NAME

lisp - CMU Common Lisp programming environment

SYNOPSIS

lisp [input-files] [switch-list]

DESCRIPTION

lisp Starts up CMU Common Lisp. If switch-list is empty, then Lisp will enter a read-eval-print loop using stdin, stdout and stderr. The optional input-files provide operands for some switches. These switches are defined:

-core filename
Specifies the suspended Lisp image (or 'core file') to start up. The default is 'lib/lisp.core'.

-edit Causes Lisp to enter the Hemlock editor. A file to edit may be specified by placing the name of the file between the program name (usually 'lisp') and the first switch.

-eval expression
Evaluates the specified Lisp expression during the start up sequence. The value of the form will not be printed unless it is wrapped in a form that does output.

-init filename
Specifies the name of a file containing user customizations that is to be loaded each time Lisp starts up (default ~/init or ~/.cmucl-init.) The loader loads any existing compiled binary, or the ".lisp" source if none.

-hinit filename
Similar to -init, but specifies the name of the Hemlock init file (default '~/hemlock-init' or ~/.hemlock-init), which is loaded only when Hemlock is started.

-noinit Suppresses loading of the init file, and also prevents -edit from loading the Hemlock init file.

-load filename
Loads the specified file into Lisp before entering Lisp's read-eval-print loop.

-slave editor-name
Specifies that Lisp should start up as a slave Lisp and try to connect to an editor Lisp. The name of the editor to connect to must be

specified. To find the editor's name, use the Hemlock 'Accept Slave Connections' command.

editor-name is of the form 'machine-name:socket', where machine-name is the internet host name for the machine and socket is the decimal number of the socket to connect to.

ENVIRONMENT

CMUCLLIB This variable points to the 'lib/' directory holding 'lisp.core' and other files used by Lisp at run-time. The default is '/usr/local/lib/cmucl/lib'.

CMUCL_EMPTYFILE
[SunOS only] If 'df /tmp' shows 'swap' as the filesystem for the '/tmp' directory, then you have a "tmpfs" filesystem. In this case, you must setenv CMUCL_EMPTYFILE to point into a pathname on a non-TMPFS filesystem that can be used instead of '/tmp/empty'.

XKEYSYMDB In order to use Motif (and the graphical debugger) with X servers from non-OSF vendors (like Sun) you may need to set the environment variable XKEYSYMDB to point to the file 'lib/XKeysymDB'. Otherwise, you will get many error messages every time a new connection is opened to the CMU CL motifd. This file is read by the X11R5 Xt in order to augment the keysym database with certain OSF vendor keysyms that Motif wants to use.

FILES

The following pathnames are specified relative to the directory where CMU CL is installed.

doc/* Various postscript and text documentation files.

bin/lisp The lisp startup program. This directory should be in PATH.

lib/lisp.core
The suspended Lisp image.

lib/site-init.lisp
Site specific initialization (see README file.)

lib/hemlock11.*,lib/mh-scan,lib/spell-dictionary.bin
Hemlock files.

lib/fonts/
X11 fonts for Hemlock.

lib/XKeysymDB
Database of X Keysym names for Motif.

lib/load-foreign.csh
Script used by LOAD-FOREIGN to run "ld".

~/init.lisp,~/.cmucl-init.lisp
User customization files loaded at lisp startup; either name is acceptable. Init files can be compiled.

~/hemlock-init.lisp,~/.hemlock-init.lisp
Hemlock initialization file, loaded when Hemlock starts.

SEE ALSO

cmucl(1), README
The ``CMU Common Lisp User's Manual'',
the ``Hemlock User's Manual'', and
the ``Hemlock Command Implementor's Manual''

BUGS

Bug reports should be sent to cmucl-bugs@cs.cmu.edu. Please consult your local CMU CL maintainer or Common Lisp expert to verify that the problem really is a bug before sending to this list.

Known problems with this version:

-- Detection of stack overflow is not very graceful. You get many "map failure" errors on stderr.

-- If file descriptors are used up, then Lisp will die.

-- Several proposed ANSI Common Lisp (CLtL II) features are not implemented: Any CLOS features not implemented by PCL, and features added since the first ANSI draft.

-- The interpreter's pre-processing freezes in the macro definitions in effect at the time an interpreted function is defined.