]+)$")

global re_extract_arguments_and_return_value_error
re_extract_arguments_and_return_value_error \
= re.compile(r"\((.*)\)[ \t]*= (-?\d+) (\w+) \([\w ]+\)$")

global re_extract_arguments_and_return_value_error_unknown
re_extract_arguments_and_return_value_error_unknown \
= re.compile(r"\((.*)\)[ \t]*= (\?) (\w+) \([\w ]+\)$")

global re_extract_arguments_and_return_value_ext
re_extract_arguments_and_return_value_ext \
= re.compile(r"\((.*)\)[ \t]*= (-?\d+) \(([^()]+)\)$")

global re_extract_arguments_and_return_value_ext_hex
re_extract_arguments_and_return_value_ext_hex \
```

```
# 16:08:17.082102 libpagemanager.so.1->mdb_env_create(0x6469c8, 0x7ff1e7c0d250,
0x646a48, 0x7ff1e7e41ea0) = 0 <0.000181>

normal= re.compile ('\s*%s +(?:%s\-\>)?%s\(%s\= %s %s' % (timestamp_parser,
    caller_parser, funcname_parser, params_parser, result_parser, time_parser))

# 16:08:17.192471 libDocumentAccess-Mh.so.1->MF_DeleteCollection(0x6490a8, 55,
0x6490a8, 0x7ff1e83c44e0 <unfinished ...>

unfinished= re.compile ('\s*%s +(?:%s\-\>)?%s\(%s\<unfinished \.\.\.\>' %
    (timestamp_parser, caller_parser, funcname_parser, params_parser))

# 16:08:17.203365 <... MF_DeleteCollection resumed> ) = 0x7ea000000000 <0.001550>

resumed= re.compile ('\s*%s +\<\.\.\. %s resumed\>.*\= %s %s' % (timestamp_parser,
funcname_parser, result_parser, time_parser))

# 11:44:55.470482  libpagemanager.so.1->mdb_txn_begin(0x646a80, 0, 0,
0x7fff6d7769f8 <no return ...>

# the point of...
no_return= re.compile ('\s*%s +(?:%s\-\>)?%s\(%s\<no return \.\.\.\>' %
    (timestamp_parser, caller_parser, funcname_parser, params_parser))

# anything, really, most likely
# 11:46:38.322997 +++ killed by SIGTERM +++
just_time= re.compile ('\s*%s' % timestamp_parser)
```

```
} elsif ($cmd eq $STAT) {
   if ($debug > 0) {
      printf(STRACE_LOG "FOUND A STAT!!!\n");
   }
   $CmdCounter{"stat"}++;
   $temp = floor($sec - $BeginTime) + 1;
   # Time in second referenced to beginning time
   $IOPS_Total[$temp]++;

   # Increment total IO time and command counters
   $IOTimeSum = $IOTimeSum + $elapsed_time;
   $IOTime_count++;
} elsif ($cmd eq $FSTAT) {
   if ($debug > 0) {
      printf(STRACE_LOG "FOUND A FSTAT!!!\n");
   }
   $CmdCounter{"fstat"}++;
   $temp = floor($sec - $BeginTime) + 1;
   # Time in second referenced to beginning time
   $IOPS_Total[$temp]++;

   # Increment total IO time and command counters
   $IOTimeSum = $IOTimeSum + $elapsed_time;
   $IOTime_count++;
} elsif ($cmd eq $STAT64) {
   if ($debug > 0) {
      printf(STRACE_LOG "FOUND A STAT64!!!\n");
   }
```

```
$ cat rpm-utils/spp
#!/bin/awk -f
#
# Copyright (C) 2003  Dmitry V. Levin <ldv@altlinux.org>
# Copyright (C) 2007  Alexey Tourbin <at@altlinux.org>
#
# The strace post processor.
#
# This program is free software; you can redistribute it and/or mod
# it under the terms of the GNU General Public License as published
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1
```

```
/^[1-9][0-9]*[[:space:]]+.* <unfinished \.\.\.>$/ {
    pid = $1;
    sub(" <unfinished \\.\\.\\.>$", "");
    trace[pid] = $0;
    next;
}

/^[1-9][0-9]*[[:space:]]+<\.\.\. [[:alnum:]_]+ resumed>/ {
    pid = $1;
    if (trace[pid] != "")
        sub("^[1-9][0-9]*[[:space:]]+<\\.\\.\\. [[:alnum:]_]+ resumed>", trace[pid]
}

match($0, /^[1-9][0-9]*[[:space:]]+([[:alnum:]_]+at)\([^",]+, "(\/[^"]*)", .*\) +=
    delete trace[$1];
    output(ary[1], ary[2]);
    next;
}

match($0, /^[1-9][0-9]*[[:space:]]+([[:alnum:]_]+)\("(\/[^"]*)", .*\) += [[:alnum:]
    delete trace[$1];
    output(ary[1], ary[2]);
    next;
}

/^[1-9][0-9]*[[:space:]]+/ { delete trace[$1]; }
```

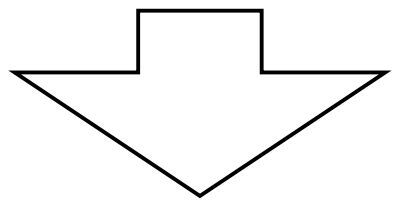
```
$ strace -etrace=pwritev -ewrite=1 -s2 ./preadv-pwritev
...
pwritev(1, [{iov_base="01"..., iov_len=3}, {iov_base="34"..., iov_len=5}, ...],
3, 0) = 15
 * 3 bytes in buffer 0
 | 00000  30 31 32                                          012             |
 * 5 bytes in buffer 1
 | 00000  33 34 35 36 37                                    34567           |
 * 7 bytes in buffer 2
 | 00000  38 39 61 62 63 64 65                              89abcde         |
```

```
$ strace -c sync                                                16-10-01 2:57:07
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00    0.012000       12000         1           sync
  0.00    0.000000           0         1           read
  0.00    0.000000           0         3           open
  0.00    0.000000           0         5           close
  0.00    0.000000           0         3           fstat
  0.00    0.000000           0         7           mmap
  0.00    0.000000           0         4           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         3         3 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.012000                    33         3 total
```

```
SYS_FUNC(swapon)
{
    unsigned int flags = tcp->u_arg[1];
    unsigned int prio = flags & SWAP_FLAG_PRIO_MASK;
    flags &= ~SWAP_FLAG_PRIO_MASK;

    printpath(tcp, tcp->u_arg[0]);
    tprints(", ");
    if (flags) {
        printflags(swap_flags, flags, "SWAP_FLAG_???");
        if (prio)
            tprintf("|%u", prio);
       } else {
           tprintf("%u", prio);
       }

    return RVAL_DECODED;
}
```

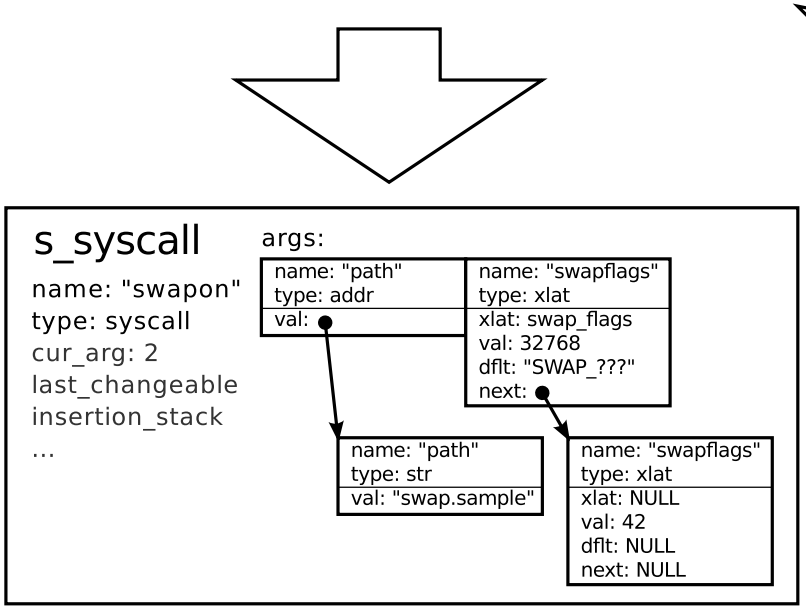


```
SYS_FUNC(swapon)
{
    s_push_path("path");
    s_push_xlat_flags_int("swapflags", swap_flags,
        NULL, SWAP_FLAG_PRIO_MASK,
        "SWAP_FLAG_???", NULL, false, 0);

    return RVAL_DECODED;
}
```

```
SYS_FUNC(swapon)
{
    s_push_path("path");
    s_push_xlat_flags_int("swapflags", swap_flags,
        NULL, SWAP_FLAG_PRIO_MASK,
        "SWAP_FLAG_???", NULL, false, 0);

    return RVAL_DECODED;
}
```



Legacy:

```
swapon("swap.sample", SWAP_FLAG_PREFER|42) =
    -1 EPERM (Operation not permitted)
```

JSON:

```
{
    "name": "swapon", "type": "syscall",
    "return": -1, "errno": 1, "errnostr": "EPERM",
    "retstring": "Operation not permitted",
    "args": [ {
            "name": "path", "type": "address",
            "addr": 4199168,
            "value": { "name": "path", "type": "str",
                    "value": "swap.sample" }
        }, {
            "name": "swapflags", "type": "xlat",
            "value": [
                { "default": false, "value": 32768,
                        "str": "SWAP_FLAG_PREFER" },
                { "default": false, "value": 42 }
] } ] }
```

```
{
    "name": "getrandom",
    "type": "syscall",
    "args": [
        {
            "name": "buf",
            "type": "changeable",
            "entering_value": null,
            "exiting_value": {
                "name": "buf",
                "type": "address",
                "addr": 140722827922048,
                "value": {
                    "name": "buf",
                    "type": "str",
                    "value": "\\x26\\x4d\\x4e",
                    "size": 3,
                    "truncated": true
                }
            }
        },
        {
            "name": "count",
            "value": 3
        },
        {
            "name": "flags",
            "type": "xlat",
            "value": [
                {
                    "default": true,
                    "value": 0
                }
            ]
        }
    ],
    "return": 3
}
```

- Уменьшились избыточность и потенциал для багов

- Уменьшились избыточность и потенциал для багов
- Теперь везде указываются имена аргументов
- Комментарии к аргументам

- Уменьшились избыточность и потенциал для багов
- Теперь везде указываются имена аргументов
- Комментарии к аргументам
- Исчезла необходимость в неравной борьбе

- Уменьшились избыточность и потенциал для багов
- Теперь везде указываются имена аргументов
- Комментарии к аргументам
- Исчезла необходимость в неравной борьбе
- `-z`

- Уменьшились избыточность и потенциал для багов
- Теперь везде указываются имена аргументов
- Комментарии к аргументам
- Исчезла необходимость в неравной борьбе
- `-z`
- Out и in-out аргументы реализуются проще

https://github.com/lineprinter/strace/tree/structured