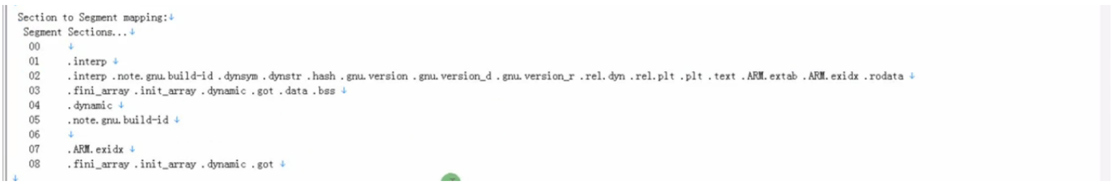
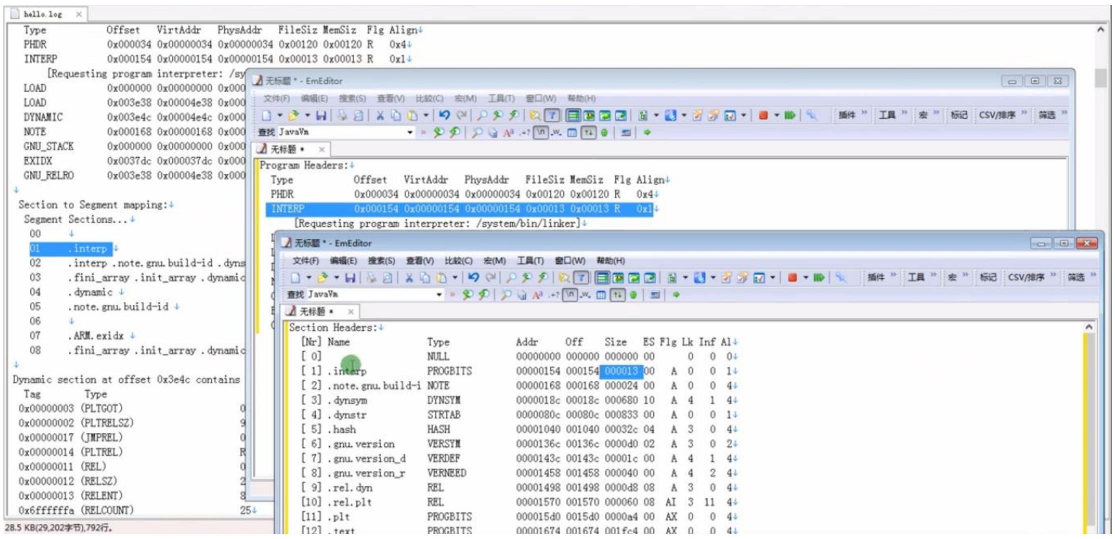
1. Section Header And Program Header

链接视图中用到 Section Header，执行视图中用到 Program Header。两者理论上都能指出 so 文件的所有信息，那么它们之间是否存在关联性



```
Section to Segment mapping:
 Segment Sections...
  00
  01     .interp
  02     .interp .note.gnu.build-id .dynsym .dynstr .hash .gnu.version .gnu.version_d .gnu.version_r .rel.dyn .rel.plt .plt .text .ARM.extab .ARM.exidx .rodata
  03     .fini_array .init_array .dynamic .got .data .bss
  04     .dynamic
  05     .note.gnu.build-id
  06
  07     .ARM.exidx
  08     .fini_array .init_array .dynamic .got
```

如上图，Segment 和 Section 的映射关系是一对一或者一对多。



```
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x00000034 0x00000034 0x00120 0x00120 R   0x4
  INTERP         0x000154 0x00000154 0x00000154 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /sy
  LOAD           0x000000 0x00000000 0x000
  LOAD           0x003e38 0x00004e38 0x000
  DYNAMIC        0x003e4c 0x00004e4c 0x000
  NOTE           0x000168 0x00000168 0x000
  GNU_STACK      0x000000 0x00000000 0x000
  EXIDX          0x0037dc 0x000037dc 0x000
  GNU_RELRO      0x003e38 0x00004e38 0x000

 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp
   02     .interp .note.gnu.build-id .dyns
   03     .fini_array .init_array .dynamic
   04     .dynamic
   05     .note.gnu.build-id
   06
   07     .ARM.exidx
   08     .fini_array .init_array .dynamic

Dynamic section at offset 0x3e4c contains
  Tag        Type
 0x00000003 (PLTGOT)                     0
 0x00000002 (PLTRELSZ)                   9
 0x00000017 (JMPREL)                     0
 0x00000014 (PLTREL)                     R
 0x00000011 (REL)                        0
 0x00000012 (RELSZ)                      2
 0x00000013 (RELENT)                     8
 0x6ffffffa (RELCOUNT)                   25
```

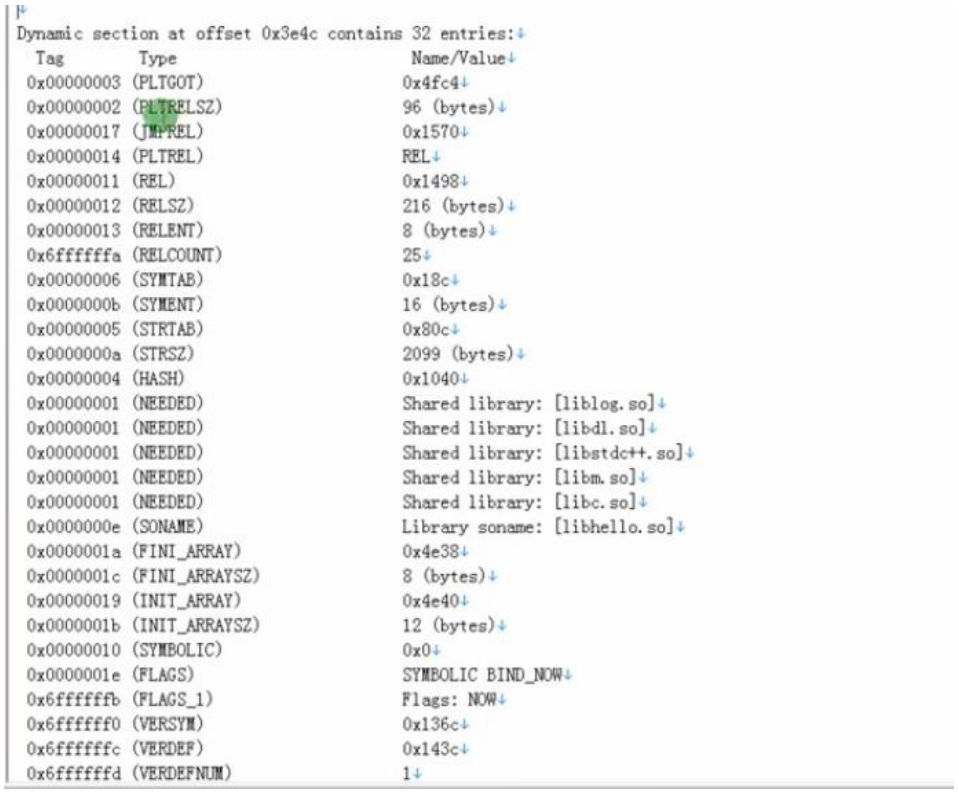
28.5 KB(29,202字节),792行,

```
Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x00000034 0x00000034 0x00120 0x00120 R   0x4
  INTERP         0x000154 0x00000154 0x00000154 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /system/bin/linker]
```

```
Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .interp           PROGBITS        00000154 000154 000013 00   A  0   0  1
  [ 2] .note.gnu.build-i NOTE            00000168 000168 000024 00   A  0   0  4
  [ 3] .dynsym           DYNSYM          0000018c 00018c 000680 10   A  4   1  4
  [ 4] .dynstr           STRTAB          0000080c 00080c 000833 00   A  0   0  1
  [ 5] .hash             HASH            00001040 001040 00032c 04   A  3   0  4
  [ 6] .gnu.version      VERSYM          0000136c 00136c 0000d0 02   A  3   0  2
  [ 7] .gnu.version_d    VERDEF          0000143c 00143c 00001c 00   A  4   1  4
  [ 8] .gnu.version_r    VERNEED         00001458 001458 000040 00   A  4   2  4
  [ 9] .rel.dyn          REL             00001498 001498 0000d8 08   A  3   0  4
  [10] .rel.plt          REL             00001570 001570 000060 08  AI  3  11  4
  [11] .plt              PROGBITS        000015d0 0015d0 0000a4 00  AX  0   0  4
  [12] .text             PROGBITS        00001674 001674 001fc4 00  AX  0   0  4
```

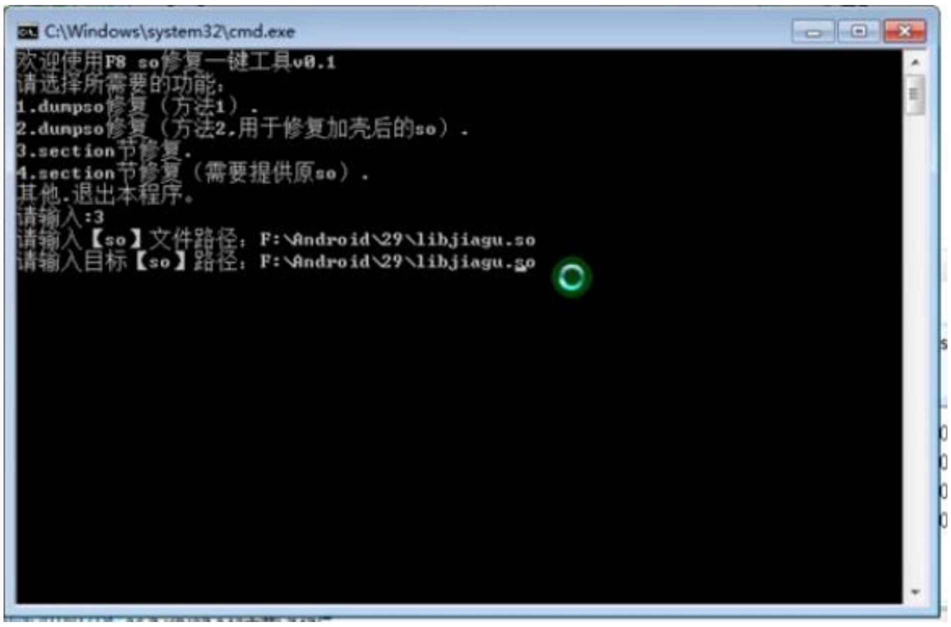
无标题 * - EmEditor
文件(F) 编辑(E) 搜索(S) 查看(V) 比较(C) 宏(M) 工具(T) 窗口(W) 帮助(H)
查找 JavaVm
无标题 •
hello.log

Dynamic Section(重要)



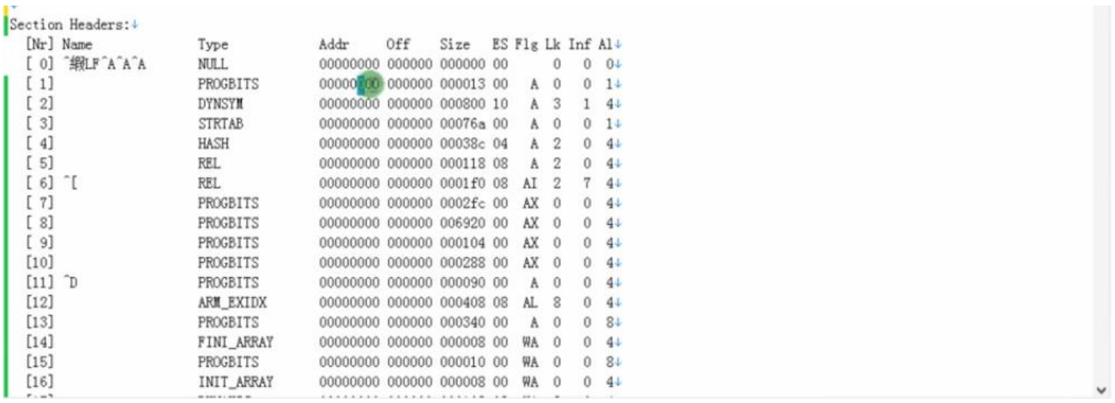
```
Dynamic section at offset 0x3e4c contains 32 entries:
  Tag        Type                         Name/Value
 0x00000003 (PLTGOT)                     0x4fc4
 0x00000002 (PLTRELSZ)                   96 (bytes)
 0x00000017 (JMPREL)                     0x1570
 0x00000014 (PLTREL)                     REL
 0x00000011 (REL)                        0x1498
 0x00000012 (RELSZ)                      216 (bytes)
 0x00000013 (RELENT)                     8 (bytes)
 0x6ffffffa (RELCOUNT)                   25
 0x00000006 (SYMTAB)                     0x18c
 0x0000000b (SYMENT)                     16 (bytes)
 0x00000005 (STRTAB)                     0x80c
 0x0000000a (STRSZ)                      2099 (bytes)
 0x00000004 (HASH)                       0x1040
 0x00000001 (NEEDED)                     Shared library: [liblog.so]
 0x00000001 (NEEDED)                     Shared library: [libdl.so]
 0x00000001 (NEEDED)                     Shared library: [libstdc++.so]
 0x00000001 (NEEDED)                     Shared library: [libm.so]
 0x00000001 (NEEDED)                     Shared library: [libc.so]
 0x0000000e (SONAME)                     Library soname: [libhello.so]
 0x0000001a (FINI_ARRAY)                 0x4e38
 0x0000001c (FINI_ARRAYSZ)               8 (bytes)
 0x00000019 (INIT_ARRAY)                 0x4e40
 0x0000001b (INIT_ARRAYSZ)               12 (bytes)
 0x00000010 (SYMBOLIC)                   0x0
 0x0000001e (FLAGS)                      SYMBOLIC BIND_NOW
 0x6ffffffb (FLAGS_1)                    Flags: NOW
 0x6ffffff0 (VERSYM)                     0x136c
 0x6ffffffc (VERDEF)                     0x143c
 0x6ffffffd (VERDEFNUM)                  1
```

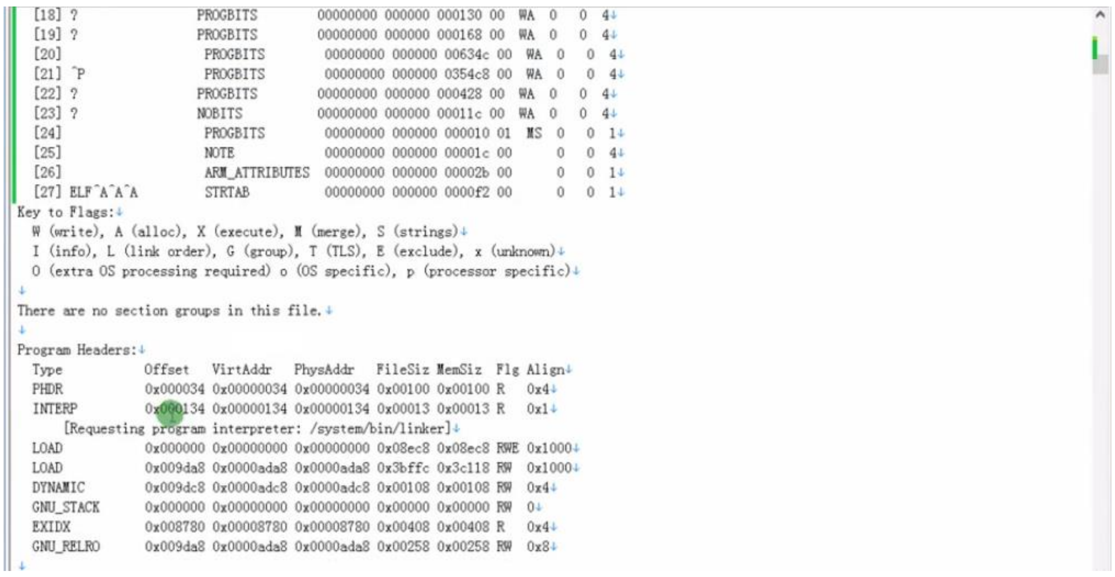
2. Section 修复（工具）



```
C:\Windows\system32\cmd.exe
欢迎使用F8 so修复一键工具v0.1
请选择所需要的功能:
1.dumpso修复（方法1）.
2.dumpso修复（方法2,用于修复加壳后的so）.
3.section节修复.
4.section节修复（需要提供原so）.
其他.退出本程序。
请输入:3
请输入【so】文件路径: F:\Android\29\libjiagu.so
请输入目标【so】路径: F:\Android\29\libjiagu.so
```

3. Section 修复（手动）



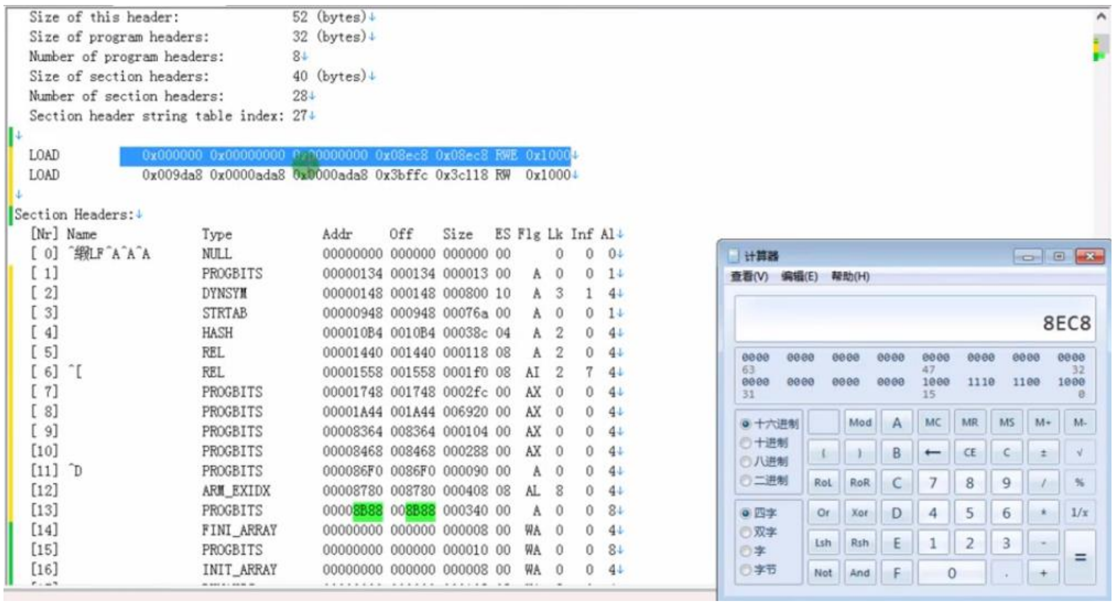


```
Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0] ^缃LF^A^A^A       NULL            00000000 000000 000000 00      0   0  0
  [ 1]                   PROGBITS        00000000 000000 000013 00   A  0   0  1
  [ 2]                   DYNSYM          00000000 000000 000800 10   A  3   1  4
  [ 3]                   STRTAB          00000000 000000 00076a 00   A  0   0  1
  [ 4]                   HASH            00000000 000000 00038c 04   A  2   0  4
  [ 5]                   REL             00000000 000000 000118 08   A  2   0  4
  [ 6] ^[                REL             00000000 000000 0001f0 08  AI  2   7  4
  [ 7]                   PROGBITS        00000000 000000 0002fc 00  AX  0   0  4
  [ 8]                   PROGBITS        00000000 000000 006920 00  AX  0   0  4
  [ 9]                   PROGBITS        00000000 000000 000104 00  AX  0   0  4
  [10]                   PROGBITS        00000000 000000 000288 00  AX  0   0  4
  [11] ^D                PROGBITS        00000000 000000 000090 00   A  0   0  4
  [12]                   ARM_EXIDX       00000000 000000 000408 08  AL  8   0  4
  [13]                   PROGBITS        00000000 000000 000340 00   A  0   0  8
  [14]                   FINI_ARRAY      00000000 000000 000008 00  WA  0   0  4
  [15]                   PROGBITS        00000000 000000 000010 00  WA  0   0  8
  [16]                   INIT_ARRAY      00000000 000000 000008 00  WA  0   0  4

   [18] ?                PROGBITS        00000000 000000 000130 00  WA  0   0  4
   [19] ?                PROGBITS        00000000 000000 000168 00  WA  0   0  4
   [20]                   PROGBITS        00000000 000000 00634c 00  WA  0   0  4
   [21] ^P                PROGBITS        00000000 000000 0354c8 00  WA  0   0  4
   [22] ?                PROGBITS        00000000 000000 000428 00  WA  0   0  4
   [23] ?                NOBITS          00000000 000000 00011c 00  WA  0   0  4
   [24]                   PROGBITS        00000000 000000 000010 01  MS  0   0  1
   [25]                   NOTE            00000000 000000 00001c 00      0   0  4
   [26]                   ARM_ATTRIBUTES  00000000 000000 00002b 00      0   0  1
   [27] ELF^A^A^A         STRTAB          00000000 000000 0000f2 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)

There are no section groups in this file.

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x00000034 0x00000034 0x00100 0x00100 R   0x4
  INTERP         0x000134 0x00000134 0x00000134 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /system/bin/linker]
  LOAD           0x000000 0x00000000 0x00000000 0x08ec8 0x08ec8 RWE 0x1000
  LOAD           0x009da8 0x0000ada8 0x0000ada8 0x3bffc 0x3c118 RW  0x1000
  DYNAMIC        0x009dc8 0x0000adc8 0x0000adc8 0x00108 0x00108 RW  0x4
  GNU_STACK      0x000000 0x00000000 0x00000000 0x00000 0x00000 RW  0
  EXIDX          0x008780 0x00008780 0x00008780 0x00408 0x00408 R   0x4
  GNU_RELRO      0x009da8 0x0000ada8 0x0000ada8 0x00258 0x00258 RW  0x8
```

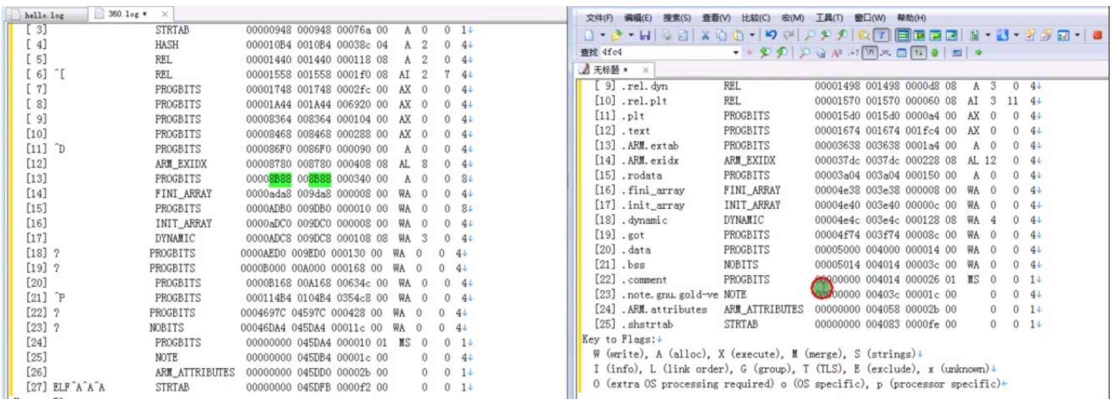


```
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         8
  Size of section headers:           40 (bytes)
  Number of section headers:         28
  Section header string table index: 27

  LOAD           0x000000 0x00000000 0x00000000 0x08ec8 0x08ec8 RWE 0x1000
  LOAD           0x009da8 0x0000ada8 0x0000ada8 0x3bffc 0x3c118 RW  0x1000

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0] ^缈LF^A^A^A       NULL            00000000 000000 000000 00      0   0  0
  [ 1]                   PROGBITS        00000134 000134 000013 00   A  0   0  1
  [ 2]                   DYNSYM          00000148 000148 000800 10   A  3   1  4
  [ 3]                   STRTAB          00000948 000948 00076a 00   A  0   0  1
  [ 4]                   HASH            000010B4 0010B4 00038c 04   A  2   0  4
  [ 5]                   REL             00001440 001440 000118 08   A  2   0  4
  [ 6] ^[                REL             00001558 001558 0001f0 08  AI  2   7  4
  [ 7]                   PROGBITS        00001748 001748 0002fc 00  AX  0   0  4
  [ 8]                   PROGBITS        00001A44 001A44 006920 00  AX  0   0  4
  [ 9]                   PROGBITS        00008364 008364 000104 00  AX  0   0  4
  [10]                   PROGBITS        00008468 008468 000288 00  AX  0   0  4
  [11] ^D                PROGBITS        000086F0 0086F0 000090 00   A  0   0  4
  [12]                   ARM_EXIDX       00008780 008780 000408 08  AL  8   0  4
  [13]                   PROGBITS        00008B88 008B88 000340 00   A  0   0  8
  [14]                   FINI_ARRAY      00000000 000000 000008 00  WA  0   0  4
  [15]                   PROGBITS        00000000 000000 000010 00  WA  0   0  8
  [16]                   INIT_ARRAY      00000000 000000 000008 00  WA  0   0  4
```

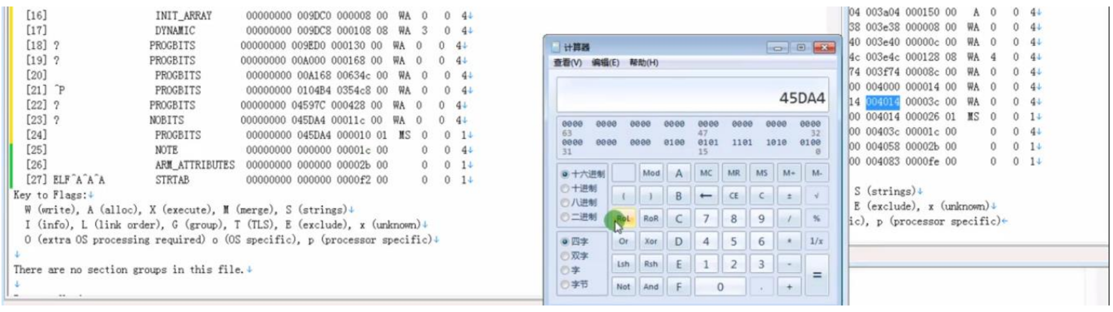
因为第一个 LOAD 在 8EC8 位置结束，所以接下来将从第二个 LOAD 的起始位置 ada8 开始而不是 8EC8


hello.log
360.log •
文件(F) 编辑(E) 搜索(S) 查看(V) 比较(C) 宏(M) 工具(T) 窗口(W) 帮助(H)
查找 4fc4
无标题 •

```
  [ 3]                   STRTAB          00000948 000948 00076a 00   A  0   0  1
  [ 4]                   HASH            000010B4 0010B4 00038c 04   A  2   0  4
  [ 5]                   REL             00001440 001440 000118 08   A  2   0  4
  [ 6] ^[                REL             00001558 001558 0001f0 08  AI  2   7  4
  [ 7]                   PROGBITS        00001748 001748 0002fc 00  AX  0   0  4
  [ 8]                   PROGBITS        00001A44 001A44 006920 00  AX  0   0  4
  [ 9]                   PROGBITS        00008364 008364 000104 00  AX  0   0  4
  [10]                   PROGBITS        00008468 008468 000288 00  AX  0   0  4
  [11] ^D                PROGBITS        000086F0 0086F0 000090 00   A  0   0  4
  [12]                   ARM_EXIDX       00008780 008780 000408 08  AL  8   0  4
  [13]                   PROGBITS        00008B88 008B88 000340 00   A  0   0  8
  [14]                   FINI_ARRAY      0000ada8 009da8 000008 00  WA  0   0  4
  [15]                   PROGBITS        0000ADB0 009DB0 000010 00  WA  0   0  8
  [16]                   INIT_ARRAY      0000aDC0 009DC0 000008 00  WA  0   0  4
  [17]                   DYNAMIC         0000ADC8 009DC8 000108 08  WA  3   0  4
  [18] ?                PROGBITS        0000AED0 009ED0 000130 00  WA  0   0  4
  [19] ?                PROGBITS        0000B000 00A000 000168 00  WA  0   0  4
  [20]                   PROGBITS        0000B168 00A168 00634c 00  WA  0   0  4
  [21] ^P                PROGBITS        000114B4 0104B4 0354c8 00  WA  0   0  4
  [22] ?                PROGBITS        0004697C 04597C 000428 00  WA  0   0  4
  [23] ?                NOBITS          00046DA4 045DA4 00011c 00  WA  0   0  4
  [24]                   PROGBITS        00000000 045DA4 000010 01  MS  0   0  1
  [25]                   NOTE            00000000 045DB4 00001c 00      0   0  4
  [26]                   ARM_ATTRIBUTES  00000000 045DD0 00002b 00      0   0  1
  [27] ELF^A^A^A         STRTAB          00000000 045DFB 0000f2 00      0   0  1
```

```
  [ 9] .rel.dyn          REL             00001498 001498 0000d8 08   A  3   0  4
  [10] .rel.plt          REL             00001570 001570 000060 08  AI  3  11  4
  [11] .plt              PROGBITS        000015d0 0015d0 0000a4 00  AX  0   0  4
  [12] .text             PROGBITS        00001674 001674 001fc4 00  AX  0   0  4
  [13] .ARM.extab        PROGBITS        00003638 003638 0001a4 00   A  0   0  4
  [14] .ARM.exidx        ARM_EXIDX       000037dc 0037dc 000228 08  AL 12   0  4
  [15] .rodata           PROGBITS        00003a04 003a04 000150 00   A  0   0  4
  [16] .fini_array       FINI_ARRAY      00004e38 003e38 000008 00  WA  0   0  4
  [17] .init_array       INIT_ARRAY      00004e40 003e40 00000c 00  WA  0   0  4
  [18] .dynamic          DYNAMIC         00004e4c 003e4c 000128 08  WA  4   0  4
  [19] .got              PROGBITS        00004f74 003f74 00008c 00  WA  0   0  4
  [20] .data             PROGBITS        00005000 004000 000014 00  WA  0   0  4
  [21] .bss              NOBITS          00005014 004014 00003c 00  WA  0   0  4
  [22] .comment          PROGBITS        00000000 004014 000026 01  MS  0   0  1
  [23] .note.gnu.gold-ve NOTE            00000000 00403c 00001c 00      0   0  4
  [24] .ARM.attributes   ARM_ATTRIBUTES  00000000 004058 00002b 00      0   0  1
  [25] .shstrtab         STRTAB          00000000 004083 0000fe 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)
```

MS 类型的 offset 和上一个相同



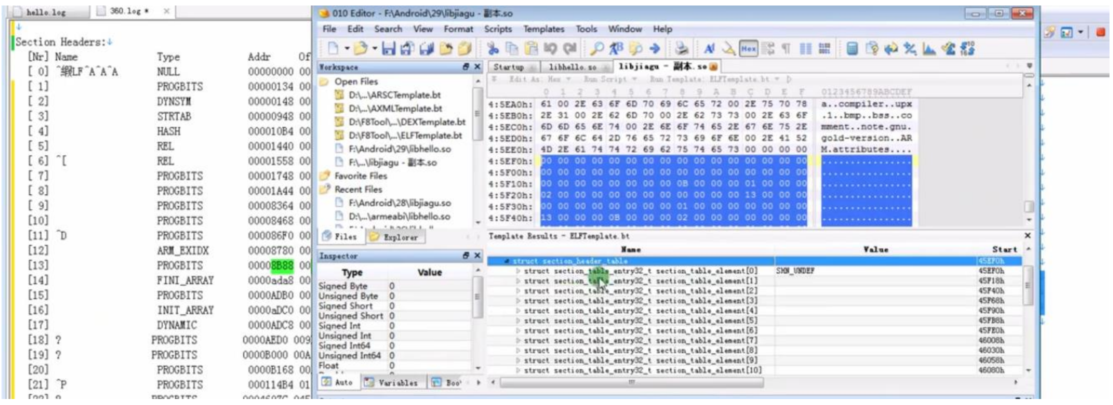
```
  [16]                   INIT_ARRAY      00000000 009DC0 000008 00  WA  0   0  4
  [17]                   DYNAMIC         00000000 009DC8 000108 08  WA  3   0  4
  [18] ?                PROGBITS        00000000 009ED0 000130 00  WA  0   0  4
  [19] ?                PROGBITS        00000000 00A000 000168 00  WA  0   0  4
  [20]                   PROGBITS        00000000 00A168 00634c 00  WA  0   0  4
  [21] ^P                PROGBITS        00000000 0104B4 0354c8 00  WA  0   0  4
  [22] ?                PROGBITS        00000000 04597C 000428 00  WA  0   0  4
  [23] ?                NOBITS          00000000 045DA4 00011c 00  WA  0   0  4
  [24]                   PROGBITS        00000000 045DA4 000010 01  MS  0   0  1
  [25]                   NOTE            00000000 000000 00001c 00      0   0  4
  [26]                   ARM_ATTRIBUTES  00000000 000000 00002b 00      0   0  1
  [27] ELF^A^A^A         STRTAB          00000000 000000 0000f2 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)

There are no section groups in this file.
```

```
04 003a04 000150 00   A  0   0  4
38 003e38 000008 00  WA  0   0  4
40 003e40 00000c 00  WA  0   0  4
4c 003e4c 000128 08  WA  4   0  4
74 003f74 00008c 00  WA  0   0  4
00 004000 000014 00  WA  0   0  4
14 004014 00003c 00  WA  0   0  4
00 004014 000026 01  MS  0   0  1
00 00403c 00001c 00      0   0  4
00 004058 00002b 00      0   0  1
00 004083 0000fe 00      0   0  1

 S (strings)
 E (exclude), x (unknown)
ic), p (processor specific)
```

将对应的 section 数据修改回去后 section 段能够正常识别

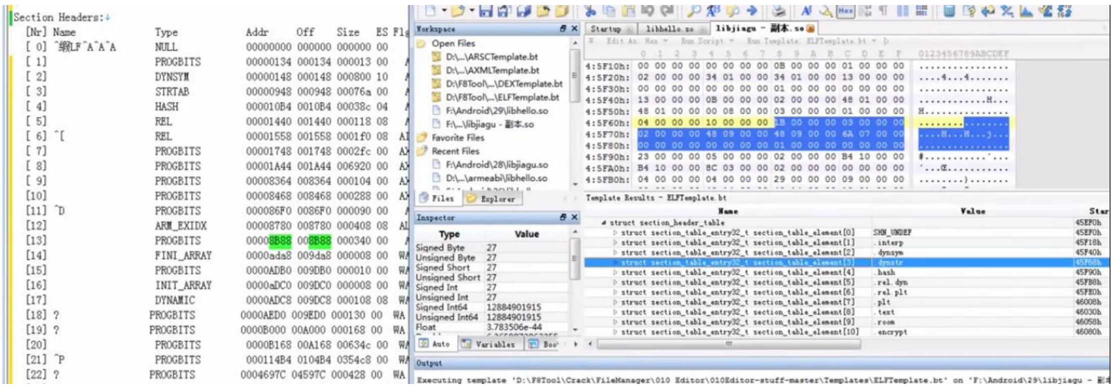


```
Section Headers:
  [Nr] Name              Type            Addr     Of
  [ 0] ^缀LF^A^A^A       NULL            00000000 00
  [ 1]                   PROGBITS        00000134 00
  [ 2]                   DYNSYM          00000148 00
  [ 3]                   STRTAB          00000948 00
  [ 4]                   HASH            000010B4 00
  [ 5]                   REL             00001440 00
  [ 6] ^[                REL             00001558 00
  [ 7]                   PROGBITS        00001748 00
  [ 8]                   PROGBITS        00001A44 00
  [ 9]                   PROGBITS        00008364 00
  [10]                   PROGBITS        00008468 00
  [11] ^D                PROGBITS        000086F0 00
  [12]                   ARM_EXIDX       00008780 00
  [13]                   PROGBITS        00008B88 00
  [14]                   FINI_ARRAY      0000ada8 00
  [15]                   PROGBITS        0000ADB0 00
  [16]                   INIT_ARRAY      0000aDC0 00
  [17]                   DYNAMIC         0000ADC8 00
  [18] ?                PROGBITS        0000AED0 009
  [19] ?                PROGBITS        0000B000 00A
  [20]                   PROGBITS        0000B168 00
  [21] ^P                PROGBITS        000114B4 01
```

```
010 Editor - F:\Android\29\libjiagu - 副本.so

4:5EA0h: 61 00 2E 63 6F 6D 70 69 6C 65 72 00 2E 75 70 78  a..compiler..upx
4:5EB0h: 2E 31 00 2E 62 6D 70 00 2E 62 73 73 00 2E 63 6F  .1..bmp..bss..co
4:5EC0h: 6D 6D 65 6E 74 00 2E 6E 6F 74 65 2E 67 6E 75 2E  mment..note.gnu.
4:5ED0h: 67 6F 6C 64 2D 76 65 72 73 69 6F 6E 00 2E 41 52  gold-version..AR
4:5EE0h: 4D 2E 61 74 74 72 69 62 75 74 65 73 00 00 00 00  M.attributes....
```

```
Template Results - ELFTemplate.bt
Name                                                       Value       Start
struct section_header_table                                            45EF0h
  struct section_table_entry32_t section_table_element[0]  SHN_UNDEF   45EF0h
  struct section_table_entry32_t section_table_element[1]              45F18h
  struct section_table_entry32_t section_table_element[2]              45F40h
  struct section_table_entry32_t section_table_element[3]              45F68h
  struct section_table_entry32_t section_table_element[4]              45F90h
  struct section_table_entry32_t section_table_element[5]              45FB8h
  struct section_table_entry32_t section_table_element[6]              45FE0h
  struct section_table_entry32_t section_table_element[7]              46008h
  struct section_table_entry32_t section_table_element[8]              46030h
  struct section_table_entry32_t section_table_element[9]              46058h
  struct section_table_entry32_t section_table_element[10]             46080h
```

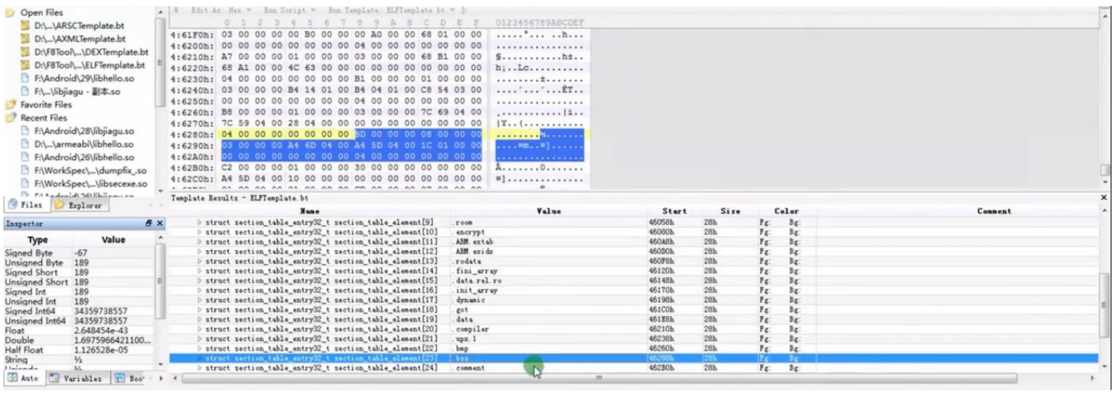


```
Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg
  [ 0] ^缀LF^A^A^A       NULL            00000000 000000 000000 00
  [ 1]                   PROGBITS        00000134 000134 000013 00   A
  [ 2]                   DYNSYM          00000148 000148 000800 10   A
  [ 3]                   STRTAB          00000948 000948 00076a 00   A
  [ 4]                   HASH            000010B4 0010B4 00038c 04   A
  [ 5]                   REL             00001440 001440 000118 08   A
  [ 6] ^[                REL             00001558 001558 0001f0 08  AI
  [ 7]                   PROGBITS        00001748 001748 0002fc 00  AX
  [ 8]                   PROGBITS        00001A44 001A44 006920 00  AX
  [ 9]                   PROGBITS        00008364 008364 000104 00  AX
  [10]                   PROGBITS        00008468 008468 000288 00  AX
  [11] ^D                PROGBITS        000086F0 0086F0 000090 00   A
  [12]                   ARM_EXIDX       00008780 008780 000408 08  AL
  [13]                   PROGBITS        00008B88 008B88 000340 00   A
  [14]                   FINI_ARRAY      0000ada8 009da8 000008 00  WA
  [15]                   PROGBITS        0000ADB0 009DB0 000010 00  WA
  [16]                   INIT_ARRAY      0000aDC0 009DC0 000008 00  WA
  [17]                   DYNAMIC         0000ADC8 009DC8 000108 08  WA
  [18] ?                PROGBITS        0000AED0 009ED0 000130 00  WA
  [19] ?                PROGBITS        0000B000 00A000 000168 00  WA
  [20]                   PROGBITS        0000B168 00A168 00634c 00  WA
  [21] ^P                PROGBITS        000114B4 0104B4 0354c8 00  WA
  [22] ?                PROGBITS        0004697C 04597C 000428 00  WA
```

```
4:5F40h: 13 00 00 00 0B 00 00 00 02 00 00 00 48 01 00 00  ............H...
4:5F50h: 48 01 00 00 00 08 00 00 03 00 00 00 01 00 00 00  H...............
4:5F60h: 04 00 00 00 10 00 00 00 1B 00 00 00 03 00 00 00  ................
4:5F70h: 02 00 00 00 48 09 00 00 48 09 00 00 6A 07 00 00  ....H...H...j...
4:5F80h: 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
4:5F90h: 23 00 00 00 05 00 00 00 02 00 00 00 B4 10 00 00  #...............
4:5FA0h: B4 10 00 00 8C 03 00 00 02 00 00 00 00 00 00 00  ....Œ...........
4:5FB0h: 04 00 00 00 04 00 00 00 29 00 00 00 09 00 00 00  ........).......
```

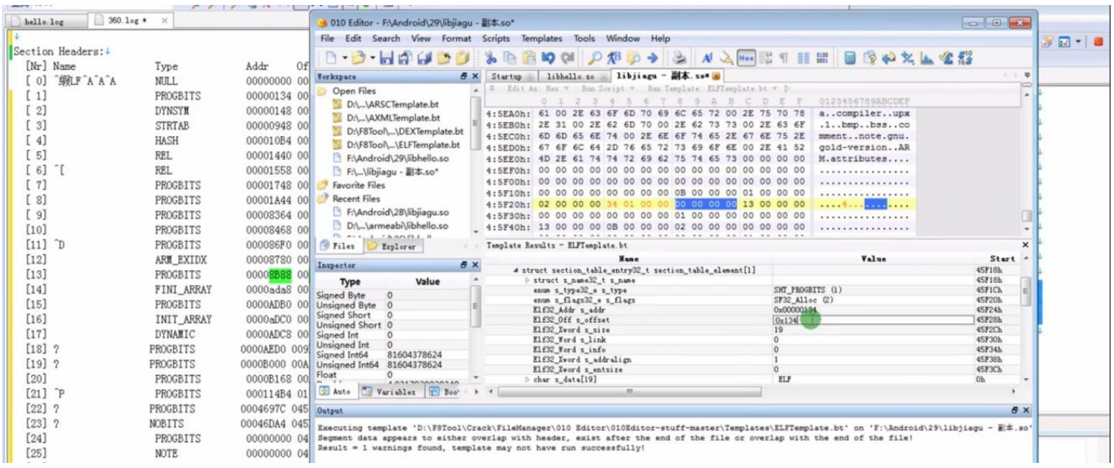
```
Template Results - ELFTemplate.bt
Name                                                       Value       Start
struct section_header_table                                            45EF0h
  struct section_table_entry32_t section_table_element[0]  SHN_UNDEF   45EF0h
  struct section_table_entry32_t section_table_element[1]  .interp     45F18h
  struct section_table_entry32_t section_table_element[2]  .dynsym     45F40h
  struct section_table_entry32_t section_table_element[3]  .dynstr     45F68h
  struct section_table_entry32_t section_table_element[4]  .hash       45F90h
  struct section_table_entry32_t section_table_element[5]  .rel.dyn    45FB8h
  struct section_table_entry32_t section_table_element[6]  .rel.plt    45FE0h
  struct section_table_entry32_t section_table_element[7]  .plt        46008h
  struct section_table_entry32_t section_table_element[8]  .text       46030h
  struct section_table_entry32_t section_table_element[9]  .room       46058h
  struct section_table_entry32_t section_table_element[10] .encrypt    46080h

Output
Executing template 'D:\F8Tool\Crack\FileManager\010 Editor\010Editor-stuff-master\Templates\ELFTemplate.bt' on 'F:\Android\29\libjiagu - 副本
```



```
4:61F0h: 03 00 00 00 00 B0 00 00 00 A0 00 00 68 01 00 00
4:6200h: 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00
4:6210h: A7 00 00 00 01 00 00 00 03 00 00 00 68 B1 00 00
4:6220h: 68 A1 00 00 4C 63 00 00 00 00 00 00 00 00 00 00
4:6230h: 04 00 00 00 00 00 00 00 B1 00 00 00 01 00 00 00
4:6240h: 03 00 00 00 B4 14 01 00 B4 04 01 00 C8 54 03 00  ....'...'...ÈT..
4:6250h: 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  ................
4:6260h: B8 00 00 00 01 00 00 00 03 00 00 00 7C 69 04 00  ,...........|i..
4:6270h: 7C 59 04 00 28 04 00 00 00 00 00 00 00 00 00 00  |Y..(...........
4:6280h: 04 00 00 00 00 00 00 00 BD 00 00 00 08 00 00 00  ........½.......
4:6290h: 03 00 00 00 A4 6D 04 00 A4 5D 04 00 1C 01 00 00  ....¤m..¤]......
4:62A0h: 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  ................
4:62B0h: C2 00 00 00 01 00 00 00 30 00 00 00 00 00 00 00  Â.......0.......
4:62C0h: A4 5D 04 00 10 00 00 00 00 00 00 00 00 00 00 00  ¤]..............
```

```
Template Results - ELFTemplate.bt
Name                                                       Value         Start   Size  Color   Comment
  struct section_table_entry32_t section_table_element[9]  .room         46058h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[10] .encrypt      46080h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[11] .ARM.extab    460A8h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[12] .ARM.exidx    460D0h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[13] .rodata       460F8h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[14] .fini_array   46120h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[15] .data.rel.ro  46148h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[16] .init_array   46170h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[17] .dynamic      46198h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[18] .got          461C0h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[19] .data         461E8h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[20] .compiler     46210h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[21] .upx.1        46238h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[22] .bmp          46260h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[23] .bss          46288h  28h   Fg: Bg:
  struct section_table_entry32_t section_table_element[24] .comment      462B0h  28h   Fg: Bg:
```



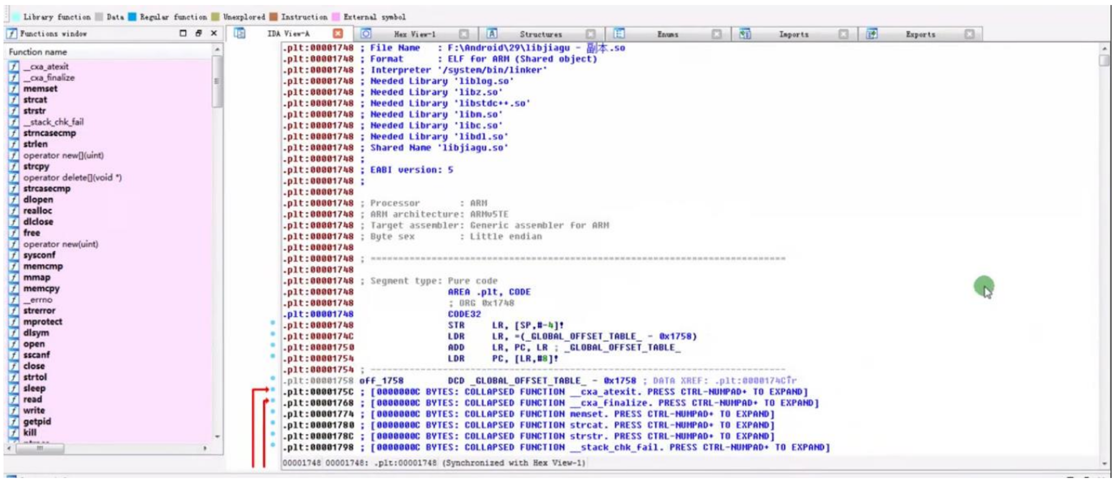
```
010 Editor - F:\Android\29\libjiagu - 副本.so*

4:5EC0h: 6D 6D 65 6E 74 00 2E 6E 6F 74 65 2E 67 6E 75 2E  mment..note.gnu.
4:5ED0h: 67 6F 6C 64 2D 76 65 72 73 69 6F 6E 00 2E 41 52  gold-version..AR
4:5EE0h: 4D 2E 61 74 74 72 69 62 75 74 65 73 00 00 00 00  M.attributes....
4:5EF0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
4:5F00h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
4:5F10h: 00 00 00 00 00 00 00 00 0B 00 00 00 01 00 00 00  ................
4:5F20h: 02 00 00 00 34 01 00 00 00 00 00 00 13 00 00 00  ....4...........
4:5F30h: 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
4:5F40h: 13 00 00 00 0B 00 00 00 02 00 00 00 00 00 00 00  ................
```

```
Template Results - ELFTemplate.bt
Name                                                      Value              Start
struct section_table_entry32_t section_table_element[1]                      45F18h
  struct s_name32_t s_name                                                   45F18h
  enum s_type32_e s_type                                  SHT_PROGBITS (1)   45F1Ch
  enum s_flags32_e s_flags                                SF32_Alloc (2)     45F20h
  Elf32_Addr s_addr                                       0x00000134         45F24h
  Elf32_Off s_offset                                      0x134              45F28h
  Elf32_Xword s_size                                      19                 45F2Ch
  Elf32_Word s_link                                       0                  45F30h
  Elf32_Word s_info                                       0                  45F34h
  Elf32_Xword s_addralign                                 1                  45F38h
  Elf32_Xword s_entsize                                   0                  45F3Ch
  char s_data[19]                                         ELF                0h

Output
Executing template 'D:\F8Tool\Crack\FileManager\010 Editor\010Editor-stuff-master\Templates\ELFTemplate.bt' on 'F:\Android\29\libjiagu - 副本.so'
Segment data appears to either overlap with header, exist after the end of the file or overlap with the end of the file!
Result = 1 warnings found, template may not have run successfully!
```

ida 打开能够读取成功



```
__cxa_atexit
__cxa_finalize
memset
strcat
strstr
__stack_chk_fail
strncasecmp
strlen
operator new[](uint)
strcpy
operator delete[](void *)
strcasecmp
dlopen
realloc
dlclose
free
operator new(uint)
sysconf
memcmp
mmap
memcpy
__errno
strerror
mprotect
dlsym
open
sscanf
close
strtol
sleep
read
write
getpid
kill
```

```
.plt:00001748 ; File Name   : F:\Android\29\libjiagu - 副本.so
.plt:00001748 ; Format      : ELF for ARM (Shared object)
.plt:00001748 ; Interpreter '/system/bin/linker'
.plt:00001748 ; Needed Library 'liblog.so'
.plt:00001748 ; Needed Library 'libz.so'
.plt:00001748 ; Needed Library 'libstdc++.so'
.plt:00001748 ; Needed Library 'libm.so'
.plt:00001748 ; Needed Library 'libc.so'
.plt:00001748 ; Needed Library 'libdl.so'
.plt:00001748 ; Shared Name 'libjiagu.so'
.plt:00001748 ;
.plt:00001748 ; EABI version: 5
.plt:00001748 ;
.plt:00001748
.plt:00001748 ; Processor        : ARM
.plt:00001748 ; ARM architecture: ARMv5TE
.plt:00001748 ; Target assembler: Generic assembler for ARM
.plt:00001748 ; Byte sex         : Little endian
.plt:00001748
.plt:00001748 ; ===========================================================================
.plt:00001748
.plt:00001748 ; Segment type: Pure code
.plt:00001748                 AREA .plt, CODE
.plt:00001748                 ; ORG 0x1748
.plt:00001748                 CODE32
.plt:00001748                 STR     LR, [SP,#-4]!
.plt:0000174C                 LDR     LR, =(_GLOBAL_OFFSET_TABLE_ - 0x1758)
.plt:00001750                 ADD     LR, PC, LR ; _GLOBAL_OFFSET_TABLE_
.plt:00001754                 LDR     PC, [LR,#8]!
.plt:00001754 ; ---------------------------------------------------------------------------
.plt:00001758 off_1758        DCD _GLOBAL_OFFSET_TABLE_ - 0x1758 ; DATA XREF: .plt:0000174C↑r
.plt:0000175C ; [0000000C BYTES: COLLAPSED FUNCTION __cxa_atexit. PRESS CTRL-NUMPAD+ TO EXPAND]
.plt:00001768 ; [0000000C BYTES: COLLAPSED FUNCTION __cxa_finalize. PRESS CTRL-NUMPAD+ TO EXPAND]
.plt:00001774 ; [0000000C BYTES: COLLAPSED FUNCTION memset. PRESS CTRL-NUMPAD+ TO EXPAND]
.plt:00001780 ; [0000000C BYTES: COLLAPSED FUNCTION strcat. PRESS CTRL-NUMPAD+ TO EXPAND]
.plt:0000178C ; [0000000C BYTES: COLLAPSED FUNCTION strstr. PRESS CTRL-NUMPAD+ TO EXPAND]
.plt:00001798 ; [0000000C BYTES: COLLAPSED FUNCTION __stack_chk_fail. PRESS CTRL-NUMPAD+ TO EXPAND]
00001748 00001748: .plt:00001748 (Synchronized with Hex View-1)
```