NAME

elftc_string_table_create, **elftc_string_table_destroy**, **elftc_string_table_from_section**, **elftc_string_table_image**, **elftc_string_table_insert**, **elftc_string_table_lookup**, **elftc_string_table_remove**, **elftc_string_table_to_string** - convenience routines for handling ELF string tables

SYNOPSIS

#include <libelftc.h>

*Elftc_String_Table **
elftc_string_table_create(*size_t sizehint*);

void
elftc_string_table_destroy(*Elftc_String_Table *table*);

*Elftc_String_Table **
elftc_string_table_from_section(*Elf_Scn *scn, size_t sizehint*);

*const char **
elftc_string_table_image(*Elftc_String_Table *table, size_t *size*);

size_t
elftc_string_table_insert(*Elftc_String_Table *table, const char *string*);

size_t
elftc_string_table_lookup(*Elftc_String_Table *table, const char *string*);

int
elftc_string_table_remove(*Elftc_String_Table *table, const char *string*);

*const char **
elftc_string_table_to_string(*Elftc_String_Table *table, size_t offset*);

DESCRIPTION

This manual page documents convenience routines for handling ELF string tables.

Function **elftc_string_table_create**() creates a new, empty string table. The argument *sizehint* provides a hint about the expected number of bytes of string data in the table. If the argument *sizehint* is zero, an implementation-defined default will be used instead.

Function **elftc_string_table_destroy**() destroys the previously allocated string table specified by argument *table*, and frees the internal resources allocated for it.

Function **elftc_string_table_from_section**() creates a new string table and initializes it based on the contents of the section specified by argument *scn*. This section must be of type SHT_STRTAB. The argument *sizehint* provides a hint about expected number of bytes of string data in the table. If the value of *sizehint* is zero, an implementation-default will be used instead.

Function **elftc_string_table_image**() returns a pointer to the ELF representation of the contents of the string table specified by argument *table*. If argument *size* is not NULL, the size of the ELF representation of the string table is stored in the location pointed to by argument *size*. The function **elftc_string_table_image**() will compact the string table if the table contains deleted strings. The string offsets returned by prior calls to **elftc_string_table_insert**() and **elftc_string_table_lookup**() should be treated as invalid after a call to this function.

Function **elftc_string_table_insert**() inserts the NUL-terminated string pointed to by argument *string* into the string table specified by argument *table*, and returns an offset value usable in ELF data structures. Multiple insertions of the same content will return the same offset. The offset returned will remain valid until the next call to **elftc_string_table_image**().

Function **elftc_string_table_lookup**() looks up the string referenced by argument *string* in the string table specified by argument *table*, and if found, returns the offset associated with the string. The returned offset will be valid until the next call to **elftc_string_table_image**().

Function **elftc_string_table_remove**() removes the string pointed by argument *string* from the string table referenced by argument *table*, if it is present in the string table.

Function **elftc_string_table_to_string**() returns a pointer to the NUL-terminated string residing at argument *offset* in the string table specified by argument *table*. The value of argument *offset* should be one returned by a prior call to **elftc_string_table_insert**() or **elftc_string_table_lookup**(). The returned pointer will remain valid until the next call to **elftc_string_table_insert**() or **elftc_string_table_image**().

Memory Management

The library "libelftc" library manages its own memory allocations. The application should not free the pointers returned by the string table functions.

IMPLEMENTATION NOTES

The current implementation is optimized for the case where strings are added to a string table, but rarely removed from it.

The functions **elftc_string_table_insert**(), **elftc_string_table_lookup**(), **elftc_string_table_remove**() and **elftc_string_table_to_string**() have O(1) asymptotic behavior. The function **elftc_string_table_image**() can have O(size) asymptotic behavior, where *size* denotes the size of the string table.

RETURN VALUES

Functions **elftc_string_table_create**() and **elftc_string_table_from_section**() return a valid pointer to an opaque structure of type *Elftc_String_Table* on success, or NULL in case of an error.

The function **elftc_string_table_image**() returns a pointer to an in-memory representation of an ELF string table on success, or NULL in case of an error.

Functions **elftc_string_table_insert**() and **elftc_string_table_lookup**() return a non-zero offset on success, or zero in case of an error.

Function **elftc_string_table_remove**() returns a positive value on success, or zero in case of an error.

Function **elftc_string_table_to_string**() returns a valid pointer on success, or NULL in case of an error.

SEE ALSO

dwarf(3), elf(3), elftc(3)