

readelf和objdump

1.readelf 只能看elf文件的信息

- 选项-h(elfheader), 显示elf文件开始的文件头信息。后面文章会补上具体说明。
- 选项-l(programheaders), segments显示程序头(段头)信息(如果有数据的话)。后面文章会补上具体说明。
- 选项-S(sectionheaders), sections显示节头信息(如果有数据的话)。后面文章会补上具体说明。
- 选项-a, all显示全部信息, 等价于-h-l-S-s-r-d-V-A-I。
- 选项-g(sectiongroups), 显示节组信息(如果有数据的话)。
- 选项-t, section-details显示节的详细信息(-S的)。
- 选项-s, symbols显示符号表段中的项(如果有数据的话)。
- 选项-e, headers显示全部头信息, 等价于:-h-l-S。
- 选项-n, notes显示note段(内核注释)的信息。
- 选项-r, relocs显示可重定位段的信息。
- 选项-u, unwind显示unwind段信息。当前只支持IA64ELF的unwind段信息。
- 选项-d, dynamic显示动态段的信息。
- 选项-V, version-info显示版本段的信息。
- 选项-A, arch-specific显示CPU构架信息。
- 选项-I, histogram显示符号的时候, 显示bucketlist长度的柱状图。
- 选项-x,hex-dump=以16进制方式显示指定段内内容。number指定段表中段的索引, 或字符串指定文件中的段名
- 选项-D, use-dynamic使用动态段中的符号表显示符号, 而不是使用符号段。
- 选项-v, version显示readelf的版本信息。
- 选项-H, help显示readelf所支持的命令行选项。

原文链接: https://blog.csdn.net/qq_43414142/article/details/90316965

2.objdump - 查看目标文件的信息

objdump - display information from object files.

objdump 可以看目标文件的信息

```
1 objdump -d a.out //-d 参数看代码段反汇编结果
2 objdump -t a.out //显示文件的符号表入口。
3 objdump -t libc.a grep -w printf //查找 printf 在 libc.a 库的哪个目标文件
4 objdump -h simple.o //显示目标文件各个section的头部摘要信息
5 objdump -r simple.o //查看重定位表
6 objdump -f simple.o //显示objfile中每个文件的整体头部摘要信息
7 objdump -s simple.o //显示指定section的完整内容
8 objdump -x simple.o //显示所可用的头信息
9 objdump -a simple.o //显示档案库的成员信息
```

3.readelf和objdump 的区别

3.1 概念的区别

readelf 显示elf文件的信息

objdump 显示目标文件的信息

3.2反汇编

readelf 并不提供反汇编功能

objdump 提供反汇编功能

3.3 调试

readelf 可以显示调试信息

objdump 则不能显示调试信息

但是实际上bfd库支持DWARF的处理，通过简单处理objdump也可以显示调试信息

3.4 文件读取

objdump 使用了bfd库进行文件读取

BFD库 (Binary File Descriptor Library) , 它是一个GNU项目

它的目标就是希望通过一种统一的接口来处理不同的目标文件格式。

而readelf则没有，另外写的一套代码，且对一些条件的判断并不是很严格

readelf的一些信息

```
1 root@f7bf9d5e9abb:/have-fun-debugging/linux-5.15/newtest# readelf -a a.out
2 ELF Header:
3   Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00
4   Class:                               ELF64
5   Data:                               2's complement, little endian
6   Version:                             1 (current)
7   OS/ABI:                               UNIX - System V
8   ABI Version:                           0
9   Type:                                 DYN (Shared object file)
10  Machine:                               Advanced Micro Devices X86-64
11  Version:                               0x1
12  Entry point address:                   0x1060
13  Start of program headers:              64 (bytes into file)
14  Start of section headers:             14712 (bytes into file)
15  Flags:                                  0x0
16  Size of this header:                   64 (bytes)
17  Size of program headers:               56 (bytes)
18  Number of program headers:              13
19  Size of section headers:               64 (bytes)
20  Number of section headers:              31
21  Section header string table index: 30
22
23 Section Headers:
24   [Nr] Name                Type                Address              Offset
25       Size                EntSize            Flags  Link  Info  Align
26   [ 0]                      NULL                0000000000000000    00000000
27       0000000000000000    0000000000000000          0   0   0
28   [ 1] .interp                PROGBITS            0000000000000318    00000318
29       000000000000001c    0000000000000000    A     0   0   1
30   [ 2] .note.gnu.property      NOTE                0000000000000338    00000338
31       0000000000000020    0000000000000000    A     0   0   8
```

```

32 [ 3] .note.gnu.build-id NOTE 0000000000000358 00000358
33 0000000000000024 0000000000000000 A 0 0 4
34 [ 4] .note.ABI-tag NOTE 000000000000037c 0000037c
35 0000000000000020 0000000000000000 A 0 0 4
36 [ 5] .gnu.hash GNU_HASH 00000000000003a0 000003a0
37 0000000000000024 0000000000000000 A 6 0 8
38 [ 6] .dynsym DYNSYM 00000000000003c8 000003c8
39 00000000000000a8 0000000000000018 A 7 1 8
40 [ 7] .dynstr STRTAB 0000000000000470 00000470
41 0000000000000084 0000000000000000 A 0 0 1
42 [ 8] .gnu.version VERSYM 00000000000004f4 000004f4
43 000000000000000e 0000000000000002 A 6 0 2
44 [ 9] .gnu.version_r VERNEED 0000000000000508 00000508
45 0000000000000020 0000000000000000 A 7 1 8
46 [10] .rela.dyn RELA 0000000000000528 00000528
47 00000000000000c0 0000000000000018 A 6 0 8
48 [11] .rela.plt RELA 00000000000005e8 000005e8
49 0000000000000018 0000000000000018 AI 6 24 8
50 [12] .init PROGBITS 0000000000001000 00001000
51 000000000000001b 0000000000000000 AX 0 0 4
52 [13] .plt PROGBITS 0000000000001020 00001020
53 0000000000000020 0000000000000010 AX 0 0 16
54 [14] .plt.got PROGBITS 0000000000001040 00001040
55 0000000000000010 0000000000000010 AX 0 0 16
56 [15] .plt.sec PROGBITS 0000000000001050 00001050
57 0000000000000010 0000000000000010 AX 0 0 16
58 [16] .text PROGBITS 0000000000001060 00001060
59 00000000000000185 0000000000000000 AX 0 0 16
60 [17] .fini PROGBITS 00000000000011e8 000011e8
61 000000000000000d 0000000000000000 AX 0 0 4
62 [18] .rodata PROGBITS 0000000000002000 00002000
63 0000000000000011 0000000000000000 A 0 0 4
64 [19] .eh_frame_hdr PROGBITS 0000000000002014 00002014
65 0000000000000044 0000000000000000 A 0 0 4
66 [20] .eh_frame PROGBITS 0000000000002058 00002058
67 00000000000000108 0000000000000000 A 0 0 8
68 [21] .init_array INIT_ARRAY 0000000000003db8 00002db8
69 0000000000000008 0000000000000008 WA 0 0 8
70 [22] .fini_array FINI_ARRAY 0000000000003dc0 00002dc0
71 0000000000000008 0000000000000008 WA 0 0 8
72 [23] .dynamic DYNAMIC 0000000000003dc8 00002dc8
73 000000000000001f0 0000000000000010 WA 7 0 8
74 [24] .got PROGBITS 0000000000003fb8 00002fb8
75 0000000000000048 0000000000000008 WA 0 0 8
76 [25] .data PROGBITS 0000000000004000 00003000
77 0000000000000010 0000000000000000 WA 0 0 8
78 [26] .bss NOBITS 0000000000004010 00003010
79 0000000000000008 0000000000000000 WA 0 0 1
80 [27] .comment PROGBITS 0000000000000000 00003010
81 000000000000002a 0000000000000001 MS 0 0 1
82 [28] .symtab SYMTAB 0000000000000000 00003040
83 00000000000000618 0000000000000018 29 46 8
84 [29] .strtab STRTAB 0000000000000000 00003658
85 00000000000000204 0000000000000000 0 0 1
86 [30] .shstrtab STRTAB 0000000000000000 0000385c
87 0000000000000011a 0000000000000000 0 0 1

```

Key to Flags:

```

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

```

```

93
94 There are no section groups in this file.
95
96 Program Headers:
97   Type                Offset                VirtAddr                PhysAddr
98   FileSiz             MemSiz             Flags  Align
99   PHDR                 0x0000000000000040 0x0000000000000040 0x0000000000000040
100   0x00000000000002d8 0x00000000000002d8 R      0x8
101   INTERP               0x0000000000000318 0x0000000000000318 0x0000000000000318
102   0x00000000000001c 0x00000000000001c R      0x1
103   [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
104   LOAD                 0x0000000000000000 0x0000000000000000 0x0000000000000000
105   0x0000000000000600 0x0000000000000600 R      0x1000
106   LOAD                 0x00000000000001000 0x00000000000001000 0x00000000000001000
107   0x00000000000001f5 0x00000000000001f5 R E    0x1000
108   LOAD                 0x00000000000002000 0x00000000000002000 0x00000000000002000
109   0x0000000000000160 0x0000000000000160 R      0x1000
110   LOAD                 0x00000000000002db8 0x00000000000003db8 0x00000000000003db8
111   0x0000000000000258 0x0000000000000260 RW     0x1000
112   DYNAMIC              0x00000000000002dc8 0x00000000000003dc8 0x00000000000003dc8
113   0x00000000000001f0 0x00000000000001f0 RW     0x8
114   NOTE                 0x0000000000000338 0x0000000000000338 0x0000000000000338
115   0x0000000000000020 0x0000000000000020 R      0x8
116   NOTE                 0x0000000000000358 0x0000000000000358 0x0000000000000358
117   0x0000000000000044 0x0000000000000044 R      0x4
118   GNU_PROPERTY         0x0000000000000338 0x0000000000000338 0x0000000000000338
119   0x0000000000000020 0x0000000000000020 R      0x8
120   GNU_EH_FRAME         0x00000000000002014 0x00000000000002014 0x00000000000002014
121   0x0000000000000044 0x0000000000000044 R      0x4
122   GNU_STACK            0x0000000000000000 0x0000000000000000 0x0000000000000000
123   0x0000000000000000 0x0000000000000000 RW     0x10
124   GNU_RELRO            0x00000000000002db8 0x00000000000003db8 0x00000000000003db8
125   0x0000000000000248 0x0000000000000248 R      0x1
126
127 Section to Segment mapping:
128 Segment Sections...
129 00
130 01 .interp
131 02 .interp .note.gnu.property .note.gnu.build-id .note.ABI-tag .gnu.hash .dynsym
   .dynstr .gnu.version .gnu.version_r .rela.dyn .rela.plt
132 03 .init .plt .plt.got .plt.sec .text .fini
133 04 .rodata .eh_frame_hdr .eh_frame
134 05 .init_array .fini_array .dynamic .got .data .bss
135 06 .dynamic
136 07 .note.gnu.property
137 08 .note.gnu.build-id .note.ABI-tag
138 09 .note.gnu.property
139 10 .eh_frame_hdr
140 11
141 12 .init_array .fini_array .dynamic .got
142
143 Dynamic section at offset 0x2dc8 contains 27 entries:
144 Tag                Type                Name/Value
145 0x0000000000000001 (NEEDED)           Shared library: [libc.so.6]
146 0x000000000000000c (INIT)             0x1000
147 0x000000000000000d (FINI)             0x11e8
148 0x0000000000000019 (INIT_ARRAY)       0x3db8
149 0x000000000000001b (INIT_ARRAYSZ)      8 (bytes)
150 0x000000000000001a (FINI_ARRAY)       0x3dc0
151 0x000000000000001c (FINI_ARRAYSZ)      8 (bytes)
152 0x000000006ffffef5 (GNU_HASH)           0x3a0

```

```

153 0x0000000000000005 (STRTAB)          0x470
154 0x0000000000000006 (SYMTAB)          0x3c8
155 0x000000000000000a (STRSZ)           132 (bytes)
156 0x000000000000000b (SYMENT)          24 (bytes)
157 0x0000000000000015 (DEBUG)            0x0
158 0x0000000000000003 (PLTGOT)          0x3fb8
159 0x0000000000000002 (PLTRELSZ)        24 (bytes)
160 0x0000000000000014 (PLTREL)           RELA
161 0x0000000000000017 (JMPREL)           0x5e8
162 0x0000000000000007 (RELA)             0x528
163 0x0000000000000008 (RELASZ)           192 (bytes)
164 0x0000000000000009 (RELAENT)          24 (bytes)
165 0x000000000000001e (FLAGS)             BIND_NOW
166 0x0000000006fffffffb (FLAGS_1)        Flags: NOW PIE
167 0x0000000006fffffffe (VERNEED)        0x508
168 0x0000000006fffffff (VERNEEDNUM)      1
169 0x0000000006fffffff0 (VERSYM)          0x4f4
170 0x0000000006fffffff9 (RELACOUNT)      3
171 0x0000000000000000 (NULL)             0x0
172
173 Relocation section '.rela.dyn' at offset 0x528 contains 8 entries:
174   Offset          Info          Type          Sym. Value      Sym. Name + Addend
175 000000003db8 0000000000000008 R_X86_64_RELATIVE          1140
176 000000003dc0 0000000000000008 R_X86_64_RELATIVE          1100
177 000000004008 0000000000000008 R_X86_64_RELATIVE          4008
178 000000003fd8 0001000000000006 R_X86_64_GLOB_DAT 0000000000000000 _ITM_deregisterTMClone + 0
179 000000003fe0 0003000000000006 R_X86_64_GLOB_DAT 0000000000000000
__libc_start_main@GLIBC_2.2.5 + 0
180 000000003fe8 0004000000000006 R_X86_64_GLOB_DAT 0000000000000000 __gmon_start__ + 0
181 000000003ff0 0005000000000006 R_X86_64_GLOB_DAT 0000000000000000 _ITM_registerTMCloneTa + 0
182 000000003ff8 0006000000000006 R_X86_64_GLOB_DAT 0000000000000000 __cxa_finalize@GLIBC_2.2.5 +
0
183
184 Relocation section '.rela.plt' at offset 0x5e8 contains 1 entry:
185   Offset          Info          Type          Sym. Value      Sym. Name + Addend
186 000000003fd0 0002000000000007 R_X86_64_JUMP_SLO 0000000000000000 printf@GLIBC_2.2.5 + 0
187
188 The decoding of unwind sections for machine type Advanced Micro Devices X86-64 is not
currently supported.
189
190 Symbol table '.dynsym' contains 7 entries:
191   Num:   Value          Size Type   Bind   Vis     Ndx Name
192   0: 0000000000000000      0 NOTYPE LOCAL DEFAULT UND
193   1: 0000000000000000      0 NOTYPE WEAK  DEFAULT UND _ITM_deregisterTMCloneTab
194   2: 0000000000000000      0 FUNC  GLOBAL DEFAULT UND printf@GLIBC_2.2.5 (2)
195   3: 0000000000000000      0 FUNC  GLOBAL DEFAULT UND __libc_start_main@GLIBC_2.2.5
(2)
196   4: 0000000000000000      0 NOTYPE WEAK  DEFAULT UND __gmon_start__
197   5: 0000000000000000      0 NOTYPE WEAK  DEFAULT UND _ITM_registerTMCloneTable
198   6: 0000000000000000      0 FUNC  WEAK  DEFAULT UND __cxa_finalize@GLIBC_2.2.5 (2)
199
200 Symbol table '.symtab' contains 65 entries:
201   Num:   Value          Size Type   Bind   Vis     Ndx Name
202   0: 0000000000000000      0 NOTYPE LOCAL DEFAULT UND
203   1: 0000000000000318      0 SECTION LOCAL DEFAULT 1
204   2: 0000000000000338      0 SECTION LOCAL DEFAULT 2
205   3: 0000000000000358      0 SECTION LOCAL DEFAULT 3
206   4: 000000000000037c      0 SECTION LOCAL DEFAULT 4
207   5: 00000000000003a0      0 SECTION LOCAL DEFAULT 5
208   6: 00000000000003c8      0 SECTION LOCAL DEFAULT 6
209   7: 0000000000000470      0 SECTION LOCAL DEFAULT 7

```

210	8: 00000000000004f4	0	SECTION	LOCAL	DEFAULT	8
211	9: 0000000000000508	0	SECTION	LOCAL	DEFAULT	9
212	10: 0000000000000528	0	SECTION	LOCAL	DEFAULT	10
213	11: 00000000000005e8	0	SECTION	LOCAL	DEFAULT	11
214	12: 0000000000001000	0	SECTION	LOCAL	DEFAULT	12
215	13: 0000000000001020	0	SECTION	LOCAL	DEFAULT	13
216	14: 0000000000001040	0	SECTION	LOCAL	DEFAULT	14
217	15: 0000000000001050	0	SECTION	LOCAL	DEFAULT	15
218	16: 0000000000001060	0	SECTION	LOCAL	DEFAULT	16
219	17: 00000000000011e8	0	SECTION	LOCAL	DEFAULT	17
220	18: 0000000000002000	0	SECTION	LOCAL	DEFAULT	18
221	19: 0000000000002014	0	SECTION	LOCAL	DEFAULT	19
222	20: 0000000000002058	0	SECTION	LOCAL	DEFAULT	20
223	21: 0000000000003db8	0	SECTION	LOCAL	DEFAULT	21
224	22: 0000000000003dc0	0	SECTION	LOCAL	DEFAULT	22
225	23: 0000000000003dc8	0	SECTION	LOCAL	DEFAULT	23
226	24: 0000000000003fb8	0	SECTION	LOCAL	DEFAULT	24
227	25: 0000000000004000	0	SECTION	LOCAL	DEFAULT	25
228	26: 0000000000004010	0	SECTION	LOCAL	DEFAULT	26
229	27: 0000000000000000	0	SECTION	LOCAL	DEFAULT	27
230	28: 0000000000000000	0	FILE	LOCAL	DEFAULT	ABS crtstuff.c
231	29: 0000000000001090	0	FUNC	LOCAL	DEFAULT	16 deregister_tm_clones
232	30: 00000000000010c0	0	FUNC	LOCAL	DEFAULT	16 register_tm_clones
233	31: 0000000000001100	0	FUNC	LOCAL	DEFAULT	16 __do_global_dtors_aux
234	32: 0000000000004010	1	OBJECT	LOCAL	DEFAULT	26 completed.8060
235	33: 0000000000003dc0	0	OBJECT	LOCAL	DEFAULT	22 __do_global_dtors_aux_fin
236	34: 0000000000001140	0	FUNC	LOCAL	DEFAULT	16 frame_dummy
237	35: 0000000000003db8	0	OBJECT	LOCAL	DEFAULT	21 __frame_dummy_init_array_
238	36: 0000000000000000	0	FILE	LOCAL	DEFAULT	ABS test.c
239	37: 0000000000000000	0	FILE	LOCAL	DEFAULT	ABS crtstuff.c
240	38: 000000000000215c	0	OBJECT	LOCAL	DEFAULT	20 __FRAME_END__
241	39: 0000000000000000	0	FILE	LOCAL	DEFAULT	ABS
242	40: 0000000000003dc0	0	NOTYPE	LOCAL	DEFAULT	21 __init_array_end
243	41: 0000000000003dc8	0	OBJECT	LOCAL	DEFAULT	23 _DYNAMIC
244	42: 0000000000003db8	0	NOTYPE	LOCAL	DEFAULT	21 __init_array_start
245	43: 0000000000002014	0	NOTYPE	LOCAL	DEFAULT	19 __GNU_EH_FRAME_HDR
246	44: 0000000000003fb8	0	OBJECT	LOCAL	DEFAULT	24 _GLOBAL_OFFSET_TABLE_
247	45: 0000000000001000	0	FUNC	LOCAL	DEFAULT	12 _init
248	46: 00000000000011e0	5	FUNC	GLOBAL	DEFAULT	16 __libc_csu_fini
249	47: 0000000000000000	0	NOTYPE	WEAK	DEFAULT	UND _ITM_deregisterTMCloneTab
250	48: 0000000000004000	0	NOTYPE	WEAK	DEFAULT	25 data_start
251	49: 0000000000004010	0	NOTYPE	GLOBAL	DEFAULT	25 _edata
252	50: 00000000000011e8	0	FUNC	GLOBAL	HIDDEN	17 _fini
253	51: 0000000000000000	0	FUNC	GLOBAL	DEFAULT	UND printf@@GLIBC_2.2.5
254	52: 0000000000000000	0	FUNC	GLOBAL	DEFAULT	UND __libc_start_main@@GLIBC_
255	53: 0000000000004000	0	NOTYPE	GLOBAL	DEFAULT	25 __data_start
256	54: 0000000000000000	0	NOTYPE	WEAK	DEFAULT	UND __gmon_start__
257	55: 0000000000004008	0	OBJECT	GLOBAL	HIDDEN	25 __dso_handle
258	56: 0000000000002000	4	OBJECT	GLOBAL	DEFAULT	18 _IO_stdin_used
259	57: 0000000000001170	101	FUNC	GLOBAL	DEFAULT	16 __libc_csu_init
260	58: 0000000000004018	0	NOTYPE	GLOBAL	DEFAULT	26 _end
261	59: 0000000000001060	47	FUNC	GLOBAL	DEFAULT	16 _start
262	60: 0000000000004010	0	NOTYPE	GLOBAL	DEFAULT	26 __bss_start
263	61: 0000000000001149	32	FUNC	GLOBAL	DEFAULT	16 main
264	62: 0000000000004010	0	OBJECT	GLOBAL	HIDDEN	25 __TMC_END__
265	63: 0000000000000000	0	NOTYPE	WEAK	DEFAULT	UND _ITM_registerTMCloneTable
266	64: 0000000000000000	0	FUNC	WEAK	DEFAULT	UND __cxa_finalize@@GLIBC_2.2

267
 268 Histogram for `gnu.hash' bucket list length (total of 2 buckets):

Length	Number	% of total	Coverage
0	1	(50.0%)	

```

271      1 1          ( 50.0%)   100.0%
272
273 Version symbols section '.gnu.version' contains 7 entries:
274 Addr: 0x00000000000004f4 Offset: 0x0004f4 Link: 6 (.dynsym)
275 000: 0 (*local*)      0 (*local*)      2 (GLIBC_2.2.5)  2 (GLIBC_2.2.5)
276 004: 0 (*local*)      0 (*local*)      2 (GLIBC_2.2.5)
277
278 Version needs section '.gnu.version_r' contains 1 entry:
279 Addr: 0x0000000000000508 Offset: 0x000508 Link: 7 (.dynstr)
280 000000: Version: 1 File: libc.so.6 Cnt: 1
281 0x0010: Name: GLIBC_2.2.5 Flags: none Version: 2
282
283 Displaying notes found in: .note.gnu.property
284 Owner          Data size  Description
285 GNU            0x00000010  NT_GNU_PROPERTY_TYPE_0
286 Properties: x86 feature: IBT, SHSTK
287
288 Displaying notes found in: .note.gnu.build-id
289 Owner          Data size  Description
290 GNU            0x00000014  NT_GNU_BUILD_ID (unique build ID bitstring)
291 Build ID: b1d41147044210fa3a8c5a2acb1a0bf9dd9caabb
292
293 Displaying notes found in: .note.ABI-tag
294 Owner          Data size  Description
295 GNU            0x00000010  NT_GNU_ABI_TAG (ABI version tag)
296 OS: Linux, ABI: 3.2.0
297

```

objdump的一些调试信息

-t //显示文件的符号表入口。

```

1 root@f7bf9d5e9abb:/have-fun-debugging/linux-5.15/newtest# objdump -t a.out
2
3 a.out:      file format elf64-x86-64
4
5 SYMBOL TABLE:
6 0000000000000318 l d .interp 0000000000000000 .interp
7 0000000000000338 l d .note.gnu.property 0000000000000000 .note.gnu.property
8 0000000000000358 l d .note.gnu.build-id 0000000000000000 .note.gnu.build-id
9 000000000000037c l d .note.ABI-tag 0000000000000000 .note.ABI-tag
10 00000000000003a0 l d .gnu.hash 0000000000000000 .gnu.hash
11 00000000000003c8 l d .dynsym 0000000000000000 .dynsym
12 0000000000000470 l d .dynstr 0000000000000000 .dynstr
13 00000000000004f4 l d .gnu.version 0000000000000000 .gnu.version
14 0000000000000508 l d .gnu.version_r 0000000000000000 .gnu.version_r
15 0000000000000528 l d .rela.dyn 0000000000000000 .rela.dyn
16 00000000000005e8 l d .rela.plt 0000000000000000 .rela.plt
17 0000000000001000 l d .init 0000000000000000 .init
18 0000000000001020 l d .plt 0000000000000000 .plt
19 0000000000001040 l d .plt.got 0000000000000000 .plt.got
20 0000000000001050 l d .plt.sec 0000000000000000 .plt.sec
21 0000000000001060 l d .text 0000000000000000 .text
22 00000000000011e8 l d .fini 0000000000000000 .fini
23 0000000000002000 l d .rodata 0000000000000000 .rodata
24 0000000000002014 l d .eh_frame_hdr 0000000000000000 .eh_frame_hdr
25 0000000000002058 l d .eh_frame 0000000000000000 .eh_frame
26 0000000000003db8 l d .init_array 0000000000000000 .init_array

```

27	0000000000003dc0	l	d	.fini_array	0000000000000000	.fini_array
28	0000000000003dc8	l	d	.dynamic	0000000000000000	.dynamic
29	0000000000003fb8	l	d	.got	0000000000000000	.got
30	0000000000004000	l	d	.data	0000000000000000	.data
31	0000000000004010	l	d	.bss	0000000000000000	.bss
32	0000000000000000	l	d	.comment	0000000000000000	.comment
33	0000000000000000	l	df	*ABS*	0000000000000000	crtstuff.c
34	0000000000001090	l	F	.text	0000000000000000	deregister_tm_clones
35	00000000000010c0	l	F	.text	0000000000000000	register_tm_clones
36	0000000000001100	l	F	.text	0000000000000000	__do_global_ctors_aux
37	0000000000004010	l	O	.bss	0000000000000001	completed.8060
38	0000000000003dc0	l	O	.fini_array	0000000000000000	
	__do_global_ctors_aux_fini_array_entry					
39	0000000000001140	l	F	.text	0000000000000000	frame_dummy
40	0000000000003db8	l	O	.init_array	0000000000000000	
	__frame_dummy_init_array_entry					
41	0000000000000000	l	df	*ABS*	0000000000000000	test.c
42	0000000000000000	l	df	*ABS*	0000000000000000	crtstuff.c
43	000000000000215c	l	O	.eh_frame	0000000000000000	__FRAME_END__
44	0000000000000000	l	df	*ABS*	0000000000000000	
45	0000000000003dc0	l		.init_array	0000000000000000	__init_array_end
46	0000000000003dc8	l	O	.dynamic	0000000000000000	__DYNAMIC
47	0000000000003db8	l		.init_array	0000000000000000	__init_array_start
48	0000000000002014	l		.eh_frame_hdr	0000000000000000	__GNU_EH_FRAME_HDR
49	0000000000003fb8	l	O	.got	0000000000000000	__GLOBAL_OFFSET_TABLE__
50	0000000000001000	l	F	.init	0000000000000000	__init
51	00000000000011e0	g	F	.text	0000000000000005	__libc_csu_fini
52	0000000000000000	w		*UND*	0000000000000000	__ITM_deregisterTMCloneTable
53	0000000000004000	w		.data	0000000000000000	data_start
54	0000000000004010	g		.data	0000000000000000	__edata
55	00000000000011e8	g	F	.fini	0000000000000000	.hidden __fini
56	0000000000000000		F	*UND*	0000000000000000	printf@@GLIBC_2.2.5
57	0000000000000000		F	*UND*	0000000000000000	__libc_start_main@@GLIBC_2.2.5
58	0000000000004000	g		.data	0000000000000000	__data_start
59	0000000000000000	w		*UND*	0000000000000000	__gmon_start__
60	0000000000004008	g	O	.data	0000000000000000	.hidden __dso_handle
61	0000000000002000	g	O	.rodata	0000000000000004	__IO_stdin_used
62	0000000000001170	g	F	.text	0000000000000065	__libc_csu_init
63	0000000000004018	g		.bss	0000000000000000	__end
64	0000000000001060	g	F	.text	000000000000002f	__start
65	0000000000004010	g		.bss	0000000000000000	__bss_start
66	0000000000001149	g	F	.text	0000000000000020	main
67	0000000000004010	g	O	.data	0000000000000000	.hidden __TMC_END__
68	0000000000000000	w		*UND*	0000000000000000	__ITM_registerTMCloneTable
69	0000000000000000	w	F	*UND*	0000000000000000	__cxa_finalize@@GLIBC_2.2.5
70						
71						