

**NAME**

`clisp` — Common Lisp language interpreter and compiler

**SYNOPSIS**

**clisp** [ **-h** ] [ **-m** *memsize* ] [ **-s** *stacksize* ] [ **-M** *memfile* ] [ **-L** *language* ] [ **-q** ] [ **-I** ] [ **-i** *initfile* ... ] [ **-c** [ **-l** ] *lispfile* ... ] [ **-x** *expression* ]

**DESCRIPTION**

Invokes the common lisp interpreter and compiler. Invoked without arguments, executes a read-eval-print loop, in which expressions are in turn read from standard input, evaluated by the lisp interpreter, and their results output to standard output. Invoked with **-c**, the specified lisp files are compiled to a bytecode that can be executed more efficiently.

**OPTIONS**

- h**        Displays a help message on how to use **clisp**.
- m** *memsize*  
Sets the amount of memory **clisp** tries to grab on startup. The amount may be given as *nnnnnnnn* (measured in bytes), *nnnnK* or *nnnnKB* (measured in kilobytes) or *nM* or *nMB* (measured in megabytes). Default is 2 megabytes. The argument is constrained between 100 KB and 16 MB. -- This version of *clisp* allocates memory dynamically. *memsize* is essentially ignored.
- s** *stacksize*  
Sets the size of the stack **clisp** allocates for itself. The syntax is the same as for *memsize*. Default is one eighth of *memsize*. The argument is constrained between 40 KB and 8 MB.
- M** *memfile*  
Specifies the initial memory image. This must be a memory dump produced by the *saveinitmem* function.
- L** *language*  
Specifies the language **clisp** uses to communicate with the user. This may be **english**, **deutsch**, **francais**.
- q**        Quiet: **clisp** displays no banner at startup and no good-bye message when quitting.
- I**        ILISP friendly: **clisp** interacts in a way that ILISP (a popular Emacs LISP interface) can deal with. Currently the only effect of this is that unnecessary prompts are not suppressed.
- i** *initfile* ...  
Specifies initialization files to be *loaded* at startup. These should be lisp files (source or compiled).
- c** *lispfile* ...  
Compiles the specified lispfiles to bytecode. The compiled files can then be *loaded* instead of the sources to gain efficiency.
- l**        A bytecode listing of the files being compiled will be produced. Useful only for debugging purposes.
- x** *expressions*  
Executes a series of arbitrary expressions instead of a read-eval-print loop. The values of the expressions will be output to standard output. Due to the argument processing done by the shell, the *expressions* must be enclosed in single quotes, and double quotes and backslashes must be preceded by backslashes.

**REFERENCE**

The language implemented conforms to  
 Guy L. Steele Jr.: Common Lisp - The Language.  
 Digital Press. 1st edition 1984, 465 pages.

("CLtL1" for short)  
 and to the older parts of  
 Guy L. Steele Jr.: Common Lisp - The Language.  
 Digital Press. 2nd edition 1990, 1032 pages.  
 ("CLtL2" for short)

**USE**

**help** to get some on-line help.  
**(apropos name)**  
 lists the symbols relating to *name*.  
**(exit)** or **(quit)** or **(bye)**  
 to quit **clisp**.  
 EOF (Ctrl-Z)  
 to leave the current read-eval-print loop.  
 arrow keys  
 for editing and viewing the input history.  
 Tab key  
 to complete the symbol's name you are just typing.

**FILES**

*lisp.exe*  
 main executable  
*lispinit.mem*  
 initial memory image  
*config.lsp*  
 site-dependent configuration  
*\*.lsp* lisp source  
*\*.fas* lisp code, compiled by **clisp**  
*\*.lib* lisp source library information, generated and used by the **clisp** compiler

**ENVIRONMENT****CLISP\_LANGUAGE**

specifies the language **clisp** uses to communicate with the user. The value may be **english**, **deutsch**, **francais** and defaults to **english**. The **-L** option can be used to override this environment variable.

**COMSPEC** (DOS, OS/2 implementations only)

is used to find the command interpreter called by the function *shell*.

**SEE ALSO**

*cmulisp*(1), *emacs*(1).

**BUGS**

The function *inspect* is not implemented.  
 Only very few extensions from CLtL2 are supported.  
 No on-line documentation beyond *apropos* and *describe* is available.

**PROJECTS**

Writing on-line documentation.  
 Building a foreign function interface (ability to call C code directly).  
 Write *inspect*.  
 Enhance the compiler such that it can inline local functions.

Specify a portable set of window and graphics operations.

**AUTHORS**

Bruno Haible <haible@ma2s2.mathematik.uni-karlsruhe.de> and Michael Stoll.