

NAME

`elf_next` – advance an ELF descriptor to the next archive member

SYNOPSIS

```
#include <libelf.h>
```

```
Elf_Cmd elf_next(Elf *elf);
```

DESCRIPTION

Advance an ELF descriptor associated with an archive file to the next available archive member.

ELF descriptors initialized from an archive file can be used to retrieve ELF descriptors for archive members one at a time using **elf_begin(3)**. **elf_next(3)** updates the archive descriptor so that **elf_begin(3)** returns the ELF descriptor of the next member of the archive. See the **EXAMPLES** section below.

RETURN VALUE

If *elf* refers to an archive member, update the state of the parent archive ELF descriptor associated with *elf* so that the next archive member can be retrieved with **elf_begin(3)**. Return the **Elf_Cmd** that was used with **elf_begin(3)** to initialize *elf*.

If *elf* was not initialized from an archive file or there are no more archive members, **elf_next(3)** returns **ELF_C_NULL**.

EXAMPLES

```
/* Open the archive. */
fd = open (archive_name, O_RDONLY);
if (fd == -1)
{
    printf ("cannot open archive file '%s'", fname);
    exit (1);
}

/* Set the ELF version. */
elf_version (EV_CURRENT);

/* Create an ELF descriptor for the archive. */
cmd = ELF_C_READ;
elf = elf_begin (fd, cmd, NULL);
if (elf == NULL)
{
    printf ("cannot create ELF descriptor: %s\n", elf_errmsg (-1));
    exit (1);
}

/* Verify this is a descriptor for an archive. */
if (elf_kind (elf) != ELF_K_AR)
{
    printf ("'%s' is not an archive\n", fname);
    exit (1);
}

/* Get the members of the archive one after the other. */
while ((subelf = elf_begin (fd, cmd, elf)) != NULL)
```

```
{
    /* Process subelf here */
    [...]

    /* elf_next updates elf, the parent archive, so that the next call
       to elf_begin returns the next archive member. */
    cmd = elf_next (subelf);
    if (elf_end (subelf) != 0)
    {
        printf ("error while freeing sub-ELF descriptor: %s\n",
                elf_errmsg (-1));
        exit (1);
    }
}

elf_end (elf);
close (fd);
```

SEE ALSO

elf_begin(3), elf_rand(3), libelf(3), elf(5)

ATTRIBUTES

Interface	Attribute	Value
elf_next()	Thread safety	MT-Safe

REPORTING BUGS

Report bugs to <elfutils-devel@sourceware.org> or <https://sourceware.org/bugzilla/>.