ltrace Library call tracer

CONCEPTS

ltrace shows calls into shared libraries (libc, libssl...) and prints their arguments. Use strace for kernel syscalls only.

Your program main binary
Shared libraries ltrace: library calls
Kernel strace: syscalls (ltrace -S)

Only calls through the PLT are seen. Statically linked programs and calls inside one library are invisible unless you use -x.

Output goes to stderr, so redirect with 2> or use -o. Without -f a forking program is traced only in the parent.

RUN AND ATTACH

ltrace *<cmd>* **[args]**
Trace a new command

ltrace -p *<pid>*
Attach to a running process

ltrace -f *<cmd>*
Follow forks and clones

ltrace -p *<pid>* **-p** *<pid2>*
Attach to several processes

ltrace -o *<file>* *<cmd>*
Write the trace to a file

ltrace -h / -V
Show help / version

FILTER CALLS

ltrace -e *<func>* *<cmd>*
Trace only this function

FILTER CALLS (cont.)

ltrace -e *<f1>*+*<f2>* *<cmd>*
Trace several functions

ltrace -e @libc.so* *<cmd>*
Only calls into libc

ltrace -e *<func>*@*<lib>* *<cmd>*
Function from a given library

ltrace -e '/^str' *<cmd>*
Regex on function names

ltrace -e -*<func>* *<cmd>*
Trace everything except this

ltrace -l *<lib>* *<cmd>*
Calls into libraries matching a pattern

ltrace -S *<cmd>*
Show system calls as well

OUTPUT FORMAT

ltrace -s *<n>* *<cmd>*
Longest printed string (default 32)

ltrace -C *<cmd>*
Demangle C++ names

TIMING AND STATISTICS

ltrace -c *<cmd>*
Count calls and time per function

ltrace -T *<cmd>*
Time spent inside each call

ltrace -r *<cmd>*
Relative timestamp per line

ltrace -t *<cmd>*
Time of day per line

ltrace -c -S *<cmd>*
Summary including system calls