NAME

dwarf_next_cu_header, **dwarf_next_cu_header_b**, **dwarf_next_cu_header_c** **-** step through compilation units in a DWARF debug context

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_next_cu_header(*Dwarf_Debug dbg*, *Dwarf_Unsigned *cu_length*, *Dwarf_Half *cu_version*, *Dwarf_Off *cu_abbrev_offset*, *Dwarf_Half *cu_pointer_size*, *Dwarf_Unsigned *cu_next_offset*, *Dwarf_Error *err*);

int
dwarf_next_cu_header_b(*Dwarf_Debug dbg*, *Dwarf_Unsigned *cu_length*, *Dwarf_Half *cu_version*, *Dwarf_Off *cu_abbrev_offset*, *Dwarf_Half *cu_pointer_size*, *Dwarf_Half *cu_offset_size*, *Dwarf_Half *cu_extension_size*, *Dwarf_Unsigned *cu_next_offset*, *Dwarf_Error *err*);

int
dwarf_next_cu_header_c(*Dwarf_Debug dbg*, *Dwarf_Bool is_info*, *Dwarf_Unsigned *cu_length*, *Dwarf_Half *cu_version*, *Dwarf_Off *cu_abbrev_offset*, *Dwarf_Half *cu_pointer_size*, *Dwarf_Half *cu_offset_size*, *Dwarf_Half *cu_extension_size*, *Dwarf_Sig8 *type_signature*, *Dwarf_Unsigned *type_offset*, *Dwarf_Unsigned *cu_next_offset*, *Dwarf_Error *err*);

DESCRIPTION

These functions are used to step through compilation or type units associated with a DWARF debug context, optionally returning information about the unit.

Function **dwarf_next_cu_header_c**() is the API recommended for new application code. Function **dwarf_next_cu_header**() and **dwarf_next_cu_header_b**() can only operate on compilation units associated with the ".debug_info" section. They are less general than function **dwarf_next_cu_header_c**(), and are deprecated for use by new application code.

Argument *dbg* should reference a DWARF debug context allocated using dwarf_init(3). If argument *is_info* is set to 1, the function returns information for compilation units found in the ".debug_info" section. If argument *is_info* is set to 0, the function returns information for type units found in the ".debug_types" sections. Argument *cu_length* should point to a location that will be set to the length of the compilation or type unit. Argument *cu_version* should point to a location that will be set to the version number for the compilation or type unit. Argument *cu_abbrev_offset* should point to a location that will be set to the starting offset (in the ".debug_abbrev" section) of the set of debugging information entry abbreviations associated with this compilation or type unit. Argument *cu_pointer_size* should point to a location that will be set to the size in bytes of an address for the machine architecture of the underlying object being debugged. Argument *cu_offset_size* should point to a location that will be set to the size in bytes for a DWARF offset in the compilation or type unit. Argument *cu_extension_size* is only needed for processing MIPS/IRIX objects that use a non-standard DWARF format. It should point to a location that will be set to 4 for normal objects and to 0 for non-standard ones. Argument *type_signature* and *type_offset* is only needed for processing type units. Argument *type_signature* should point to a location that will be set to the 64-bit unique signature of the type described in the type unit. Argument *type_offset* should point to a location that will be set to the offset of the debugging information entry that describes the type. Argument *cu_next_offset* should point to a location that will be set to the offset of the next compilation unit header in the ".debug_info" section, or the offset of the next type unit header in the ".debug_types" section. Argument *err* should point to a location that will hold an error descriptor in case of an error.

Function **dwarf_next_cu_header_b**() is identical to function **dwarf_next_cu_header_c**() except that it does not provide arguments *is_info*, *type_signature* and *type_offset*.

Function **dwarf_next_cu_header**() is identical to function **dwarf_next_cu_header_b**() except that it does not provide arguments *cu_offset_size* and *cu_extension_size*.

A value of NULL may be used for any of the arguments *cu_length*, *cu_version*, *cu_abbrev_offset*, *cu_pointer_size*, *cu_offset_size*, *cu_extension_size*, *type_signature*, *type_offset*, *cu_next_offset* and *err* if the caller is not interested in the respective value.

Iterating Through Compilation Units in a Debug Context

The first call to function **dwarf_next_cu_header_c**() for a given debug context with argument *is_info* set to 1 will return information about the first compilation unit in the ".debug_info" section. Subsequent calls to the function will iterate through the remaining compilation units in the section. On stepping past the last compilation unit in the section, function **dwarf_next_cu_header_c**() returns DW_DLV_NO_ENTRY and resets its internal state. The next call to the function will restart from the first compilation unit in the section.

Iterating Through Type Units in a Debug Context

When a DWARF debug context is allocated using dwarf_init(3), an internal pointer associated with the context will point to the first ".debug_types" section found in the debug object. The first call to function **dwarf_next_cu_header_c**() for the debug context with argument *is_info* set to 0 will return information about the first type unit in that ".debug_types" section. Subsequent calls to the function will iterate through the remaining type units in the section. On stepping past the last type unit in the debug context, function **dwarf_next_cu_header_c**() returns DW_DLV_NO_ENTRY and resets its internal state. The next call to the function will restart from the first type unit in the ".debug_types" section.

If the debug object contains multiple ".debug_types" sections, the function **dwarf_next_types_section**() can be called to move the internal pointer to the next ".debug_types" section. As a result, subsequent calls of the function **dwarf_next_cu_header_c**() will operate on the new ".debug_types" section. Function **dwarf_next_types_section**() returns DW_DLV_NO_ENTRY when there are no more ".debug_types" sections left in the debug object.

RETURN VALUES

On success, these functions return DW_DLV_OK. In case of an error, they return DW_DLV_ERROR and set argument *err*. When there are no more compilation units left to traverse, they return DW_DLV_NO_ENTRY.

ERRORS

These functions can fail with the following error:

[DW_DLE_ARGUMENT] Argument *dbg* was NULL.

SEE ALSO

dwarf(3), dwarf_get_cu_die_offset_given_cu_header_offset(3), dwarf_init(3), dwarf_next_types_section(3), dwarf_siblingof(3)