

## 1. 简介

### 1.1 关键sections

### 1.2 示例

### 1.3 SHT和Sections的关系

### 1.4 Sections与ELF head映射

## 2 SHT(section head table)结构

### 2.1 三大核心(type, flags, addr)

### 2.2 sh\_name: 节区名在节区头部字符串表节区(shstrtab)的索引

### 2.3 sh\_type: 节区(Section)类型

#### 2.3.1 示例

### 2.4 sh\_flag: 节区在进程虚拟地址空间的属性

#### 2.4.1 示例

### 2.5 sh\_link 和 sh\_info : Section的类型和链接相关

#### 2.5.1 示例

# 1. 简介

## 1.1 关键sections

那什么是所谓 sections 呢？可以说，sections 是在ELF文件里头，用以装载内容数据的最小容器。

在ELF文件里面，每一个 sections 内都装载了性质属性都一样的内容，比如：

- **.text** 里装载了可执行代码；编译后执行语句都编译成机器代码，保存在 **.text** 中
- **.data** 里面装载了被初始化的数据；初始化的全局变量和静态局部变量都存储在 **.data** 中
- **.bss** 里面装载了未被初始化的数据；未初始化的全局变量和静态局部变量都存储在 **.bss** 中。未初始化的全局变量和局部静态变量默认值为 0，本来他们也是可以被放在 **.data** 中，但是因为他们都是 0，所以在 **.data** 中分配空间存在数据 0 是没有必要的。程序在运行时他们在确定要占内存空间，并且可执行文件必须记录所有未初始化的全局变量和局部静态变量的大小总和，即为 **.bss**。所以 **.bss** 只是为未初始化的全局变量和局部静态变量预留位置而已，它并没有内容，所以他在文件中也不占据空间。
- 以 **.rel** 打头的 sections 里面装载了重定位条目；
- **.symtab** 或者 **.dynsym** section 里面装载了符号信息；
- **.strtab** 或者 **.dynstr** section 里面装载了字符串信息；
- **.rodata** 只读数据区
- **.comment** 注释信息
- **.note.GNU-stack** 堆栈提示
- 其他还有为满足不同目的所设置的 section，比方满足调试的目的、满足动态链接与加载的目的等等。
- **.bss**：保存未初始化的数据，比如那些未初始化的全局变量。因为是“未初始化”，所以也没必要在文件中占用任何空间去记录其初始值（所以类型为 **SHT\_NOBITS**）。在程序开始运行时，系统会将 **.bss** 映射的内存区域清零。
- **.comment**：保存版本控制信息。
- **.data/.data1**：保存已初始化的数据。它们会在文件中占用存储空间，这与 **.bss** 不同。
- **.debug**：保存调试相关的信息。
- **.fini / .init**：分别保存进程退出和初始化时要执行的指令。**.init** 指令会在程序入口点(**main**)之前被执行。
- **.got**：保存全局偏移量表 (**global offset table**)。
- ~~在 Android 中，GOT 分为两部分：**.got** 和 **.got.plt**。其中 **.got** 表用来保存全局变量引用的地址，而 **.got.plt** 用来保存函数引用的地址。~~
- **.hash**：保存符号哈希表，用于快速查找与其对应的符号表中的符号。
- **.interp**：保存 ELF 程序解释器（比如 Android 下的动态链接器）的路径名。
- **.line**：保存用于调试的行号信息。
- **.note**：保存一些注释信息。
- **.plt**：保存过程链接表 (**procedure linkage table**)。每个外部定义的函数都会在 **PLT** 中有对应的一项，用于定位外部函数的地址。

- **.relname/.relaname**: 保存重定位表。比如: **.rel.dyn**、**.rel.plt**。
- **.rodata / .rodata1**: 保存程序中的只读数据。
- **.shstrtab**: 存储Section名称的字符串。保存一个字符串表, 这些字符串都是 **section** 的名字。
- **.strtab**: 保存的是普通的字符串。保存字符串表, 类似于 **.dynstr**, 但 **.dynstr** 中保存的都是那些需要动态链接的符号的名字。
- **.dynstr**: 存储的是符号名称的字符串。比如符号表中的每个符号都有一个 **st\_name(符号名)**, 他是指向字符串表的索引, 这个字符串表可能就保存在 **.dynstr**。
- **.dynamic**: 保存动态链接信息。
- **.dynsym**: 保存需要动态连接的符号表。**.dynsym**包含**.symtab**的一个子集, 比如共享库所需要在runtime加载的函数对应的symbols, 它是allocable的。共享库包含的**.dynsym**是runtime必需的, 是allocable的。
- **.symtab**: 包含大量linker,debugger需要的数据, 但并不为runtime必需, 它是non-allocable的; 保存符号表(非动态链接)。

**.symtab**包含大量linker,debugger需要的数据, 但并不为runtime必需, 它是non-allocable的; **.dynsym**包含**.symtab**的一个子集, 比如共享库所需要在runtime加载的函数对应的symbols, 它是allocable的。共享库包含的**.dynsym**是runtime必需的, 是allocable的。

- **.text**: 保存可执行的指令代码。

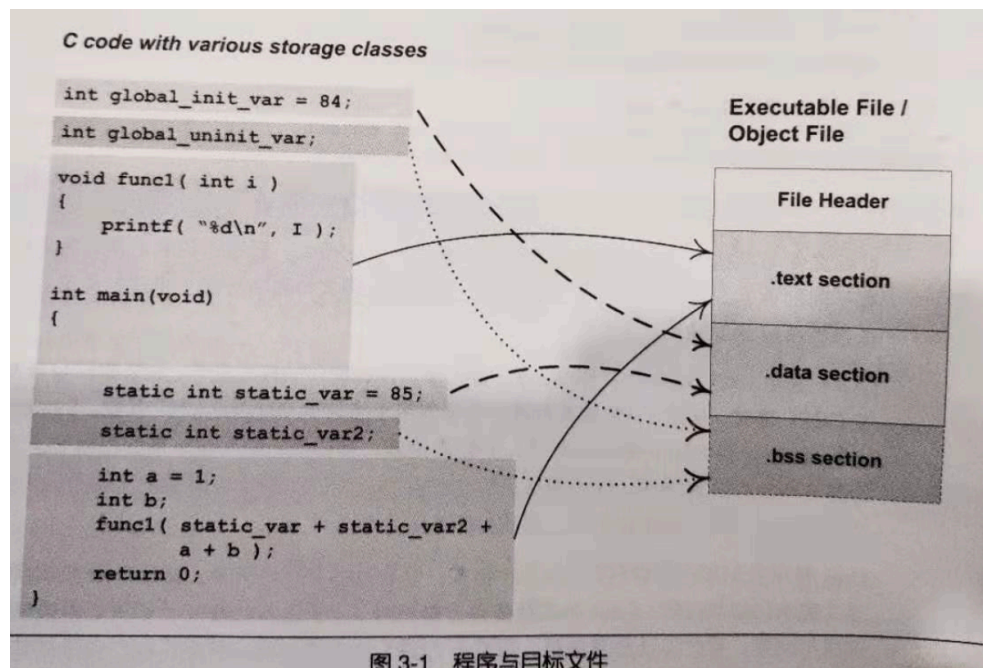


图 3-1 程序与目标文件

## 1.2 示例

使用 **readelf -S \*.so** 命令查看所有的sections

```
yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -S libtest.so
There are 25 section headers, starting at offset 0x2284:

Section Headers:
[Nr] Name                               Type             Addr             Off             Size            ES Flg Lk Inf Al
[ 0]                               NULL             00000000         000000         000000         00  0  0  0
[ 1] .note.android.id                     NOTE             00000134         000134         000098         00  A  0  0  4
[ 2] .note.gnu.build-id                   NOTE             000001cc         0001cc         000024         00  A  0  0  4
[ 3] .dynsym                             DYNSYM           000001f0         0001f0         000100         10  A  4  1  4
[ 4] .dynstr                             STRTAB           000002f0         0002f0         0000ec         00  A  0  0  1
[ 5] .hash                               HASH             000003dc         0003dc         000054         04  A  3  0  4
[ 6] .gnu.version                         VERSYM           00000430         000430         000020         02  A  3  0  2
[ 7] .gnu.version_d                       VERDEF           00000450         000450         00001c         00  A  4  1  4
[ 8] .gnu.version_r                       VERNEED          0000046c         00046c         000020         00  A  4  1  4
[ 9] .rel.dyn                             REL              0000048c         00048c         000048         08  A  3  0  4
[10] .rel.plt                             REL              000004d4         0004d4         000050         08  AI 3 18 4
[11] .plt                                 PROGBITS          00000524         000524         00008c         00  AX  0  0  4
[12] .text                               PROGBITS          000005b0         0005b0         0015a4         00  AX  0  0  4
[13] .ARM.extab                           PROGBITS          00001b54         001b54         00003c         00  A  0  0  4
[14] .ARM.exidx                           ARM_EXIDX         00001b90         001b90         000100         08  AL 12 0 4
[15] .rodata                             PROGBITS          00001c90         001c90         00000a         01  AMS 0 0 1
[16] .fini_array                          FINI_ARRAY        00002ea4         001ea4         000004         04  WA  0  0  4
[17] .dynamic                             DYNAMIC           00002ea8         001ea8         000108         08  WA  4  0  4
[18] .got                                 PROGBITS          00002fb0         001fb0         000050         00  WA  0  0  4
[19] .data                               PROGBITS          00003000         002000         000004         00  WA  0  0  4
[20] .bss                                NOBITS            00003004         002004         000000         00  WA  0  0  1
[21] .comment                             PROGBITS          00000000         002004         00012f         01  MS  0  0  1
[22] .note.gnu.gold-ve                     NOTE             00000000         002134         00001c         00  0  0  4
[23] .ARM.attributes                       ARM_ATTRIBUTES    00000000         002150         000036         00  0  0  1
[24] .shstrtab                             STRTAB           00000000         002186         0000fe         00  0  0  1

Key to Flags:
W (write), A (alloc), X (execute), M (merge), S (strings), I (info),
L (link order), O (extra OS processing required), G (group), T (TLS),
C (compressed), x (unknown), o (OS specific), E (exclude),
Y (noload), P (processor specific)
```

## 1.3 SHT和Sections的关系

节区中包含目标文件中的所有信息，除了:ELF 头部、程序头部表格、节区头部表格。节区满足以下条件:

1. 目标文件中的每个节区都有对应的节区头部描述它，反过来，有节区头部不意味着有节区。
2. 每个节区占用文件中一个连续字节区域(这个区域可能长度为 0)。
3. 文件中的节区不能重叠，不允许一个字节存在于两个节区中的情况发生。
4. 目标文件中可能包含非活动空间(INACTIVE SPACE)。这些区域不属于任何

## 1.4 Sections与ELF head映射

ELF head中出现了SHT的描述

ELF 头部中，**e\_shoff** 成员给出从文件头到节区头部表格的偏移字节数;**e\_shnum** 给出表格中条目数目;**e\_shentsize** 给出每个项目的字节数。从这些信息中可以确切地定位节区的具体位置、长度。

**e\_shoff**: 节区头部表格(SHT)的偏移量。节区头部表格(Section Header Table)的偏移量(按字节计算)。如果文件 没有节区头部表格，可以为 0。

**e\_shentsize:** 节区头部表格的表项大小。节区头部表格的表项大小(按字节计算)。即Section Header Table中每一项的大小

**e\_shnum:** 节区头部表格的表项数量。节区头部表格的表项数目，可以为0(只能运行时加载，不能用作编译链接，一般用于so的裁剪)。

**e\_shstrndx:** 节区头部表格中与节区名称字符串表相关的表项的索引。节区头部表格中与节区名称字符串表相关的表项的索引。如果文件没有节区名称字符串表，此参数可以为 SHN\_UNDEF。

## 2 SHT(section head table)结构

一个ELF文件中到底有哪些具体的 sections，由包含在这个ELF文件中的 section head table(SHT)决定。

在SHT中，针对每一个section，都设置有一个条目，用来描述对应的这个section，其内容主要包括该 section 的名称、类型、大小以及在整个ELF文件中的字节偏移位置等等。我们也可以在TISCV1.2规范中找到SHT表中条目的C结构定义：

```
typedef struct{
    Elf32_Word sh_name; //节区名，段名是一个字符串，它位于一个叫做shstrtab的字符串表。sh_name段名字符串是在".shstrtab"中的偏移。节区头部字符串表节区（Section Header String Table Section）的索引。名字是一个 NULL 结尾的字符串。
    Elf32_Word sh_type; //为节区类型, 1) PROGBITS,表示此section的内容由程序来解释,.text和.data都是; 2) NOBITS,表示此段不占用文件空间 3) STRTAB是表示字符串表;4) SYMTAB表示符号表等等
    Elf32_Word sh_flags; //节区标志。可以在该命令的最末看出不同flag标志的含义。其中Alloc，表示该section在进程执行过程中占用内存
    Elf32_Addr sh_addr; //节区虚拟地址。如果节区可以被加载，此成员为被加载后在进程地址空间中的虚拟地址；否则，此字段为 0。
    Elf32_Off sh_offset; //节区偏移。节区的第一个字节与文件头之间的偏移。如果该section存在于文件中，则表示再文件中的偏移；否则无意义。比如sh_offset对于BBS Section来说就没有意义。
    Elf32_Word sh_size; //节区的长度（字节数）。
    Elf32_Word sh_link; //此成员给出节区头部表索引链接。其具体的解释依赖于节区类型。
    Elf32_Word sh_info; //此成员给出附加信息，其解释依赖于节区类型。
```

Elf32\_Word sh\_addralign; //某些节区带有地址对齐约束。有些Section对地址对齐有要求，比如我们假设有个Section起始位置包含一个double变量，因为Intel x86系统要求浮点数的存储地址必须是本身的整数倍，也就是说保存double变量地址的必须是8字节的整数倍。这样对一个Section来说，它的sh\_addr必须是8的整数倍。由于地址对齐的数量都是2的指数倍，sh\_addralign表示是地址对齐数量中的指数，即sh\_addralign==3表示对齐为2的3次方倍，即8倍，依此类推。所以一个段的地址sh\_addr必须满足下面的条件，即sh\_addr%(2 \*\* sh\_addralign)==0。 \*\*表示指数运算。

如果sh\_addralign为0或1，则表示该Section没有对齐要求。

Elf32\_Word sh\_entsize; //Section Entry Size 项的长度。有些Section包含了一些固定大小的项，比如符号表，它包含的每个符号所占的大小都一样。对于这种Section，sh\_entsize表示每个项的大小。如果为0，则表示该Section不包含固定大小的项。

```
} Elf32_Shdr;
```

在010Editor中的展示为：

▼ struct section_header_table		2284h	3E8h
▶ struct section_table_entry32_t section_table_element[0]	SHN_UNDEF	2284h	28h
▶ struct section_table_entry32_t section_table_element[1]	.note.android.ident	22ACh	28h
▶ struct section_table_entry32_t section_table_element[2]	.note.gnu.build-id	22D4h	28h
▶ struct section_table_entry32_t section_table_element[3]	.dynsym	22FCh	28h
▶ struct section_table_entry32_t section_table_element[4]	.dynstr	2324h	28h
▶ struct section_table_entry32_t section_table_element[5]	.hash	234Ch	28h
▶ struct section_table_entry32_t section_table_element[6]	.gnu.version	2374h	28h
▶ struct section_table_entry32_t section_table_element[7]	.gnu.version_d	239Ch	28h
▶ struct section_table_entry32_t section_table_element[8]	.gnu.version_r	23C4h	28h
▶ struct section_table_entry32_t section_table_element[9]	.rel.dyn	23ECh	28h
▶ struct section_table_entry32_t section_table_element[10]	.rel.plt	2414h	28h
▶ struct section_table_entry32_t section_table_element[11]	.plt	243Ch	28h
▶ struct section_table_entry32_t section_table_element[12]	.text	2464h	28h
▶ struct section_table_entry32_t section_table_element[13]	.ARM.extab	248Ch	28h
▶ struct section_table_entry32_t section_table_element[14]	.ARM.exidx	24B4h	28h
▶ struct section_table_entry32_t section_table_element[15]	.rodata	24DCh	28h
▶ struct section_table_entry32_t section_table_element[16]	.fini_array	2504h	28h
▶ struct section_table_entry32_t section_table_element[17]	.dynamic	252Ch	28h
▶ struct section_table_entry32_t section_table_element[18]	.got	2554h	28h
▶ struct section_table_entry32_t section_table_element[19]	.data	257Ch	28h
▶ struct section_table_entry32_t section_table_element[20]	.bss	25A4h	28h
▶ struct section_table_entry32_t section_table_element[21]	.comment	25CCh	28h
▶ struct section_table_entry32_t section_table_element[22]	.note.gnu.gold-versi...	25F4h	28h
▶ struct section_table_entry32_t section_table_element[23]	.ARM.attributes	261Ch	28h
▶ struct section_table_entry32_t section_table_element[24]	.shstrtab	2644h	28h

## 2.1 三大核心(type, flags, addr)

- type:

1. PROGBITS,表示此section的内容由程序来解释,.text和.data都是
2. NOBITS,表示此段不占用文件空间
3. STRTAB是表示字符串表



#### 4. SYMTAB表示符号表等等

- flags: 可以在该命令的最末看出不同flag标志的含义。其中Alloc, 表示该section在进程执行过程中占用内存
- addr : 如果section将出现在进程的内存映像中, 此成员给出section的第一个字节应处的位置。

## 2.2 sh\_name: 节区名在节区头部字符串表节区(shstrtab)的索引

shstrtab在哪里: 从ELF 头部文件中的e\_shstrndx能够获取该section的索引  
还是从libtest.so为例, 从ELF Head中e\_shstrndx为24, 找到该section的内容, 发现如下:

▼ struct section_table_entry32_t section_table_element[24]	.shstrtab	2644h	28h	Fg:	Bg:	
▼ struct s_name32_t s_name	.shstrtab	2644h	4h	Fg:	Bg:	
enum s_name32_e s_name_off	1h	2644h	4h	Fg:	Bg:	
▶ string s_name_str[10]	.shstrtab	2187h	Ah	Fg:	Bg:	
enum s_type32_e s_type	SHT_STRTAB (3)	2648h	4h	Fg:	Bg:	
enum s_flags32_e s_flags	SF32_None (0)	264Ch	4h	Fg:	Bg:	
Elf32_Addr s_addr	0x00000000	2650h	4h	Fg:	Bg:	
Elf32_Off s_offset	2186h	2654h	4h	Fg:	Bg:	
Elf32_Xword s_size	254	2658h	4h	Fg:	Bg:	
Elf32_Word s_link	0	266Ch	4h	Fg:	Bg:	
Elf32_Word s_info	0	2660h	4h	Fg:	Bg:	
Elf32_Xword s_addralign	1	2664h	4h	Fg:	Bg:	
Elf32_Xword s_entsize	0	2668h	4h	Fg:	Bg:	
▶ char s_data[254]		2186h	FEh	Fg:	Bg:	

s\_name代表的索引为1, 通过**readelf -p 24 libtest.so**获取shstrtab section的内容如下:

```
yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -p 24 libtest.so

String dump of section '.shstrtab':
[      1] .shstrtab
```

有意思了, shstrtab的名称竟然包含在其属性中, s\_name中是shstrtab的第一个, 指向的恰好是.shstrtab

## 2.3 sh\_type: 节区(Section)类型

为节区的内容和语义进行分类, 也就是说section name并不进行任何属性绑定, 属性绑定的是type

名称	取值	说明
SHT_NULL	0	此值标志节区头部是非活动的，没有对应
SHT_PROGBITS	1	此section的内容由程序来解释 程序段、代码段、数据段都是这种类型的
SHT_SYMTAB	2	该节区内容为符号表 目前目标文件对每种类型的节区都只能包 般，SHT_SYMTAB 节区提供用于链接编 接。
SHT_STRTAB	3	该节区内容为字符串表 目标文件可能包含多个字符串表节区。
SHT_RELA	4	该节区内容为重定位表 其中可能会有补齐内容（addend），例： 拥有多个重定位节区。
SHT_HASH	5	此节区包含符号哈希表 所有参与动态链接的目标都必须包含一个 表，不过此限制将来可能会解除。
SHT_DYNAMIC	6	动态链接信息 目前一个目标文件中只能包含一个动态节
SHT_NOTE	7	提示性信息 此节区包含以某种方式来标记文件的信息
SHT_NOBITS	8	该节区再文件中没有内容，比如.bss段 这种类型的节区不占用文件中的 节区不包含任何字节，成员sh_offset 中
SHT_REL	9	该段包含重定位信息 此节区包含重定位表项，其中没有补齐 型。目标文件中可以拥有多个重定位节区
SHT_SHLIB	10	保留 此节区被保留，不过其语义是未规定的。

2.3.1 示例

```

yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -S libtest.so
There are 25 section headers, starting at offset 0x2284:

Section Headers:
  [Nr] Name                               Type            Addr      Off      Size    ES Flg Lk  Inf Al
  [ 0]                               NULL            00000000  000000  000000  00      0  0  0
  [ 1] .note.android.id                     NOTE            00000134  000134  000098  00      A  0  0  4
  [ 2] .note.gnu.build-id                  NOTE            000001cc  0001cc  000024  00      A  0  0  4
  [ 3] .dynsym                             DYNSYM          000001f0  0001f0  000100  10      A  4  1  4
  [ 4] .dynstr                             STRTAB          000002f0  0002f0  0000ec  00      A  0  0  1
  [ 5] .shstrtab                           STRTAB          000003f0  0003f0  000000  00      A  0  0  1
  [ 6] .symtab                             SYMTAB          000004f0  0004f0  000000  00      A  0  0  1
  [ 7] .text                               PROGBITS        000005f0  0005f0  000000  00      A  0  0  1
  [ 8] .data                               PROGBITS        000006f0  0006f0  000000  00      A  0  0  1
  [ 9] .bss                               NOBITS          000007f0  0007f0  000000  00      A  0  0  1
  [10] .comment                             PROGBITS        000008f0  0008f0  000000  00      A  0  0  1
  [11] .eh_frame                           PROGBITS        000009f0  0009f0  000000  00      A  0  0  1
  [12] .eh_frame                            PROGBITS        00000af0  000af0  000000  00      A  0  0  1
  [13] .eh_frame                            PROGBITS        00000bf0  000bf0  000000  00      A  0  0  1
  [14] .eh_frame                            PROGBITS        00000cf0  000cf0  000000  00      A  0  0  1
  [15] .eh_frame                            PROGBITS        00000df0  000df0  000000  00      A  0  0  1
  [16] .eh_frame                            PROGBITS        00000ef0  000ef0  000000  00      A  0  0  1
  [17] .eh_frame                            PROGBITS        00000ff0  000ff0  000000  00      A  0  0  1
  [18] .eh_frame                            PROGBITS        000010f0  0010f0  000000  00      A  0  0  1
  [19] .eh_frame                            PROGBITS        000011f0  0011f0  000000  00      A  0  0  1
  [20] .eh_frame                            PROGBITS        000012f0  0012f0  000000  00      A  0  0  1
  [21] .eh_frame                            PROGBITS        000013f0  0013f0  000000  00      A  0  0  1
  [22] .eh_frame                            PROGBITS        000014f0  0014f0  000000  00      A  0  0  1
  [23] .eh_frame                            PROGBITS        000015f0  0015f0  000000  00      A  0  0  1
  [24] .eh_frame                            PROGBITS        000016f0  0016f0  000000  00      A  0  0  1
  [25] .eh_frame                            PROGBITS        000017f0  0017f0  000000  00      A  0  0  1
  [26] .eh_frame                            PROGBITS        000018f0  0018f0  000000  00      A  0  0  1
  [27] .eh_frame                            PROGBITS        000019f0  0019f0  000000  00      A  0  0  1
  [28] .eh_frame                            PROGBITS        00001af0  001af0  000000  00      A  0  0  1
  [29] .eh_frame                            PROGBITS        00001bf0  001bf0  000000  00      A  0  0  1
  [30] .eh_frame                            PROGBITS        00001cf0  001cf0  000000  00      A  0  0  1
  [31] .eh_frame                            PROGBITS        00001df0  001df0  000000  00      A  0  0  1
  [32] .eh_frame                            PROGBITS        00001ef0  001ef0  000000  00      A  0  0  1
  [33] .eh_frame                            PROGBITS        00001ff0  001ff0  000000  00      A  0  0  1
  [34] .eh_frame                            PROGBITS        000020f0  0020f0  000000  00      A  0  0  1
  [35] .eh_frame                            PROGBITS        000021f0  0021f0  000000  00      A  0  0  1
  [36] .eh_frame                            PROGBITS        000022f0  0022f0  000000  00      A  0  0  1
  [37] .eh_frame                            PROGBITS        000023f0  0023f0  000000  00      A  0  0  1
  [38] .eh_frame                            PROGBITS        000024f0  0024f0  000000  00      A  0  0  1
  [39] .eh_frame                            PROGBITS        000025f0  0025f0  000000  00      A  0  0  1
  [40] .eh_frame                            PROGBITS        000026f0  0026f0  000000  00      A  0  0  1
  [41] .eh_frame                            PROGBITS        000027f0  0027f0  000000  00      A  0  0  1
  [42] .eh_frame                            PROGBITS        000028f0  0028f0  000000  00      A  0  0  1
  [43] .eh_frame                            PROGBITS        000029f0  0029f0  000000  00      A  0  0  1
  [44] .eh_frame                            PROGBITS        00002af0  002af0  000000  00      A  0  0  1
  [45] .eh_frame                            PROGBITS        00002bf0  002bf0  000000  00      A  0  0  1
  [46] .eh_frame                            PROGBITS        00002cf0  002cf0  000000  00      A  0  0  1
  [47] .eh_frame                            PROGBITS        00002df0  002df0  000000  00      A  0  0  1
  [48] .eh_frame                            PROGBITS        00002ef0  002ef0  000000  00      A  0  0  1
  [49] .eh_frame                            PROGBITS        00002ff0  002ff0  000000  00      A  0  0  1
  [50] .eh_frame                            PROGBITS        000030f0  0030f0  000000  00      A  0  0  1
  [51] .eh_frame                            PROGBITS        000031f0  0031f0  000000  00      A  0  0  1
  [52] .eh_frame                            PROGBITS        000032f0  0032f0  000000  00      A  0  0  1
  [53] .eh_frame                            PROGBITS        000033f0  0033f0  000000  00      A  0  0  1
  [54] .eh_frame                            PROGBITS        000034f0  0034f0  000000  00      A  0  0  1
  [55] .eh_frame                            PROGBITS        000035f0  0035f0  000000  00      A  0  0  1
  [56] .eh_frame                            PROGBITS        000036f0  0036f0  000000  00      A  0  0  1
  [57] .eh_frame                            PROGBITS        000037f0  0037f0  000000  00      A  0  0  1
  [58] .eh_frame                            PROGBITS        000038f0  0038f0  000000  00      A  0  0  1
  [59] .eh_frame                            PROGBITS        000039f0  0039f0  000000  00      A  0  0  1
  [60] .eh_frame                            PROGBITS        00003af0  003af0  000000  00      A  0  0  1
  [61] .eh_frame                            PROGBITS        00003bf0  003bf0  000000  00      A  0  0  1
  [62] .eh_frame                            PROGBITS        00003cf0  003cf0  000000  00      A  0  0  1
  [63] .eh_frame                            PROGBITS        00003df0  003df0  000000  00      A  0  0  1
  [64] .eh_frame                            PROGBITS        00003ef0  003ef0  000000  00      A  0  0  1
  [65] .eh_frame                            PROGBITS        00003ff0  003ff0  000000  00      A  0  0  1
  [66] .eh_frame                            PROGBITS        000040f0  0040f0  000000  00      A  0  0  1
  [67] .eh_frame                            PROGBITS        000041f0  0041f0  000000  00      A  0  0  1
  [68] .eh_frame                            PROGBITS        000042f0  0042f0  000000  00      A  0  0  1
  [69] .eh_frame                            PROGBITS        000043f0  0043f0  000000  00      A  0  0  1
  [70] .eh_frame                            PROGBITS        000044f0  0044f0  000000  00      A  0  0  1
  [71] .eh_frame                            PROGBITS        000045f0  0045f0  000000  00      A  0  0  1
  [72] .eh_frame                            PROGBITS        000046f0  0046f0  000000  00      A  0  0  1
  [73] .eh_frame                            PROGBITS        000047f0  0047f0  000000  00      A  0  0  1
  [74] .eh_frame                            PROGBITS        000048f0  0048f0  000000  00      A  0  0  1
  [75] .eh_frame                            PROGBITS        000049f0  0049f0  000000  00      A  0  0  1
  [76] .eh_frame                            PROGBITS        00004af0  004af0  000000  00      A  0  0  1
  [77] .eh_frame                            PROGBITS        00004bf0  004bf0  000000  00      A  0  0  1
  [78] .eh_frame                            PROGBITS        00004cf0  004cf0  000000  00      A  0  0  1
  [79] .eh_frame                            PROGBITS        00004df0  004df0  000000  00      A  0  0  1
  [80] .eh_frame                            PROGBITS        00004ef0  004ef0  000000  00      A  0  0  1
  [81] .eh_frame                            PROGBITS        00004ff0  004ff0  000000  00      A  0  0  1
  [82] .eh_frame                            PROGBITS        000050f0  0050f0  000000  00      A  0  0  1
  [83] .eh_frame                            PROGBITS        000051f0  0051f0  000000  00      A  0  0  1
  [84] .eh_frame                            PROGBITS        000052f0  0052f0  000000  00      A  0  0  1
  [85] .eh_frame                            PROGBITS        000053f0  0053f0  000000  00      A  0  0  1
  [86] .eh_frame                            PROGBITS        000054f0  0054f0  000000  00      A  0  0  1
  [87] .eh_frame                            PROGBITS        000055f0  0055f0  000000  00      A  0  0  1
  [88] .eh_frame                            PROGBITS        000056f0  0056f0  000000  00      A  0  0  1
  [89] .eh_frame                            PROGBITS        000057f0  0057f0  000000  00      A  0  0  1
  [90] .eh_frame                            PROGBITS        000058f0  0058f0  000000  00      A  0  0  1
  [91] .eh_frame                            PROGBITS        000059f0  0059f0  000000  00      A  0  0  1
  [92] .eh_frame                            PROGBITS        00005af0  005af0  000000  00      A  0  0  1
  [93] .eh_frame                            PROGBITS        00005bf0  005bf0  000000  00      A  0  0  1
  [94] .eh_frame                            PROGBITS        00005cf0  005cf0  000000  00      A  0  0  1
  [95] .eh_frame                            PROGBITS        00005df0  005df0  000000  00      A  0  0  1
  [96] .eh_frame                            PROGBITS        00005ef0  005ef0  000000  00      A  0  0  1
  [97] .eh_frame                            PROGBITS        00005ff0  005ff0  000000  00      A  0  0  1
  [98] .eh_frame                            PROGBITS        000060f0  0060f0  000000  00      A  0  0  1
  [99] .eh_frame                            PROGBITS        000061f0  0061f0  000000  00      A  0  0  1
  [100] .eh_frame                            PROGBITS        000062f0  0062f0  000000  00      A  0  0  1
  [101] .eh_frame                            PROGBITS        000063f0  0063f0  000000  00      A  0  0  1
  [102] .eh_frame                            PROGBITS        000064f0  0064f0  000000  00      A  0  0  1
  [103] .eh_frame                            PROGBITS        000065f0  0065f0  000000  00      A  0  0  1
  [104] .eh_frame                            PROGBITS        000066f0  0066f0  000000  00      A  0  0  1
  [105] .eh_frame                            PROGBITS        000067f0  0067f0  000000  00      A  0  0  1
  [106] .eh_frame                            PROGBITS        000068f0  0068f0  000000  00      A  0  0  1
  [107] .eh_frame                            PROGBITS        000069f0  0069f0  000000  00      A  0  0  1
  [108] .eh_frame                            PROGBITS        00006af0  006af0  000000  00      A  0  0  1
  [109] .eh_frame                            PROGBITS        00006bf0  006bf0  000000  00      A  0  0  1
  [110] .eh_frame                            PROGBITS        00006cf0  006cf0  000000  00      A  0  0  1
  [111] .eh_frame                            PROGBITS        00006df0  006df0  000000  00      A  0  0  1
  [112] .eh_frame                            PROGBITS        00006ef0  006ef0  000000  00      A  0  0  1
  [113] .eh_frame                            PROGBITS        00006ff0  006ff0  000000  00      A  0  0  1
  [114] .eh_frame                            PROGBITS        000070f0  0070f0  000000  00      A  0  0  1
  [115] .eh_frame                            PROGBITS        000071f0  0071f0  000000  00      A  0  0  1
  [116] .eh_frame                            PROGBITS        000072f0  0072f0  000000  00      A  0  0  1
  [117] .eh_frame                            PROGBITS        000073f0  0073f0  000000  00      A  0  0  1
  [118] .eh_frame                            PROGBITS        000074f0  0074f0  000000  00      A  0  0  1
  [119] .eh_frame                            PROGBITS        000075f0  0075f0  000000  00      A  0  0  1
  [120] .eh_frame                            PROGBITS        000076f0  0076f0  000000  00      A  0  0  1
  [121] .eh_frame                            PROGBITS        000077f0  0077f0  000000  00      A  0  0  1
  [122] .eh_frame                            PROGBITS        000078f0  0078f0  000000  00      A  0  0  1
  [123] .eh_frame                            PROGBITS        000079f0  0079f0  000000  00      A  0  0  1
  [124] .eh_frame                            PROGBITS        00007af0  007af0  000000  00      A  0  0  1
  [125] .eh_frame                            PROGBITS        00007bf0  007bf0  000000  00      A  0  0  1
  [126] .eh_frame                            PROGBITS        00007cf0  007cf0  000000  00      A  0  0  1
  [127] .eh_frame                            PROGBITS        00007df0  007df0  000000  00      A  0  0  1
  [128] .eh_frame                            PROGBITS        00007ef0  007ef0  000000  00      A  0  0  1
  [129] .eh_frame                            PROGBITS        00007ff0  007ff0  000000  00      A  0  0  1
  [130] .eh_frame                            PROGBITS        000080f0  0080f0  000000  00      A  0  0  1
  [131] .eh_frame                            PROGBITS        000081f0  0081f0  000000  00      A  0  0  1
  [132] .eh_frame                            PROGBITS        000082f0  0082f0  000000  00      A  0  0  1
  [133] .eh_frame                            PROGBITS        000083f0  0083f0  000000  00      A  0  0  1
  [134] .eh_frame                            PROGBITS        000084f0  0084f0  000000  00      A  0  0  1
  [135] .eh_frame                            PROGBITS        000085f0  0085f0  000000  00      A  0  0  1
  [136] .eh_frame                            PROGBITS        000086f0  0086f0  000000  00      A  0  0  1
  [137] .eh_frame                            PROGBITS        000087f0  0087f0  000000  00      A  0  0  1
  [138] .eh_frame                            PROGBITS        000088f0  0088f0  000000  00      A  0  0  1
  [139] .eh_frame                            PROGBITS        000089f0  0089f0  000000  00      A  0  0  1
  [140] .eh_frame                            PROGBITS        00008af0  008af0  000000  00      A  0  0  1
  [141] .eh_frame                            PROGBITS        00008bf0  008bf0  000000  00      A  0  0  1
  [142] .eh_frame                            PROGBITS        00008cf0  008cf0  000000  00      A  0  0  1
  [143] .eh_frame                            PROGBITS        00008df0  008df0  000000  00      A  0  0  1
  [144] .eh_frame                            PROGBITS        00008ef0  008ef0  000000  00      A  0  0  1
  [145] .eh_frame                            PROGBITS        00008ff0  008ff0  000000  00      A  0  0  1
  [146] .eh_frame                            PROGBITS        000090f0  0090f0  000000  00      A  0  0  1
  [147] .eh_frame                            PROGBITS        000091f0  0091f0  000000  00      A  0  0  1
  [148] .eh_frame                            PROGBITS        000092f0  0092f0  000000  00      A  0  0  1
  [149] .eh_frame                            PROGBITS        000093f0  0093f0  000000  00      A  0  0  1
  [150] .eh_frame                            PROGBITS        000094f0  0094f0  000000  00      A  0  0  1
  [151] .eh_frame                            PROGBITS        000095f0  0095f0  000000  00      A  0  0  1
  [152] .eh_frame                            PROGBITS        000096f0  0096f0  000000  00      A  0  0  1
  [153] .eh_frame                            PROGBITS        000097f0  0097f0  000000  00      A  0  0  1
  [154] .eh_frame                            PROGBITS        000098f0  0098f0  000000  00      A  0  0  1
  [155] .eh_frame                            PROGBITS        000099f0  0099f0  000000  00      A  0  0  1
  [156] .eh_frame                            PROGBITS        00009af0  009af0  000000  00      A  0  0  1
  [157] .eh_frame                            PROGBITS        00009bf0  009bf0  000000  00      A  0  0  1
  [158] .eh_frame                            PROGBITS        00009cf0  009cf0  000000  00      A  0  0  1
  [159] .eh_frame                            PROGBITS        00009df0  009df0  000000  00      A  0  0  1
  [160] .eh_frame                            PROGBITS        00009ef0  009ef0  000000  00      A  0  0  1
  [161] .eh_frame                            PROGBITS        00009ff0  009ff0  000000  00      A  0  0  1
  [162] .eh_frame                            PROGBITS        0000a0f0  00a0f0  000000  00      A  0  0  1
  [163] .eh_frame                            PROGBITS        0000a1f0  00a1f0  000000  00      A  0  0  1
  [164] .eh_frame                            PROGBITS        0000a2f0  00a2f0  000000  00      A  0  0  1
  [165] .eh_frame                            PROGBITS        0000a3f0  00a3f0  000000  00      A  0  0  1
  [166] .eh_frame                            PROGBITS        0000a4f0  00a4f0  000000  00      A  0  0  1
  [167] .eh_frame                            PROGBITS        0000a5f0  00a5f0  000000  00      A  0  0  1
  [168] .eh_frame                            PROGBITS        0000a6f0  00a6f0  000000  00      A  0  0  1
  [169] .eh_frame                            PROGBITS        0000a7f0  00a7f0  000000  00      A  0  0  1
  [170] .eh_frame                            PROGBITS        0000a8f0  00a8f0  000000  00      A  0  0  1
  [171] .eh_frame                            PROGBITS        0000a9f0  00a9f0  000000  00      A  0  0  1
  [172] .eh_frame                            PROGBITS        0000aaf0  00aaf0  000000  00      A  0  0  1
  [173] .eh_frame                            PROGBITS        0000abf0  00abf0  000000  00      A  0  0  1
  [174] .eh_frame                            PROGBITS        0000acf0  00acf0  000000  00      A  0  0  1
  [175] .eh_frame                            PROGBITS        0000adf0  00adf0  000000  00      A  0  0  1
  [176] .eh_frame                            PROGBITS        0000aef0  00aef0  000000  00      A  0  0  1
  [177] .eh_frame                            PROGBITS        0000aff0  00aff0  000000  00      A  0  0  1
  [178] .eh_frame                            PROGBITS        0000b0f0  00b0f0  000000  00      A  0  0  1
  [179] .eh_frame                            PROGBITS        0000b1f0  00b1f0  000000  00      A  0  0  1
  [180] .eh_frame                            PROGBITS        0000b2f0  00b2f0  000000  00      A  0  0  1
  [181] .eh_frame                            PROGBITS        0000b3f0  00b3f0  000000  00      A  0  0  1
  [182] .eh_frame                            PROGBITS        0000b4f0  00b4f0  000000  00      A  0  0  1
  [183] .eh_frame                            PROGBITS        0000b5f0  00b5f0  000000  00      A  0  0  1
  [184] .eh_frame                            PROGBITS        0000b6f0  00b6f0  000000  00      A  0  0  1
  [185] .eh_frame                            PROGBITS        0000b7f0  00b7f0  000000  00      A  0  0  1
  [186] .eh_frame                            PROGBITS        0000b8f0  00b8f0  000000  00      A  0  0  1
  [187] .eh_frame                            PROGBITS        0000b9f0  00b9f0  000000  00      A  0  0  1
  [188] .eh_frame                            PROGBITS        0000baf0  00baf0  000000  00      A  0  0  1
  [189] .eh_frame                            PROGBITS        0000bbf0  00bbf0  000000  00      A  0  0  1
  [190] .eh_frame                            PROGBITS        0000bcf0  00bcf0  000000  00      A  0  0  1
  [191] .eh_frame                            PROGBITS        0000bdf0  00bdf0  000000  00      A  0  0  1
  [192] .eh_frame                            PROGBITS        0000bef0  00bef0  000000  00      A  0  0  1
  [193] .eh_frame                            PROGBITS        0000bff0  00bff0  000000  00      A  0  0  1
  [194] .eh_frame                            PROGBITS        0000c0f0  00c0f0  000000  00      A  0  0  1
  [195] .eh_frame                            PROGBITS        0000c1f0  00c1f0  000000  00      A  0  0  1
  [196] .eh_frame                            PROGBITS        0000c2f0  00c2f0  000000  00      A  0  0  1
  [197] .eh_frame                            PROGBITS        0000c3f0  00c3f0  000000  00      A  0  0  1
  [198] .eh_frame                            PROGBITS        0000c4f0  00c4f0  000000  00      A  0  0  1
  [199] .eh_frame                            PROGBITS        0000c5f0  00c5f0  000000  00      A  0  0  1
  [200] .eh_frame                            PROGBITS        0000c6f0  00c6f0  000000  00      A  0  0  1
  [201] .eh_frame                            PROGBITS        0000c7f0  00c7f0  000000  00      A  0  0  1
  [202] .eh_frame                            PROGBITS        0000c8f0  00c8f0  000000  00      A  0  0  1
  [203] .eh_frame                            PROGBITS        0000c9f0  00c9f0  000000  00      A  0  0  1
  [204] .eh_frame                            PROGBITS        0000caf0  00caf0  000000  00      A  0  0  1
  [205] .eh_frame                            PROGBITS        0000cbf0  00cbf0  000000  00      A  0  0  1
  [206] .eh_frame                            PROGBITS        0000ccf0  00ccf0  000000  00      A  0  0  1
  [207] .eh_frame                            PROGBITS        0000cdf0  00cdf0  000000  00      A  0  0  1
  [208] .eh_frame                            PROGBITS        0000cef0  00cef0  000000  00      A  0  0  1
  [209] .eh_frame                            PROGBITS        0000cff0  00cff0  000000  00      A  0  0  1
  [210] .eh_frame                            PROGBITS        0000d0f0  00d0f0  000000  00      A  0  0  1
  [211] .eh_frame                            PROGBITS        0000d1f0  00d1f0  000000  00      A  0  0  1
  [212] .eh_frame                            PROGBITS        0000d2f0  00d2f0  000000  00      A  0  0  1
  [213] .eh_frame                            PROGBITS        0000d3f0  00d3f0  000000  00      A  0  0  1
  [214] .eh_frame                            PROGBITS        0000d4f0  00d4f0  000000  00      A  0  0  1
  [215] .eh_frame                            PROGBITS        0000d5f0  00d5f0  000000  00      A  0  0  1
  [216] .eh_frame                            PROGBITS        0000d6f0  00d6f0  000000  00      A  0  0  1
  [217] .eh_frame                            PROGBITS        0000d7f0  00d7f0  0000
```



名称	取值	含义
SHF_WRITE	0x1	节区包含进程执行过程中将可写的数据。
SHF_ALLOC	0x2	表示该节区再进程空间中需要分配空间 某些指示或控制节区并不出现于目标文件的 此位应设置为 0。像代码段、数据段和.bss。
SHF_EXECINSTR	0x4	该节区再进程空间中可以被执行，一般指代 节区包含可执行的机器指令
SHF_MASKPROC	0xF0000000	所有包含于此掩码中的四位都用于处理器专

2.4.1 示例

```

yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -S libtest.so
There are 25 section headers, starting at offset 0x2284:

Section Headers:
 [Nr] Name                Type              Addr      Off      Size    ES Flg Lk Inf Al
 [ 0]                     NULL              00000000  000000  000000  00  0  0  0  0
 [ 1] .note.android.id     NOTE              00000134  000134  000098  00  A  0  0  4
 [ 2] .note.gnu.build-id  NOTE              000001cc  0001cc  000024  00  A  0  0  4
 [ 3] .dynsym              DYNSYM            000001f0  0001f0  000100  10  A  4  1  4
 [ 4] .dynstr              STRTAB            000002f0  0002f0  0000ec  00  A  0  0  1
 [ 5] .hash               HASH              000003dc  0003dc  000054  04  A  3  0  4
 [ 6] .gnu.version         VERSYM            00000430  000430  000020  02  A  3  0  2
 [ 7] .gnu.version_d       VERDEF            00000450  000450  00001c  00  A  4  1  4
 [ 8] .gnu.version_r       VERNEED           0000046c  00046c  000020  00  A  4  1  4
 [ 9] .rel.dyn             REL               0000048c  00048c  000048  08  A  3  0  4
[10] .rel.plt             REL               000004d4  0004d4  000050  08  AI 3 18 4
[11] .plt                PROGBITS           00000524  000524  00008c  00  AX 0 0 4
[12] .text               PROGBITS           000005b0  0005b0  0015a4  00  AX 0 0 4
[13] .ARM.extab           PROGBITS           00001b54  001b54  00003c  00  A  0  0  4
[14] .ARM.exidx           ARM_EXIDX           00001b90  001b90  000100  08  AL 12 0 4
[15] .rodata              PROGBITS           00001c90  001c90  00000a  01  AMS 0 0 1
[16] .fini_array          FINI_ARRAY         00002ea4  001ea4  000004  04  WA 0 0 4
[17] .dynamic             DYNAMIC            00002ea8  001ea8  000108  08  WA 4 0 4
[18] .got                 PROGBITS           00002fb0  001fb0  000050  00  WA 0 0 4
[19] .data                PROGBITS           00003000  002000  000004  00  WA 0 0 4
[20] .bss                 NOBITS             00003004  002004  000000  00  WA 0 0 1
[21] .comment             PROGBITS           00000000  002004  00012f  01  MS 0 0 1
[22] .note.gnu.gold-ve    NOTE               00000000  002134  00001c  00  0  0  0  4
[23] .ARM.attributes      ARM_ATTRIBUTES     00000000  002150  000036  00  0  0  0  1
[24] .shstrtab            STRTAB             00000000  002186  0000fe  00  0  0  0  1

```

Key to Flags:

W (write), A (alloc), X (execute), M (merge), S (strings), I (info),  
L (link order), O (extra OS processing required), G (group), T (TLS),  
C (compressed), x (unknown), o (OS specific), E (exclude),  
y (noread), p (processor specific)

## 2.5 sh\_link 和 sh\_info：Section的类型和链接相关

Section的类型和链接相关，比如重定位表、符号表等，那么sh\_link和sh\_info两个成员包含的意义如下。对于其他节区，这两个成员没有意义。

sh_type	sh_link	sh_info
SHT_DYNAMIC	该节区所使用的字符串表在节区表(Section Header Table)中的下标	0
2.5.1 示例 SHT_HASH	该节区使用的符号表在节区表(Section Header Table)	0

yananhdeMacBook-Pro:armeabi-v7a yananh\$ readelf -S libtest.so  
There are 25 section headers, starting at offset 0x2284:

#### Section Headers:

[Nr]	Name	Type	Addr	Off	Size	ES	Flg	Lk	Inf	Al
[ 0]		NULL	00000000	000000	000000	00		0	0	0
[ 1]	.note.android.ide	NOTE	00000134	000134	000098	00	A	0	0	4
[ 2]	.note.gnu.build-i	NOTE	000001cc	0001cc	000024	00	A	0	0	4
[ 3]	.dynsym	DYNSYM	000001f0	0001f0	000100	10	A	4	1	4
[ 4]	.dynstr	STRTAB	000002f0	0002f0	0000ec	00	A	0	0	1
[ 5]	.hash	HASH	000003dc	0003dc	000054	04	A	3	0	4
[ 6]	.gnu.version	VERSYM	00000430	000430	000020	02	A	3	0	2
[ 7]	.gnu.version_d	VERDEF	00000450	000450	00001c	00	A	4	1	4
[ 8]	.gnu.version_r	VERNEED	0000046c	00046c	000020	00	A	4	1	4
[ 9]	.rel.dyn	REL	0000048c	00048c	000048	08	A	3	0	4
[10]	.rel.plt	REL	000004d4	0004d4	000050	08	AI	3	18	4
[11]	.plt	PROGBITS	00000524	000524	00008c	00	AX	0	0	4
[12]	.text	PROGBITS	000005b0	0005b0	0015a4	00	AX	0	0	4
[13]	.ARM.extab	PROGBITS	00001b54	001b54	00003c	00	A	0	0	4
[14]	.ARM.exidx	ARM_EXIDX	00001b90	001b90	000100	08	AL	12	0	4
[15]	.rodata	PROGBITS	00001c90	001c90	00000a	01	AMS	0	0	1
[16]	.fini_array	FINI_ARRAY	00002ea4	001ea4	000004	04	WA	0	0	4
[17]	.dynamic	DYNAMIC	00002ea8	001ea8	000108	08	WA	4	0	4
[18]	.got	PROGBITS	00002fb0	001fb0	000050	00	WA	0	0	4
[19]	.data	PROGBITS	00003000	002000	000004	00	WA	0	0	4
[20]	.bss	NOBITS	00003004	002004	000000	00	WA	0	0	1
[21]	.comment	PROGBITS	00000000	002004	00012f	01	MS	0	0	1
[22]	.note.gnu.gold-ve	NOTE	00000000	002134	00001c	00		0	0	4
[23]	.ARM.attributes	ARM_ATTRIBUTES	00000000	002150	000036	00		0	0	1
[24]	.shstrtab	STRTAB	00000000	002186	0000fe	00		0	0	1

#### Key to Flags:

W (write), A (alloc), X (execute), M (merge), S (strings), I (info),  
L (link order), O (extra OS processing required), G (group), T (TLS),  
C (compressed), x (unknown), o (OS specific), E (exclude),  
y (nored), p (processor specific)