1. readelf

1.1 elf的文件头：readelf -h

readelf -h libtest.so

```
[yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -h libtest.so
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x0
  Start of program headers:          52 (bytes into file)
  Start of section headers:          8836 (bytes into file)
  Flags:                             0x5000200, Version5 EABI, soft-float ABI
  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:         25
  Section header string table index: 24 _
```

1.2 查询重定向库文件中所有的section：readelf -S

readelf -S libtest.so

```
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 (noread), p (processor specific)
```

1.3 查询某一单项的section：readelf -p

readelf -p ${number} libtest.so

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

String dump of section '.shstrtab':
  [     1]  .shstrtab
  [     b]  .note.android.ident
  [    1f]  .note.gnu.build-id
  [    32]  .dynsym
  [    3a]  .dynstr
  [    42]  .hash
  [    48]  .gnu.version
  [    55]  .gnu.version_d
  [    64]  .gnu.version_r
  [    73]  .rel.dyn
  [    7c]  .rel.plt
  [    85]  .text
  [    8b]  .ARM.extab
  [    96]  .ARM.exidx
  [    a1]  .rodata
  [    a9]  .fini_array
  [    b5]  .dynamic
  [    be]  .got
  [    c3]  .data
  [    c9]  .bss
  [    ce]  .comment
  [    d7]  .note.gnu.gold-version
  [    ee]  .ARM.attributes
```

1.4 查询库中的符号表：readelf -s

readelf -s libtest.so

```
[yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf  -s libtest.so

Symbol table '.dynsym' contains 16 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 00000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_finalize@LIBC (2)
     2: 00000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@LIBC (2)
     3: 00000000     0 FUNC    GLOBAL DEFAULT  UND malloc@LIBC (2)
     4: 00000000     0 FUNC    GLOBAL DEFAULT  UND printf@LIBC (2)
     5: 000005ed    60 FUNC    GLOBAL DEFAULT   12 say_hello
     6: 00000000     0 FUNC    GLOBAL DEFAULT  UND snprintf@LIBC (2)
     7: 00000000     0 FUNC    WEAK   DEFAULT  UND __gnu_Unwind_Find_exidx
     8: 00000000     0 FUNC    GLOBAL DEFAULT  UND abort@LIBC (2)
     9: 00000000     0 FUNC    GLOBAL DEFAULT  UND memcpy@LIBC (2)
    10: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __cxa_begin_cleanup
    11: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __cxa_type_match
    12: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __cxa_call_unexpected
    13: 00003004     0 NOTYPE  GLOBAL DEFAULT  ABS _edata
    14: 00003004     0 NOTYPE  GLOBAL DEFAULT  ABS __bss_start
    15: 00003004     0 NOTYPE  GLOBAL DEFAULT  ABS _end
```

1.5 可执行文件的程序头表：readelf -l

```
yananhdeMacBook-Pro:armeabi-v7a yananh$ readelf -l libcronet.72.0.3626.0.so

Elf file type is DYN (Shared object file)
Entry point 0x16d000
There are 9 program headers, starting at offset 52

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x00000034 0x00000034 0x00120 0x00120 R   0x4
  LOAD           0x000000 0x00000000 0x00000000 0x16cde0 0x16cde0 R   0x1000
  LOAD           0x16d000 0x0016d000 0x0016d000 0x37abd0 0x37abd0 R E 0x1000
  LOAD           0x4e8000 0x004e8000 0x004e8000 0x1f5bc 0x26d24 RW  0x1000
  DYNAMIC        0x5062c0 0x005062c0 0x005062c0 0x000f0 0x000f0 RW  0x4
  GNU_RELRO      0x4ea000 0x004ea000 0x004ea000 0x1d5bc 0x1e000 R   0x1
  GNU_STACK      0x000000 0x00000000 0x00000000 0x00000 0x00000 RW  0
  NOTE           0x06898c 0x0006898c 0x0006898c 0x00098 0x00098 R   0x4
  EXIDX          0x000154 0x00000154 0x00000154 0x390d0 0x390d0 R   0x4

 Section to Segment mapping:
  Segment Sections...
   00
   01     .ARM.exidx .dynsym .gnu.version .gnu.version_r .gnu.hash .hash .dynstr .rel.dyn .rel.plt .note.android.ident .ARM.extab .rod
ata
   02     .text .plt
   03     .data .data.rel.ro .init_array .fini_array .dynamic .got .got.plt .bss
   04     .dynamic
   05     .data.rel.ro .init_array .fini_array .dynamic .got .got.plt
   06
   07     .note.android.ident
   08     .ARM.exidx
```

1.6 重定位文件表：readelf -r

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

Relocation section '.rel.dyn' at offset 0x48c contains 9 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00002ea4  00000017 R_ARM_RELATIVE
00002fb0  00000017 R_ARM_RELATIVE
00002fb4  00000017 R_ARM_RELATIVE
00002fb8  00000017 R_ARM_RELATIVE
00002fc0  00000017 R_ARM_RELATIVE
00002fc4  00000017 R_ARM_RELATIVE
00003000  00000017 R_ARM_RELATIVE
00002fbc  00000715 R_ARM_GLOB_DAT    00000000   __gnu_Unwind_Find_exid
00002fc8  00000c15 R_ARM_GLOB_DAT    00000000   __cxa_call_unexpected

Relocation section '.rel.plt' at offset 0x4d4 contains 10 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
00002fd8  00000216 R_ARM_JUMP_SLOT   00000000   __cxa_atexit@LIBC
00002fdc  00000116 R_ARM_JUMP_SLOT   00000000   __cxa_finalize@LIBC
00002fe0  00000316 R_ARM_JUMP_SLOT   00000000   malloc@LIBC
00002fe4  00000616 R_ARM_JUMP_SLOT   00000000   snprintf@LIBC
00002fe8  00000416 R_ARM_JUMP_SLOT   00000000   printf@LIBC
00002fec  00000716 R_ARM_JUMP_SLOT   00000000   __gnu_Unwind_Find_exid
00002ff0  00000816 R_ARM_JUMP_SLOT   00000000   abort@LIBC
00002ff4  00000916 R_ARM_JUMP_SLOT   00000000   memcpy@LIBC
00002ff8  00000a16 R_ARM_JUMP_SLOT   00000000   __cxa_begin_cleanup
00002ffc  00000b16 R_ARM_JUMP_SLOT   00000000   __cxa_type_match
```

2 file

通过file命令能够获取文件类型

```
yananhdeMacBook-Pro:armeabi-v7a yananh$ file libtest.so
libtest.so: ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically linked, BuildID[sha1]=9363ce0db74cb9979d09f
e651a98f78581c8b966, stripped
```

3 strip

通过strip命令删除冗余section

比如***strip main***