## NAME

luac - Lua compiler

## SYNOPSIS

**luac** [ *options* ] [ *filenames* ]

## DESCRIPTION

**luac** is the Lua compiler. It translates programs written in the Lua programming language into binary files that can be later loaded and executed.

The main advantages of precompiling chunks are: faster loading, protecting source code from accidental user changes, and off-line syntax checking.

Precompiling does not imply faster execution because in Lua chunks are always compiled into bytecodes before being executed. **luac** simply allows those bytecodes to be saved in a file for later execution.

Precompiled chunks are not necessarily smaller than the corresponding source. The main goal in precompiling is faster loading.

The binary files created by **luac** are portable only among architectures with the same word size and byte order.

**luac** produces a single output file containing the bytecodes for all source files given. By default, the output file is named **luac.out**, but you can change this with the **-o** option.

In the command line, you can mix text files containing Lua source and binary files containing precompiled chunks. This is useful because several precompiled chunks, even from different (but compatible) platforms, can be combined into a single precompiled chunk.

You can use **'-'** to indicate the standard input as a source file and **'--'** to signal the end of options (that is, all remaining arguments will be treated as files even if they start with **'-'**).

The internal format of the binary files produced by **luac** is likely to change when a new version of Lua is released. So, save the source files of all Lua programs that you precompile.

## OPTIONS

Options must be separate.

**-l** produce a listing of the compiled bytecode for Lua's virtual machine. Listing bytecodes is useful to learn about Lua's virtual machine. If no files are given, then **luac** loads **luac.out** and lists its contents.

**-o** *file* output to *file*, instead of the default **luac.out**. (You can use **'-'** for standard output, but not on platforms that open standard output in text mode.) The output file may be a source file because all files are loaded before the output file is written. Be careful not to overwrite precious files.

**-p** load files but do not generate any output file. Used mainly for syntax checking and for testing precompiled chunks: corrupted files will probably generate errors when loaded. Lua always performs a thorough integrity test on precompiled chunks. Bytecode that passes this test is completely safe, in the sense that it will not break the interpreter. However, there is no guarantee that such code does anything sensible. (None can be given, because the halting problem is unsolvable.) If no files are given, then **luac** loads **luac.out** and tests its contents. No messages are displayed if the file passes the integrity test.

**-s** strip debug information before writing the output file. This saves some space in very large chunks, but if errors occur when running a stripped chunk, then the error messages may not contain the full information they usually do. For instance, line numbers and names of local variables are lost.

**-v** show version information.

## FILES

**luac.out** default output file

## SEE ALSO

**lua**(1)

<http://www.lua.org/>

## DIAGNOSTICS

Error messages should be self explanatory.

## AUTHORS

L. H. de Figueiredo, R. Ierusalimschy and W. Celes