strace Syscall tracer

CONCEPTS

strace prints every system call a process makes: name, arguments, return value. Output goes to stderr.

Your program traced process
↓ syscall
strace ptrace stops it, logs
↓ resume
Kernel does the work

Read lines as name(args) = result; failures end in -1 plus an errno such as ENOENT.

RUN AND ATTACH

`strace <cmd> [args]`
Trace a new command

`strace -p <pid>`
Attach to a running process

`strace -f <cmd>`
Follow forks, clones and threads

`strace -o <file> <cmd>`
Write the trace to a file

`strace -ff -o <prefix> <cmd>`
One output file per process

FILTER SYSCALLS

`strace -e trace=<name>,<name> <cmd>`
Only these syscalls

`strace -e trace=%file <cmd>`
Syscalls taking a file name

`strace -e trace=%network <cmd>`
Network syscalls

`strace -e trace=%process <cmd>`
Fork, exec and exit

`strace -e trace=%desc <cmd>`
File descriptor syscalls

`strace -e trace=%memory <cmd>`
Memory mapping syscalls

`strace -e trace=!<name> <cmd>`
Exclude a syscall

FILTER SYSCALLS (cont.)

`strace -P <path> <cmd>`
Only calls touching this path

`strace -Z <cmd>`
Show only failed calls

OUTPUT FORMAT

`strace -s <n> <cmd>`
Print strings up to n bytes

`strace -y <cmd>`
Show paths for file descriptors

`strace -yy <cmd>`
Also show socket addresses and devices

`strace -v <cmd>`
Print full structures, no abbreviation

`strace -e write=<fd> <cmd>`
Hex dump of data written to fd

`strace -e read=<fd> <cmd>`
Hex dump of data read from fd

TIMING AND STATISTICS

`strace -c <cmd>`
Per-syscall summary instead of a trace

`strace -C <cmd>`
Summary plus the normal trace

`strace -c -S calls <cmd>`
Sort the summary by call count

`strace -T <cmd>`
Show time spent in each syscall

`strace -tt <cmd>`
Wall-clock time with microseconds

`strace -ttt <cmd>`
Unix epoch time with microseconds

`strace -r <cmd>`
Time since the previous syscall

TAMPERING

`strace --fault=<call>:error=<ERRNO> <cmd>`
Make a syscall fail