NAME

ldd − print shared object dependencies

SYNOPSIS

ldd [*option*]... *file*...

DESCRIPTION

ldd prints the shared objects (shared libraries) required by each program or shared object specified on the command line.

Security

In the usual case, **ldd** invokes the standard dynamic linker (see **ld.so**(8)) with the **LD_TRACE_LOADED_OBJECTS** environment variable set to 1, which causes the linker to display the library dependencies. Be aware, however, that in some circumstances, some versions of **ldd** may attempt to obtain the dependency information by directly executing the program. Thus, you should *never* employ **ldd** on an untrusted executable, since this may result in the execution of arbitrary code. A safer alternative when dealing with untrusted executables is:

```
$ objdump −p /path/to/program | grep NEEDED
```

OPTIONS

−−version
Print the version number of **ldd**.

−v, **−−verbose**
Print all information, including, for example, symbol versioning information.

−u, **−−unused**
Print unused direct dependencies. (Since glibc 2.3.4.)

−d, **−−data−relocs**
Perform relocations and report any missing objects (ELF only).

−r, **−−function−relocs**
Perform relocations for both data objects and functions, and report any missing objects or functions (ELF only).

−−help Usage information.

BUGS

ldd does not work on a.out shared libraries.

ldd does not work with some extremely old a.out programs which were built before **ldd** support was added to the compiler releases. If you use **ldd** on one of these programs, the program will attempt to run with *argc* = 0 and the results will be unpredictable.

SEE ALSO

pldd(1), **sprof**(1), **ld.so**(8), **ldconfig**(8)

COLOPHON

This page is part of release 4.04 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at http://www.kernel.org/doc/man−pages/.