NAME

lua − Lua interpreter

SYNOPSIS

lua [*options*] [*script* [*args*]]

DESCRIPTION

lua is the stand-alone Lua interpreter. It loads and executes Lua programs, either in textual source form or in precompiled binary form. (Precompiled binaries are output by **luac**, the Lua compiler.) **lua** can be used as a batch interpreter and also interactively.

The given *options* (see below) are executed and then the Lua program in file *script* is loaded and executed. The given *args* are available to *script* as strings in a global table named **arg**. If these arguments contain spaces or other characters special to the shell, then they should be quoted (but note that the quotes will be removed by the shell). The arguments in **arg** start at 0, which contains the string '*script*'. The index of the last argument is stored in **arg.n**. The arguments given in the command line before *script*, including the name of the interpreter, are available in negative indices in **arg**.

At the very start, before even handling the command line, **lua** executes the contents of the environment variable **LUA_INIT**, if it is defined. If the value of **LUA_INIT** is of the form '*@filename*', then *filename* is executed. Otherwise, the string is assumed to be a Lua statement and is executed.

Options start with **'−'** and are described below. You can use **'−-'** to signal the end of options.

If no arguments are given, then **−v −i** is assumed when the standard input is a terminal; otherwise, **−** is assumed.

In interactive mode, **lua** prompts the user, reads lines from the standard input, and executes them as they are read. If a line does not contain a complete statement, then a secondary prompt is displayed and lines are read until a complete statement is formed or a syntax error is found. So, one way to interrupt the reading of an incomplete statement is to force a syntax error: adding a **';'** in the middle of a statement is a sure way of forcing a syntax error (except inside multiline strings and comments; these must be closed explicitly). If a line starts with **'='**, then **lua** displays the values of all the expressions in the remainder of the line. The expressions must be separated by commas. The primary prompt is the value of the global variable **_PROMPT**, if this value is a string; otherwise, the default prompt is used. Similarly, the secondary prompt is the value of the global variable **_PROMPT2**. So, to change the prompts, set the corresponding variable to a string of your choice. You can do that after calling the interpreter or on the command line (but in this case you have to be careful with quotes if the prompt string contains a space; otherwise you may confuse the shell.) The default prompts are "> " and ">> ".

OPTIONS

−	load and execute the standard input as a file, that is, not interactively, even when the standard input is a terminal.
−e *stat*	execute statement *stat*. You need to quote *stat* if it contains spaces, quotes, or other characters special to the shell.
−i	enter interactive mode after *script* is executed.
−l *name*	call **require('***name***')** before executing *script*. Typically used to load libraries.
−v	show version information.

SEE ALSO

luac(1)
http://www.lua.org/

DIAGNOSTICS

Error messages should be self explanatory.

AUTHORS

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