NAME

elf_flagarhdr, **elf_flagdata**, **elf_flagehdr**, **elf_flagelf**, **elf_flagphdr**, **elf_flagscn**, **elf_flagshdr** **-** manipulate flags associated with ELF data structures

LIBRARY

ELF Access Library (libelf, -lelf)

SYNOPSIS

#include <libelf.h>

unsigned int
elf_flagarhdr(*Elf_Arhdr *arhdr*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagdata(*Elf_Data *data*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagehdr(*Elf *elf*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagelf(*Elf *elf*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagphdr(*Elf *elf*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagscn(*Elf_Scn *scn*, *Elf_Cmd cmd*, *unsigned int flags*);

unsigned int
elf_flagshdr(*Elf_Scn *scn*, *Elf_Cmd cmd*, *unsigned int flags*);

DESCRIPTION

These functions are used to query, set or reset flags on data structures associated with an ELF file.

Arguments *arhdr*, *data*, *elf* and *scn* denote the data structures whose flags need to be changed. These values should have been returned by prior calls to functions in the elf(3) API set:

- Argument *arhdr* should have been returned by a prior call to elf_getarhdr(3).
- Argument *data* should have been returned by a prior call to one of elf_newdata(3), elf_getdata(3) or elf_rawdata(3).
- Argument *elf* should have been allocated by a prior call to one of elf_begin(3) or elf_memory(3).

- Argument *scn* should have been returned by a prior call to one of elf_getscn(3), elf_newscn(3) or elf_nextscn(3).

These values are allowed to be NULL to simplify error handling in application code.

Argument *cmd* may have the following values:

ELF_C_CLR
The argument *flags* specifies the flags to be cleared.

ELF_C_SET The argument *flags* specifies the flags to be set.

The argument *flags* is allowed to have the following flags set:

ELF_F_ARCHIVE This flag is only valid with the **elf_flagelf**() API. It informs the library that the application desires to create an ar(1) archive. Argument *elf* should have been opened for writing using the ELF_C_WRITE command to function **elf_begin**().

ELF_F_ARCHIVE_SYSV This flag is used in conjunction with the ELF_F_ARCHIVE flag to indicate that library should create archives that conform to System V layout rules. The default is to create BSD style archives.

ELF_F_DIRTY Mark the associated data structure as needing to be written back to the underlying file. A subsequent call to elf_update(3) will resynchronize the library's internal data structures.

ELF_F_LAYOUT This flag is only valid with the **elf_flagelf**() API. It informs the library that the application will take responsibility for the layout of the file and that the library is not to insert any padding in between sections.

Marking a given data structure as "dirty" affects all of its contained elements. Thus marking an ELF descriptor *elf* with **elf_flagelf**(*elf*, *ELF_C_SET*, *ELF_F_DIRTY*) means that the entire contents of the descriptor are "dirty".

Using a value of zero for argument *flags* will return the current set of flags for the data structure being queried.

RETURN VALUES

These functions return the updated flags if successful, or zero if an error is detected.

COMPATIBILITY

The **elf_flagarhdr**() function and the ELF_F_ARCHIVE and ELF_F_ARCHIVE_SYSV flags are an extension to the elf(3) API.

ERRORS

These functions may fail with the following errors:

[ELF_E_ARGUMENT]
An unsupported value was used for the *cmd* argument.

[ELF_E_ARGUMENT]
Argument *flags* had unsupported flags set.

[ELF_E_ARGUMENT]
The argument *elf* was not a descriptor for an ELF object.

[ELF_E_MODE] The ELF_F_ARCHIVE flag was used with an ELF descriptor that had not been opened for writing.

[ELF_E_SEQUENCE] Function **elf_flagehdr**() was called without an executable header being allocated.

[ELF_E_SEQUENCE] Function **elf_flagphdr**() was called without a program header being allocated.

SEE ALSO

elf(3), elf32_newehdr(3), elf32_newphdr(3), elf64_newehdr(3), elf64_newphdr(3), elf_newdata(3), elf_update(3), gelf(3), gelf_newehdr(3), gelf_newphdr(3), gelf_update_dyn(3), gelf_update_move(3), gelf_update_rel(3), gelf_update_rela(3), gelf_update_sym(3), gelf_update_syminfo(3)