# ELF1 1B Section Groups

Young W. Lim

2022-04-26 Tue

## Outline

## Based on

"Study of ELF loading and relocs", 1999
http://netwinder.osuosl.org/users/p/patb/public_html/elf_relocs.html

## Compling 32-bit program on 64-bit gcc

- `gcc -v`
- `gcc -m32 t.c`
- `sudo apt-get install gcc-multilib`
- `sudo apt-get install g++-multilib`
- `gcc-multilib`
- `g++-multilib`
- `gcc -m32`
- `objdump -m i386`

## (1) ELF header and program headers

- the ELF file has an header that describes
  the overall layout of the file.
- the ELF header actually points to
  another group of headers called the program headers
    - these headers describe to the operating system
      anything that might be required for it
      to load the binary into memory and execute it.
    - segments are described by program headers,
      but so are some other things required
      to get the executable running.

https://www.bottomupcs.com/elf.xhtml

## (2) ELF32_Ehdr

### ELF File Header

```
typedef struct {
        unsigned char e_ident[EI_NIDENT];
        Elf32_Half    e_type;
        Elf32_Half    e_machine;
        Elf32_Word    e_version;
        Elf32_Addr    e_entry;
        Elf32_Off     e_phoff;
        Elf32_Off     e_shoff;  ........ for section header table
        Elf32_Word    e_flags;
        Elf32_Half    e_ehsize;
        Elf32_Half    e_phentsize;
        Elf32_Half    e_phnum;
        Elf32_Half    e_shentsize; ..... for section header table
        Elf32_Half    e_shnum;  ........ for section header table
        Elf32_Half    e_shstrndx; ...... section header table index for
} Elf32_Ehdr;                            the section name string table
```

https://www.bottomupcs.com/elf.xhtml

## (3) locating section headers

- in the ELF (File) header structure type

| | |
|---|---|
| e_shoff | the offset in the file where the section header table starts |
| e_shentsize | the size of an entry of in the section header table |
| e_shnum | the number of entries in the section header table |

- with these three fields, the file's section headers can be located and accessed

https://www.bottomupcs.com/elf.xhtml

## (4) locating program headers

- in the ELF (File) header structure type

| | |
|---|---|
| e_phoff | the offset in the file where the program header table starts |
| e_phentsize | the size of an entry of in the program header table |
| e_shnum | the number of entries in the program header table |

- with these three fields, the file's program headers can be located and accessed

https://www.bottomupcs.com/elf.xhtml

## TOC: Section header table

## Section header table (1)

- A section header table contains information describing the file's sections
- Every section has an entry in the table.
- Each entry gives information such as
    - the section name,
    - the section size, and so forth.

`https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html`

## Section header table (2)

- a section header table is
  an array of `Elf32_Shdr` entries.
- a section header table index is
  a subscript into this array

- object files used in link-editing
  *must have* a section header table
- other object files
  *might* or *might not* have a section header table

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Elf32_Shdr and section header table

```
typedef struct {
    Elf32_Word    sh_name;
    Elf32_Word    sh_type;  // <--
    Elf32_Word    sh_flags; // <--
    Elf32_Addr    sh_addr;
    Elf32_Off     sh_offset;
    Elf32_Word    sh_size;
    Elf32_Word    sh_link;
    Elf32_Word    sh_info;
    Elf32_Word    sh_addralign;
    Elf32_Word    sh_entsize;
} Elf32_Shdr;
```

- sh_type :
  categorizes the section's contents and semantics
- sh_flags :
  sections support 1-bit flags that describe miscellaneous attributes.

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## TOC: Group section

## Section of type SHT_GROUP (1)

sh_type = SHT_GROUP

- a section of type SHT_GROUP
  defines a grouping of sections
- the name of a symbol
  from one of the containing object's symbol tables
  provides a signature for the section group
- the section header of the SHT_GROUP section
  specifies the identifying symbol entry

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section of type SHT_GROUP (2)

sh_type = SHT_GROUP

- sh_link
  - the section header index of the associated symbol table
- sh_info
  - the symbol table index of an entry in the associated symbol table
  - the name of the specified symbol table entry provides a signature for the section group

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section of type SHT_GROUP (3)

sh_type = SHT_GROUP

- the sh_flags member of the section header contains 0
- the name of the section (sh_name) is not specified.

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section of type `SHT_GROUP` (4)

`sh_type` = `SHT_GROUP`

- the section data of a `SHT_GROUP` section is an array of `Elf32_Word` entries.
- the first entry is a flag word.
- The remaining entries are a sequence of section header indices

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section header's `sh_flags`

- `SHF_GROUP`
    - This section is a member of a section group
      perhaps the only one
    - the section must be referenced by a section of type `SHT_GROUP`
    - The `SHF_GROUP` flag can be set only for sections
      contained in relocatable objects,
      objects with the ELF header `e_type` member set to `ET_REL`

`https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html`

## Section groups (1)

- Some sections occur in *interrelated* groups.
- For example,
    - an out-of-line definition of an inline function might require,
    - in *addition* to the section containing its executable instructions,
    - a read-only data section containing *literals* referenced,
    - one or more debugging information sections and
    - other informational sections.

`https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html`

## Section groups (2)

- Furthermore, there may be internal references among these sections that would not make sense
  - if one of the sections were removed or
  - replaced by a duplicate from another object.
- Therefore, such groups must be included or omitted from the linked object *as a unit*.

`https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html`

## Section groups (4)

- The section data of a SHT_GROUP section is an array of Elf32_Word entries.
- The first entry is a flag word
- The remaining entries are a sequence of section header indices

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section groups (4')

- GRP_COMDAT
  - This is a COMDAT group.
  - It may duplicate another COMDAT group in another object file, where duplication is defined as having the same group signature
  - In such cases, only one of the duplicate groups will be retained by the link-editor, and the members of the remaining groups will be discarded

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section groups (5)

- The section header indices in the `SHT_GROUP` section identify the sections that make up the group.
- Each such section must have the `SHF_GROUP` flag set in its `sh_flags` section header member.
- If the link-editor decides to remove the section group, it will remove all members of the group.
- To facilitate removing a group without leaving dangling references and with only minimal processing of the symbol table, the following rules are followed:

`https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html`

## Section groups (6)

- References to the sections comprising a group from sections outside of the group must be made through symbol table entries with STB_GLOBAL or STB_WEAK binding and section index SHN_UNDEF .
- If there is a definition of the same symbol in the object containing the references, it must have a separate symbol table entry from the references.
- Sections outside of the group may not reference symbols with STB_LOCAL binding for addresses contained in the group's sections, including symbols with type STT_SECTION

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html

## Section groups (7)

- There may not be non-symbol references to the sections comprising a group from outside the group.
  - For example, you cannot use a group member's section header index in an `sh_link` or `sh_info` member.
- A symbol table entry that is defined
  relative to one of the group's sections and
  that is contained in a symbol table section
  that is not part of the group,
  will be removed if the group members are discarded.

https://docs.oracle.com/cd/E19683-01/816-1386/6m7qcoblh/index.html