readelf -h addr_space.o

```
ELF 头：
  Magic：   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  类别:                              ELF64
  数据:                              2 补码，小端序 (little endian)
  版本:                              1 (current)
  OS/ABI:                            UNIX - System V
  ABI 版本:                          0
  类型:                              REL (可重定位文件)
  系统架构:                          Advanced Micro Devices X86-64
  版本:                              0x1
  入口点地址：               0x0
  程序头起点：               0 (bytes into file)
  Start of section headers:          1224 (bytes into file)
  标志：                     0x0
  本头的大小：               64 (字节)
  程序头大小：               0 (字节)
  Number of program headers:         0
  节头大小：                 64 (字节)
  节头数量：                 14
  字符串表索引节头：         11
```

readelf -S -W addr_space.o

共有 14 个节头，从偏移量 0x4c8 开始：

```
节头：
  [Nr] Name              Type            Address          Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            0000000000000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        0000000000000000 000040 000082 00  AX  0   0  1
  [ 2] .rela.text        RELA            0000000000000000 000388 0000a8 18   I 12   1  8
  [ 3] .data             PROGBITS        0000000000000000 0000d0 000020 00  WA  0   0 16
  [ 4] .rela.data        RELA            0000000000000000 000430 000018 18   I 12   3  8
  [ 5] .bss              NOBITS          0000000000000000 0000f0 000000 00  WA  0   0  1
  [ 6] .rodata           PROGBITS        0000000000000000 0000f0 00003e 00   A  0   0  8
  [ 7] .comment          PROGBITS        0000000000000000 00012e 000035 01  MS  0   0  1
  [ 8] .note.GNU-stack   PROGBITS        0000000000000000 000163 000000 00      0   0  1
  [ 9] .eh_frame         PROGBITS        0000000000000000 000168 000038 00   A  0   0  8
  [10] .rela.eh_frame    RELA            0000000000000000 000448 000018 18   I 12   9  8
  [11] .shstrtab         STRTAB          0000000000000000 000460 000066 00      0   0  1
  [12] .symtab           SYMTAB          0000000000000000 0001a0 000198 18     13   9  8
  [13] .strtab           STRTAB          0000000000000000 000338 00004a 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings), l (large)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)
```

objdump -h addr_space.o

addr_space.o： 文件格式 elf64-x86-64

```
节:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .text         00000082  0000000000000000  0000000000000000  00000040  2**0
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000020  0000000000000000  0000000000000000  000000d0  2**4
                  CONTENTS, ALLOC, LOAD, RELOC, DATA
  2 .bss          00000000  0000000000000000  0000000000000000  000000f0  2**0
                  ALLOC
  3 .rodata       0000003e  0000000000000000  0000000000000000  000000f0  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .comment      00000035  0000000000000000  0000000000000000  0000012e  2**0
                  CONTENTS, READONLY
  5 .note.GNU-stack 00000000  0000000000000000  0000000000000000  00000163  2**0
                  CONTENTS, READONLY
  6 .eh_frame     00000038  0000000000000000  0000000000000000  00000168  2**3
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, DATA
```

几个变量分别为 char *myname；char gdata[128]; char bdata[16]； char * ldata[16]; char * ddata;

我们比较的 Section 是.text（存放代码）、.data（存放全局静态变量和局部静态变量）和.bss（存未初始化的全局变量和局部静态变量）

于是有

myname .data

gdata .data

bdata .data

但是实际测试中发现，gdata 被分在了.common 节内

```
SYMBOL TABLE:
0000000000000000 l    df *ABS*  0000000000000000 addr_space.c
0000000000000000 l    d  .text  0000000000000000 .text
0000000000000000 l    d  .data  0000000000000000 .data
0000000000000000 l    d  .bss   0000000000000000 .bss
0000000000000000 l    d  .rodata        0000000000000000 .rodata
0000000000000000 l    d  .note.GNU-stack        0000000000000000 .note.GNU-stack
0000000000000000 l    d  .eh_frame      0000000000000000 .eh_frame
0000000000000000 l    d  .comment       0000000000000000 .comment
0000000000000000 g     O .data  0000000000000008 myname
0000000000000080       O *COM*  0000000000000020 gdata
0000000000000010 g     O .data  0000000000000010 bdata
0000000000000000 g     F .text  0000000000000082 main
0000000000000000         *UND*  0000000000000000 malloc
0000000000000000         *UND*  0000000000000000 printf
0000000000000000         *UND*  0000000000000000 free
0000000000000000         *UND*  0000000000000000   stack chk fail
```