NAME

dwarf_get_relocation_info **-** retrieve generated relocation arrays

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_relocation_info(*Dwarf_P_Debug dbg*, *Dwarf_Signed *elf_section_index*,
*Dwarf_Signed *elf_section_link*, *Dwarf_Unsigned *reloc_entry_count*,
*Dwarf_Relocation_Data *reloc_buf*, *Dwarf_Error *err*);

DESCRIPTION

The function **dwarf_get_relocation_info**() is used to retrieve the relocation arrays generated by a prior call to dwarf_transform_to_disk_form(3).

Each call to this function retrieves the next available relocation array. Application code should call this function repeatly to retrieve all the relocation arrays. The total number of generated relocation arrays retrievable by this function may be obtained by calling function dwarf_get_relocation_info_count(3).

Argument *dbg* should reference a DWARF producer instance allocated using dwarf_producer_init(3) in sequence. or dwarf_producer_init_b(3). The DW_DLC_SYMBOLIC_RELOCATIONS flag should have been set on the DWARF producer instance.

Argument *elf_section_index* should point to a location which will be set to the ELF section index of the relocation section to which the retrieved relocation array belongs.

Argument *elf_section_link* should point to a location which will be set to the section index of the ELF section to which the retrieved relocation array applies.

Argument *reloc_entry_count* should point to a location which will be set to the total number of relocation entries contained in the relocation array.

Argument *reloc_buf* should point to a location which will be set to a pointer to the retrieved array of relocation entries.

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

The retrieved relocation entries are described using structure *Dwarf_Relocation_Data_s*, defined in the header file <*libdwarf.h*>:

```
typedef struct Dwarf_Relocation_Data_s {
        unsigned char drd_type;
        unsigned char drd_length;
        Dwarf_Unsigned drd_offset;
        Dwarf_Unsigned drd_symbol_index;
} *Dwarf_Relocation_Data;
```

Struct *Dwarf_Relocation_Data_s* consists of following fields:

drd_type The type code of the relocation entry. The *Dwarf_Rel_Type* enumeration defined in the header file <*libdwarf.h*> specifies legal values for this field.

drd_length The size in bytes of the field to be relocated.

drd_offset The section-relative offset of the field to be relocated.

drd_symbol_index The symbol index associated with the relocation entry.

Memory Management

The memory area used for the relocation arrays is managed by the DWARF Access Library (libdwarf, -ldwarf). The function **dwarf_producer_finish**() may be used to release it, along with other resources associated with the producer instance.

RETURN VALUES

On success, function **dwarf_get_relocation_info**() returns DW_DLV_OK. It returns DW_DLV_NO_ENTRY if there were no more relocation arrays to retrieve, or if the flag DW_DLC_SYMBOLIC_RELOCATIONS was not set on the producer instance. In case of an error, function **dwarf_get_relocation_info**() returns DW_DLV_ERROR and sets the argument *err*.

EXAMPLES

To generate relocation entries and retrieve them, use:

```
Dwarf_P_Debug dbg;
Dwarf_Relocation_Data buf;
Dwarf_Signed count, index, link;
Dwarf_Unsigned reloc_cnt, entry_cnt;
Dwarf_Error de;
int version, i, j;

/*
 * Assume that dbg refers to a DWARF producer instance created
 * created with DW_DLC_SYMBOLIC_RELOCATIONS flag set and that
 * application code has added DWARF debugging information
 * to the producer instance.
 */
if ((count = dwarf_transform_to_disk_form(dbg, &de)) ==
    DW_DLV_NOCOUNT) {
        warnx("dwarf_transform_to_disk_form failed: %s",
            dwarf_errmsg(-1));
        return;
}

/* ... process generated section byte streams ... */
if (dwarf_get_relocation_info_count(dbg, &reloc_cnt, &version, &de) !=
    DW_DLV_OK) {
        warnx("dwarf_get_relocation_info_count failed: %s",
            dwarf_errmsg(-1));
        return;
}

for (i = 0; (Dwarf_Unsigned) i < reloc_cnt; i++) {
        if (dwarf_get_relocation_info(dbg, &index, &link, &entry_cnt,
            &buf, &de) != DW_DLV_OK) {
                warnx("dwarf_get_relocation_info failed: %s",
                    dwarf_errmsg(-1));
                continue;
        }
        for (j = 0; (Dwarf_Unsigned) j < entry_cnt; j++) {
                /* ...use each reloc data in buf[j]... */
        }
}

dwarf_producer_finish(dbg, &de);
```

ERRORS

Function **dwarf_get_relocation_info**() can fail with:

[DW_DLE_ARGUMENT] One of the arguments *dbg*, *elf_section_index*, *elf_section_link*, *reloc_entry_count* or *reloc_buf* was NULL.

[DW_DLE_NO_ENTRY] There were no more ELF relocation arrays to retrieve.

[DW_DLE_NO_ENTRY] The flag DW_DLC_SYMBOLIC_RELOCATIONS was not set on the producer instance.

[DW_DLE_NO_ENTRY] Function dwarf_transform_to_disk_form(3) was not called prior to calling function **dwarf_get_relocation_info**().

SEE ALSO

dwarf(3), dwarf_get_relocation_info_count(3), dwarf_producer_finish(3), dwarf_producer_init(3), dwarf_producer_init_b(3), dwarf_reset_section_bytes(3), dwarf_transform_to_disk_form(3)