NAME

ltrace − A library call tracer

SYNOPSIS

ltrace [−e *filter*|−L] [−l|−−library=*library_pattern*] [−x *filter*] [−S] [−b|−−no-signals] [−i] [−w|−−where=*nr*] [−r|−t|−tt|−ttt] [−T] [−F *filename*] [−A *maxelts*] [−s *strsize*] [−C|−−demangle] [−a|−−align *column*] [−n|−−indent *nr*] [−o|−−output *filename*] [−D|−−debug *mask*] [−u *username*] [−f] [−p *pid*] [[−−] *command [arg ...]*]

ltrace -c [−e *filter*|−L] [−l|−−library=*library_pattern*] [−x *filter*] [−S] [−o|−−output *filename*] [−f] [−p *pid*] [[−−] *command [arg ...]*]

ltrace −V|−−version

ltrace −h|−−help

DESCRIPTION

ltrace is a program that simply runs the specified *command* until it exits. It intercepts and records the dynamic library calls which are called by the executed process and the signals which are received by that process. It can also intercept and print the system calls executed by the program.

Its use is very similar to **strace(1)**.

OPTIONS

−a, −−align *column*
: Align return values in a specific *column* (default column is 5/8 of screen width).

−A *maxelts*
: Maximum number of array elements to print before suppressing the rest with an ellipsis ("..."). This also limits number of recursive structure expansions.

−b, −−no-signals
: Disable printing of signals recieved by the traced process.

−c
: Count time and calls for each library call and report a summary on program exit.

−C, −−demangle
: Decode (demangle) low-level symbol names into user-level names. Besides removing any initial underscore prefix used by the system, this makes C++ function names readable.

−D, −−debug mask
: Show debugging output of **ltrace** itself. *mask* is a number with internal meaning that's not really well defined at all. *mask* of 77 shows all debug messages, which is what you usually need.

−e *filter*
: A qualifying expression which modifies which library calls to trace. The format of the filter expression is described in the section **FILTER EXPRESSIONS**. If more than one −e option appears on the command line, the library calls that match any of them are traced. If no −e is given, **@MAIN** is assumed as a default.

−f
: Trace child processes as they are created by currently traced processes as a result of the fork(2) or clone(2) system calls. The new process is attached immediately.

−F *filename*
: Load an alternate config file. Normally, /etc/ltrace.conf and ~/.ltrace.conf will be read (the latter only if it exists). Use this option to load the given file or files instead of those two default files. See ltrace.conf(5) for details on the syntax of ltrace configuration files.

−h, −−help
: Show a summary of the options to ltrace and exit.

−i Print the instruction pointer at the time of the library call.

−l, −−library *library_pattern*
Display only calls to functions implemented by libraries that match *library_pattern.* Multiple library patters can be specified with several instances of this option. Syntax of library_pattern is described in section **FILTER EXPRESSIONS**.

Note that while this option selects calls that might be directed to the selected libraries, there's no actual guarantee that the call won't be directed elsewhere due to e.g. LD_PRELOAD or simply dependency ordering. If you want to make sure that symbols in given library are actually called, use **-x** @*library_pattern* instead.

−L When no -e option is given, don't assume the default action of **@MAIN**.

−n, −−indent *nr*
Indent trace output by *nr* spaces for each level of call nesting. Using this option makes the program flow visualization easy to follow. This indents uselessly also functions that never return, such as service functions for throwing exceptions in the C++ runtime.

−o, −−output *filename*
Write the trace output to the file *filename* rather than to stderr.

−p *pid* Attach to the process with the process ID *pid* and begin tracing. This option can be used together with passing a command to execute. It is possible to attach to several processes by passing more than one option −p.

−r Print a relative timestamp with each line of the trace. This records the time difference between the beginning of successive lines.

−s *strsize*
Specify the maximum string size to print (the default is 32).

−S Display system calls as well as library calls

−t Prefix each line of the trace with the time of day.

−tt If given twice, the time printed will include the microseconds.

−ttt If given thrice, the time printed will include the microseconds and the leading portion will be printed as the number of seconds since the epoch.

−T Show the time spent inside each call. This records the time difference between the beginning and the end of each call.

−u *username*
Run command with the userid, groupid and supplementary groups of *username*. This option is only useful when running as root and enables the correct execution of setuid and/or setgid binaries.

−w, --where *nr*
Show backtrace of *nr* stack frames for each traced function. This option enabled only if libunwind support was enabled at compile time.

−x *filter*
A qualifying expression which modifies which symbol table entry points to trace. The format of the filter expression is described in the section **FILTER EXPRESSIONS**. If more than one −x option appears on the command line, the symbols that match any of them are traced. No entry points are traced if no −x is given.

−V, −−version
Show the version number of ltrace and exit.

FILTER EXPRESSIONS

Filter expression is a chain of glob- or regexp-based rules that are used to pick symbols for tracing from libraries that the process uses. Most of it is intuitive, so as an example, the following would trace calls to

malloc and free, except those done by libc:

-e malloc+free-@libc.so*

This reads: trace malloc and free, but don't trace anything that comes from libc. Semi-formally, the syntax of the above example looks approximately like this:

{[+-][*symbol_pattern*][@*library_pattern*]}

Symbol_pattern is used to match symbol names, *library_pattern* to match library SONAMEs. Both are implicitly globs, but can be regular expressions as well (see below). The glob syntax supports meta-characters ***** and **?** and character classes, similarly to what basic bash globs support. **^** and **$** are recognized to mean, respectively, start and end of given name.

Both *symbol_pattern* and *library_pattern* have to match the whole name. If you want to match only part of the name, surround it with one or two *'s as appropriate. The exception is if the pattern is not mentioned at all, in which case it's as if the corresponding pattern were *****. (So **malloc** is really **malloc@*** and **@libc.*** is really ***@libc.***.)

In libraries that don't have an explicit SONAME, basename is taken for SONAME. That holds for main binary as well: **/bin/echo** has an implicit SONAME of **echo**. In addition to that, special library pattern **MAIN** always matches symbols in the main binary and never a library with actual SONAME **MAIN** (use e.g. **^MAIN** or **[M]AIN** for that).

If the symbol or library pattern is surrounded in slashes (/like this/), then it is considered a regular expression instead. As a shorthand, instead of writing **/x/@/y/**, you can write **/x@y/**.

If the library pattern starts with a slash, it is not a SONAME expression, but a path expression, and is matched against the library path name.

The first rule may lack a sign, in which case **+** is assumed. If, on the other hand, the first rule has a **-** sign, it is as if there was another rule **@** in front of it, which has the effect of tracing complement of given rule.

The above rules are used to construct the set of traced symbols. Each candidate symbol is passed through the chain of above rules. Initially, the symbol is *unmarked*. If it matches a **+** rule, it becomes *marked*, if it matches a **-** rule, it becomes *unmarked* again. If, after applying all rules, the symbol is *marked*, it will be traced.

BUGS

It has most of the bugs stated in **strace(1)**.

It only works on Linux and in a small subset of architectures.

If you would like to report a bug, send a message to the mailing list (ltrace-devel@lists.alioth.debian.org), or use the **reportbug(1)** program if you are under the Debian GNU/Linux distribution.

FILES

/etc/ltrace.conf
: System configuration file

~/.ltrace.conf
: Personal config file, overrides */etc/ltrace.conf*

AUTHOR

Juan Cespedes <cespedes@debian.org>
Petr Machata <pmachata@redhat.com>

SEE ALSO

ltrace.conf(5), strace(1), **ptrace(2)**