NAME

dwarf_loclist_from_expr, **dwarf_loclist_from_expr_a**, **dwarf_loclist_from_expr_b** - translate DWARF location expression bytes

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_loclist_from_expr(*Dwarf_Debug dbg, Dwarf_Ptr bytes_in, Dwarf_Unsigned bytes_len, Dwarf_Locdesc **llbuf, Dwarf_Signed *listlen, Dwarf_Error *err*);

int
dwarf_loclist_from_expr_a(*Dwarf_Debug dbg, Dwarf_Ptr bytes_in, Dwarf_Unsigned bytes_len, Dwarf_Half addr_size, Dwarf_Locdesc **llbuf, Dwarf_Signed *listlen, Dwarf_Error *err*);

int
dwarf_loclist_from_expr_b(*Dwarf_Debug dbg, Dwarf_Ptr bytes_in, Dwarf_Unsigned bytes_len, Dwarf_Half addr_size, Dwarf_Half offset_size, Dwarf_Small version, Dwarf_Locdesc **llbuf, Dwarf_Signed *listlen, Dwarf_Error *error*);

DESCRIPTION

Function **dwarf_loclist_from_expr**() translates DWARF location expression bytes into a *Dwarf_Locdesc* descriptor. The size for address related data is taken to be the default address size for the object being read.

Argument *dbg* should reference a DWARF debug context allocated using dwarf_init(3).

Argument *bytes_in* should point to an array of DWARF location expression bytes.

Argument *bytes_len* should specify the number of the location expression bytes to be translated.

Argument *llbuf* should point to a location which will be set to a pointer to a returned *Dwarf_Locdesc* descriptor.

Argument *listlen* should point to a location which will hold the number of the *Dwarf_Locdesc* descriptors returned. In this case it is always set to 1.

If argument *err* is not NULL, it will be used to store error information in case of an error.

Function **dwarf_loclist_from_expr_a**() is identical to function **dwarf_loclist_from_expr**(), except that it requires one additional argument *addr_size*, which specifies the address size to use when translating the location expression bytes.

Function **dwarf_loclist_from_expr_b**() is identical to function **dwarf_loclist_from_expr_a**() except that it requires two additional arguments for translating the location expression bytes. Argument *offset_size* specifies the offset size, and argument *version* specifies the DWARF version. These values are required to correctly translate the DW_OP_GNU_implicit_pointer opcode.

Memory Management

The memory area used for the descriptor returned in argument *llbuf* is allocated by DWARF Access Library (libdwarf, -ldwarf). When the descriptor is no longer needed, application code should use function dwarf_dealloc(3) to free the memory area in two steps:

1. First, the array of *Dwarf_Loc* descriptors pointed to by the *ld_s* field of the *Dwarf_Locdesc* descriptor should be deallocated using the allocation type DW_DLA_LOC_BLOCK.
2. Next, the application should free the *llbuf* pointer using the allocation type DW_DLA_LOCDESC.

RETURN VALUES

On success, these functions returns DW_DLV_OK. In case of an error, they return DW_DLV_ERROR and set the argument *err*.

ERRORS

These functions may fail with the following errors:

[DW_DLE_ARGUMENT]	One of the arguments *dbg*, *bytes_in*, *llbuf* or *listlen* was NULL.
[DW_DLE_ARGUMENT]	Argument *bytes_len* was 0.
[DW_DLE_ARGUMENT]	The value of argument *addr_size* was invalid.
[DW_DLE_LOC_EXPR_BAD]	An unknown or invalid operation was found in the location expression bytes provided in argument *bytes_in*.
[DW_DLE_MEMORY]	An out of memory condition was encountered during the execution of this function.

SEE ALSO

dwarf(3), dwarf_dealloc(3), dwarf_get_fde_info_for_all_regs3(3), dwarf_get_fde_info_for_cfa_reg3(3),

dwarf_get_fde_info_for_reg3(3), dwarf_get_loclist_entry(3), dwarf_loclist_n(3)