

NAME

elf_getarsym - retrieve the symbol table of an archive

LIBRARY

ELF Access Library (libelf, -lelf)

SYNOPSIS

#include <libelf.h>

Elf_Arsym *

elf_getarsym(*Elf* *elf, *size_t* *ptr);

DESCRIPTION

The function **elf_getarsym**() retrieves the symbol table for an ar(1) archive, if one is available.

Argument *elf* should be a descriptor for an ar(1) archive opened using **elf_begin**() or **elf_memory**().

If the archive *elf* contains a symbol table with *n* entries, this function returns a pointer to an array of *n*+1 *Elf_Arsym* structures. An *Elf_Arsym* structure has the following elements:

char * *as_name*

This structure member is a pointer to a null-terminated symbol name.

off_t *as_off*

This structure member contains the byte offset from the beginning of the archive to the header for the archive member. This value is suitable for use with **elf_rand**(3).

unsigned long *as_hash*

This structure member contains a portable hash value for the symbol name, as computed by **elf_hash**(3).

The last entry of the returned array will have a NULL value for member *as_name*, a zero value for member *as_off* and an illegal value of ~0UL for *as_hash*.

If argument *ptr* is non-null, the **elf_getarsym**() function will store the number of table entries returned (including the sentinel entry at the end) into the location it points to.

RETURN VALUES

Function **elf_getarsym**() returns a pointer to an array of *Elf_Arsym* structures if successful, or a NULL pointer if an error was encountered.

If argument *ptr* is non-null and there was no error, the library will store the number of archive symbol entries returned into the location it points to. If argument *ptr* is non-null and an error was encountered,

the library will set the location pointed to by it to zero.

ERRORS

Function **elf_getarsym()** may fail with the following errors:

[ELF_E_ARGUMENT]

Argument *elf* was NULL.

[ELF_E_ARGUMENT]

Argument *elf* was not a descriptor for an ar(1) archive.

SEE ALSO

elf(3), elf_begin(3), elf_getarhdr(3), elf_hash(3), elf_memory(3), elf_next(3), elf_rand(3)