**NAME**

elf_getscn, elf_ndxscn, elf_newscn, elf_nextscn – get section information

**SYNOPSIS**

**cc** [ *flag* ... ] *file* ... **–lelf** [ *library* ... ]

**#include <libelf.h>**

**Elf_Scn *elf_getscn(Elf** **elf*, **size_t** *index***);**

**size_t elf_ndxscn(Elf_Scn** **scn***);**

**Elf_Scn *elf_newscn(Elf** **elf***);**

**Elf_Scn *elf_nextscn(Elf** **elf*, **Elf_Scn** **scn***);**

**MT-LEVEL**

Unsafe

**DESCRIPTION**

These functions provide indexed and sequential access to the sections associated with the ELF descriptor *elf*. If the program is building a new file, it is responsible for creating the file's ELF header before creating sections; see **elf32_getehdr**(3E).

**elf_getscn( )** returns a section descriptor, given an *index* into the file's section header table. Note the first ''real'' section has index 1. Although a program can get a section descriptor for the section whose *index* is 0 (**SHN_UNDEF**, the undefined section), the section has no data and the section header is ''empty'' (though present). If the specified section does not exist, an error occurs, or *elf* is null, **elf_getscn( )** returns a null pointer.

**elf_newscn( )** creates a new section and appends it to the list for *elf*. Because the **SHN_UNDEF** section is required and not ''interesting'' to applications, the library creates it automatically. Thus the first call to **elf_newscn( )** for an ELF descriptor with no existing sections returns a descriptor for section 1. If an error occurs or *elf* is null, **elf_newscn( )** returns a null pointer.

After creating a new section descriptor, the program can use **elf32_getshdr( )** to retrieve the newly created, ''clean'' section header. The new section descriptor will have no associated data (see **elf_getdata**(3E)). When creating a new section in this way, the library updates the **e_shnum** member of the ELF header and sets the **ELF_F_DIRTY** bit for the section (see **elf_flagdata**(3E)). If the program is building a new file, it is responsible for creating the file's ELF header (see **elf32_getehdr**(3E)) before creating new sections.

**elf_nextscn( )** takes an existing section descriptor, *scn*, and returns a section descriptor for the next higher section. One may use a null *scn* to obtain a section descriptor for the section whose index is 1 (skipping the section whose index is **SHN_UNDEF**). If no further sections are present or an error occurs, **elf_nextscn( )** returns a null pointer.

**elf_ndxscn( )** takes an existing section descriptor, *scn*, and returns its section table index. If *scn* is null or an error occurs, **elf_ndxscn( )** returns **SHN_UNDEF**.

**EXAMPLES**

An example of sequential access appears below. Each pass through the loop processes the next section in the file; the loop terminates when all sections have been processed.

```
scn = 0;
while ((scn = elf_nextscn(elf, scn)) != 0)
{
        /* process section */
}
```

**SEE ALSO**

**elf**(3E), **elf32_getehdr**(3E), **elf32_getshdr**(3E), **elf_begin**(3E), **elf_flagdata**(3E), **elf_getdata**(3E)