Hello~, ELF of the world!

想熟悉 ELF 格式，可以先阅读《Executable and Linkable Format (ELF)》，也有网友把她翻译成了中文。然后结合实例加深理解，或结合阅读 readelf 命令源码(在 gnu 的 binutils 包中)。尽管《Executable and Linkable Format (ELF)》做了详尽的描述，但有时就是这样，虽然说的很明白了，可就是还是不明白。所以本文以一种罗嗦的方式来做进一步详尽的举例分析。

附录 A 中附上完整的 Helloworld 可执行二进制文件内容这，附录 B 附上该文件从头至尾几乎所有二进制内容的文本解释(节头表中。20C5-2618 是什么？：))。我们会以我们熟悉的 Hello,world 为例，尽量来剖析一下怎样将附录 A 中的 ELF 可执行文件的格式如何解释为附录 B 的内容。不过，对于其中的链接(静态链接，动态链接)相关主题不做深入研究，将会把这个议题单独拿出做进一步讨论。至于 ELF 文件在操作系统角度来如何执行运行，也在其它文档中做相应的论述。

Executable and Linkable Format (ELF)

可执行连接格式是 UNIX 系统实验室(USL)作为应用程序二进制接口(Application Binary Interface(ABI)而开发和发布的。工具接口标准委员会(TIS)选择了正在发展中的 ELF 标准作为工作在 32 位 INTEL 体系上不同操作系统之间可移植的二进制文件格式。

就象我们了解的那样，可执行文件就是一串 0、1 的组合，里面有机器指令也有数据。色彩缤纷的计算机世界便是由这 0、1 组成，可想而知，最初的真正的计算机或软件高手，估计都是由那些看习惯灯亮灯灭的硬件人员发展而来，弓着腰猫在电脑前面咣咣地敲着键盘，而所谓的键盘可能上面只有两个键，那就是数字 0 和 1(这才是传说中真正的高手，呵呵)。电脑看起来如此亲切的机器指令对于高级的人类来说却是莫大的考验：太难记了，太容易出错了。于是，人们便不遗余力地努力着，先使用一些相对容易记忆书写的助记符写出代码，这就是汇编语言，然后再把这些肋记符翻译成机器指令。过了几天，人们发现这是这些好记的助记符也不是那么好接受，于是就发明了更为好记的语言，然后花更大的气力把它们翻译成机器语言。这就是什么 B 语言，C 语言，VC，BC，JAVA 等的由来。人们总是变着法子，把电脑世界的思维纳入人类思维的轨迹之内。这样看来，估计那些总是尝试让自己能够理解模仿电脑思维的人更有可能成为电脑高手，但这也是有着很大风险的举动。因为当你习惯了电脑世界简单的 0、1 思维，你就可能发现自己会经常性的困惑于这个现实世界。现实世界是这样的复杂，不是简单的 0 和 1，好人和坏人能区分的。所以你会渐渐地孤僻，深陷机器世界而远离现实世界。高级语言的产生应该还是符合时代潮流滴，那就是让人拥有人的思维吧。

呵呵，跑题了。回来接着往下说。通常，编译器和汇编器产生目标文件，而链接器需要最小程度的理解目标机器特性。一般来说，有三类 ELF 文件：

✧ .可执行文件：包含了代码和数据。具有可执行的程序。 例如：

```
# file hello
hello: ELF 32-bit LSB executable, Intel 80386, version 1, dynamically linked (uses shared libs), not stripped
```

✧ .可重定位文件：包含了代码和数据（这些数据是和其他重定位文件和共享的 object 文件一起连接时使用的）。例如：

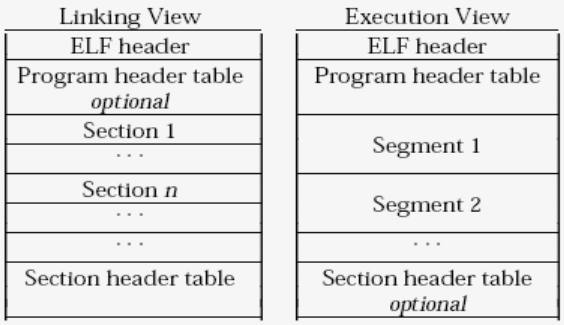
```
# file libfoo.o
libfoo.o: ELF 32-bit LSB relocatable, Intel 80386, version 1, not stripped
```

✧ .共享 object 文件（又可叫做共享库）：包含了代码和数据（这些数据是在连接时候被连接器 ld 和运行 时动态连接器使用的）。动态连接器可能称为 ld.so.2,libc.so.2 或者 ld-linux.so.2。例如：

```
# file libfoo.so
libfoo.so: ELF 32-bit LSB shared object, Intel 80386, version1, not stripped
```

编译器和汇编器可以产生重定位的目标文件，而链接器则产生可执行文件。目标文件参与程序的链接及执行。目标文件格式，站在不同的角度有如下不同的内容：

Figure 1-1: Object File Format



除了 ELF 头，在链接的角度上看，目标文件包括指令，数据，符号表，重定位信息等；在执行的角度上看，目标文件必须至少有一个程序头表，用来告诉系统如何来创建一个进程的内存映像。

在分析 ELF 格式之前，让我们先熟悉一下这些数据结构类型定义。这些结构定义都可以在 Linux 源码中找到(\linux-2.6.5\include\linux\elf.h)。

```
/* 32-bit ELF base types. */
typedef __u32 Elf32_Addr;
typedef __u16 Elf32_Half;
typedef __u32 Elf32_Off;
typedef __s32 Elf32_Sword;
typedef __u32 Elf32_Word;
```

如大家在 Figure1-1 中看到的，一个 ELF 可执行文件的内容，我们基本关心三个方面，那就是：ELF 头，程序头表，节头表。

为了更有效地介绍这些内容，还是在下面结合实际的可执行文件内容来依次讲解。在后面附录中附上整个 Helloworld 可执行文件二进制内容。在文中为了不会因为图占用太大篇幅而影响大家对叙述内容连贯性的理解，采用在需要对照二进制内容处只截取讲述部分的二进制内容而不是整个二进制文件内容，大家若想看看相关段落的上下文可自行对照附录 A 中完整的内容。附录 B 中为 ELF 文件的文本解释内容。

Hello,world

看一下著名的 Hello,world：

```
<onlyforos>[/home/onlyforos/hello]%cat hello.c
```

```
#include <stdio.h>
int main()
{
	printf("hello,world\n");
	return 0;
}

<onlyforos>[/home/onlyforos/hello]%gcc -o hello hello.c
<onlyforos>[/home/onlyforos/hello]%hello
hello,world
<onlyforos>[/home/onlyforos/hello]%ll hello
-rwxr-xr-x	1 onlyforos onlyforos	11362	6 月	9 15:15 hello*
<onlyforos>[/home/onlyforos/hello]%
```

从附录 A 中可以看到 Hello 文件的大小为 0x2c62，即 11362 字节。使用可以查看二进制文件的编辑器如 Ultra-edit，或使用 hexdump 等命令，分析其内容的含义。

ELF Header

一个ELF可执行文件的开始，一定是一个ELF头。ELF头的数据结构(所有结构参见include/linux/elf.h)为：

```
#define EI_NIDENT	16

typedef struct elf32_hdr{
  unsigned char	e_ident[EI_NIDENT];
  Elf32_Half	e_type;
  Elf32_Half	e_machine;
  Elf32_Word	e_version;
  Elf32_Addr	e_entry;  /* Entry point */
  Elf32_Off	e_phoff;
  Elf32_Off	e_shoff;
  Elf32_Word	e_flags;
  Elf32_Half	e_ehsize;
  Elf32_Half	e_phentsize;
  Elf32_Half	e_phnum;
  Elf32_Half	e_shentsize;
  Elf32_Half	e_shnum;
  Elf32_Half	e_shstrndx;
} Elf32_Ehdr;
```

下面简要介绍一下结构中各字段的含义

---

e_ident：这16个字节的头4个字节，固定为'7f' 'E' 'L' 'F'，接下来4-6(数组下标从0开始)个字节分别意味着：机器类型(1,32位机；2,64位机)；字节序(1，小端；2，大端)；版本(要设为EV_CURRENT，即1，同下面e_version一致)，之后的字节均为填充。

e_type：目标文件类型(其中，1，重定位文件，2，可执行文件，3，共享目标文件，4，core文件等)

e_machine：机器体系结构(如3,intel 80386等)

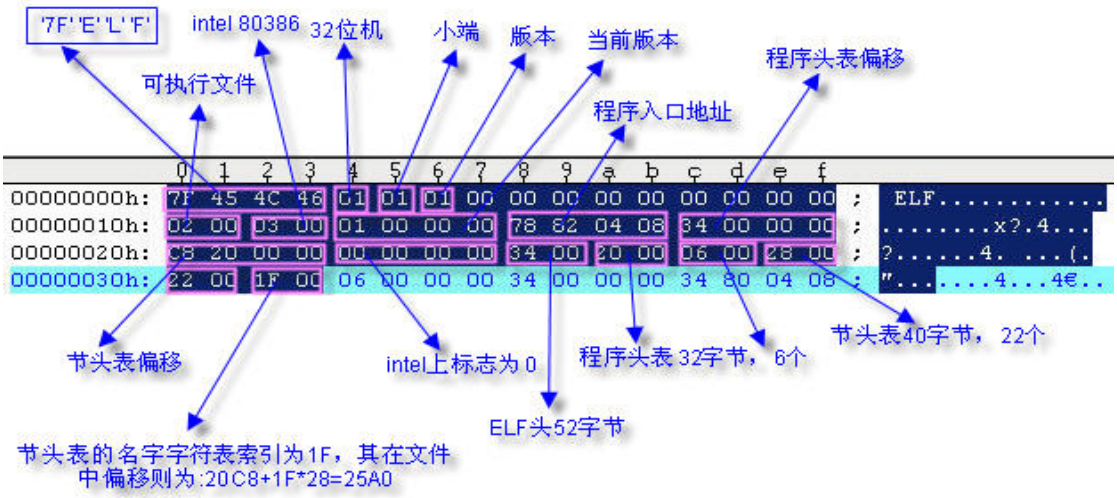
e_version： EV_CURRENT，当前版本

e_entry： 程序入口地址

e_phoff： 程序头表在文件的偏移

e_shoff： 节头表在文件中的偏移
e_flags： 保存着相关文件的特定处理器标志。32位Intel体系上未定义该标记，所以为0。
e_ehsize： ELF头的大小
e_phentsize：程序头表的大小
e_phnum： 程序头表的个数
e_shentsize：节头表的大小
e_shnum： 节头表的个数
e_shstrndx： 节头名字字符表的索引

对照可执行内容来看：



从上图可看出，ELF头之后，就应该是程序头表。因为程序头表的偏移就是从文件的0x34字节处开始。因为Intel i386体系结构为小端字节序，要注意字节在内存中的排列。如程序入口地址在内存中看到的为：78 82 04 08，但因为在小端机器上，所以实际程序入口地址则为：0x8048278。若对于字节序还不是很了解，请参看文档《Byte_order_and_Bit_order》。从ELF头可知，Helloworld可执行文件共有6个程序头表。下面开始分析程序头表。

Program Header

对于可执行文件来说，ELF 头之后就是程序头表。从 ELF 头中可以得知，程序头表大小为 0x20，总共有 6 个。

我们首先来看看程序头表的数据结构，并简要介绍一下各字段含义，更为详细的介绍请参考《Executable and Linkable Format (ELF)》：

```
typedef struct elf32_phdr{
  Elf32_Word    p_type;
  Elf32_Off     p_offset;
  Elf32_Addr    p_vaddr;
  Elf32_Addr    p_paddr;
  Elf32_Word    p_filesz;
  Elf32_Word    p_memsz;
  Elf32_Word    p_flags;
  Elf32_Word    p_align;
} Elf32_Phdr;
```

可以看到，sizeof(Elf32_Phdr)=32。各字段的含义：

p_type :指出了这个数组的元素描述了什么类型的段，或怎样解释该数组元素的信息。主要类型有：

Name	Value
PT_NULL	0
PT_LOAD	1
PT_DYNAMIC	2
PT_INTERP	3
PT_NOTE	4
PT_SHLIB	5
PT_PHDR	6
PT_LOPROC	0x70000000
PT_HIPROC	0x7fffffff

具体各类型含义结合实例讲述。

p_offset : 该段的驻留位置相对于文件开始处的偏移。

p_vaddr : 该段在内存中的首字节地址。

p_paddr : 该段的物理地址。

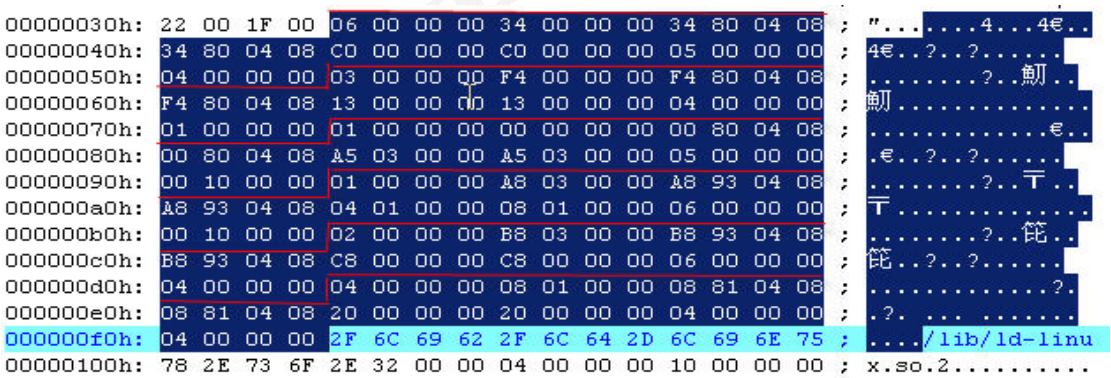
p_filesz : 文件映像中该段的字节数。

p_memsz : 内存映像中该段的字节数。

p_flags : 该段相关的标志。(PF_R，PF_W，PF_X)

p_align : 对齐字节。

ELF 头之后的 6 个程序头表如下图：



下面我们依次分析这 6 个程序头表的含义。

1. PT_PHDR

p_type = PT_PHDR: 指定了程序头表本身的位置和大小（包括在文件中和在存映像中）。

p_offset = 0x34: 从文件的 0x34 偏移处开始

p_vaddr = 0x8048034: 加载的虚存地址

p_paddr = 0x8048034: 加载的物理地址

p_filesz = 0xC0: 加载的大小为 0xC0 字节，即加载从偏移 0x34 到 0xF4 处的内容。

p_memsz = 0xC0: 加载内存的大小。

p_flags = 5: 可读可执行

p_align = 4: 4 字节对齐

Linux 下链接器默认加载地址为 0x8048000。所以该程序头表指示把从文件偏移 0x34 处开始的 0xC0 字节内容，加载到虚存 0x8048034 处，可读可执行，4 字节对齐。那么这 0xC0 字节内容其实就是 6 个程序头表本身。

2. PT_INTERP

p_type = PT_INTERP: 指定一个 null-terminated 路径名的位置和大小（作为解释程序）。

p_offset = 0xF4: 从文件的 0xF4 偏移处开始

p_vaddr = 0x80480F4: 加载的虚存地址

p_paddr = 0x80480F4: 加载的物理地址

p_filesz = 0x13: 加载的大小为 0xC0 字节，即加载从偏移 0x34 到 0xF4 处的内容。

p_memsz = 0x13: 加载内存的大小。

p_flags = 4: 可读

p_align = 1: 1 字节对齐

把从 0xF4 开始的 0x13 字节，即到 0x107 的内容接着前面内容顺序加载到 0x80480F4。那么这 0x13 字节又是什么内容呢？

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 244 –n 19 –C hello >> ph2
<onlyforos>[/home/onlyforos/hello]%cat ph2
000000f4    2f 6c 69 62 2f 6c 64 2d    6c 69 6e 75 78 2e 73 6f    |/lib/ld-linux.so|
00000104    2e 32 00                                              |.2.             |
```

可见，这个程序头表加载的内容就是加载动态库程序名称：/lib/ld-linux.so.2。

3. PT_LOAD

p_type = PT_LOAD: 指定一个可载入的段，由 p_filesz 和 p_memsz 描述。

p_offset = 0x00000000: 从文件的 0x34 偏移处开始

p_vaddr = 0x8048000: 加载的虚存地址

p_paddr = 0x8048000: 加载的物理地址

p_filesz = 0x3A5: 加载的大小为 0xC0 字节，即加载从偏移 0x34 到 0xF4 处的内容。

p_memsz = 0x3A5: 加载内存的大小。

p_flags = 5: 可读可执行

p_align = 0x100000: 4K 字节对齐

可见，这个程序头表把从文件开始直到 0x3A5 偏移处的内容全部加载到从 0x8048000 开始的内存空间中去。内容较多，请自行参照附录 A。

4. PT_ LOAD

p_type = PT_ LOAD: 指定一个可载入的段，由 p_filesz 和 p_memsz 描述。p_memsz 比 p_filesz 大，是因为.bss session 具有 SHT_NOBITS 的类型。尽管在文件中不占用空间，它在段的内存映像中起作用。通常，没有初始化的数据驻留在段尾，因此使得在相关的程序头元素中的 p_memsz 比 p_filesz 大。

p_offset = 0x3A8: 从文件的 0x34 偏移处开始

p_vaddr = 0x80493A8: 加载的虚存地址

p_paddr = 0x80493A8: 加载的物理地址

p_filesz = 0x104: 加载文件的大小为 0x104 字节。

p_memsz = 0x108: 加载内存的大小。

p_flags = 5: 可读可执行

p_align = 0x100000: 4K 字节对齐

将数据段加载到 0x84093a8。从附录 B 可以看到，数据段包括了.data, . eh_frame, . dynamic, . ctors, . dtors, . jcr, .got, .bss 等 session 的数据。

5. PT_ DYNAMIC

p_type = PT_ DYNAMIC: 该数组元素指定动态链接信息。

p_offset = 0x3B8: 从文件的 0x3B8 偏移处开始

p_vaddr = 0x80493B8: 加载的虚存地址

p_paddr = 0x80493B8: 加载的物理地址

p_filesz = 0xC8: 加载的文件大小为 0xC8 字节。

p_memsz = 0xC8: 加载内存的大小。

p_flags = 5: 可读可执行

p_align = 4: 4 字节对齐

将. dynamic session 数据加载到 0x80493B8。

6. PT_NOTE

p_type = PT_NOTE: 指定辅助信息的位置和大小。

p_offset = 0x108: 从文件的 0x34 偏移处开始

p_vaddr = 0x8048108: 加载的虚存地址

p_paddr = 0x8048108: 加载的物理地址

p_filesz = 0x20: 加载的大小为 0xC0 字节，即加载从偏移 0x34 到 0xF4 处的内容。

p_memsz = 0x20: 加载内存的大小。

p_flags = 4: 可读

p_align = 4: 4 字节对齐

7. All programm header

其实，readelf 工具很容易就可以告诉你程序头表的含义：

```
<onlyforos>[/home/onlyforos/hello]%readelf –l hello >> ph.txt
<onlyforos>[/home/onlyforos/hello]%cat ph.txt
Elf file type is EXEC (Executable file)
Entry point 0x8048278
There are 6 program headers, starting at offset 52

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x08048034 0x08048034 0x000c0 0x000c0 R E 0x4
  INTERP         0x0000f4 0x080480f4 0x080480f4 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /lib/ld-linux.so.2]
  LOAD           0x000000 0x08048000 0x08048000 0x003a5 0x003a5 R E 0x1000
  LOAD           0x0003a8 0x080493a8 0x080493a8 0x00104 0x00108 RW  0x1000
  DYNAMIC        0x0003b8 0x080493b8 0x080493b8 0x000c8 0x000c8 RW  0x4
  NOTE           0x000108 0x08048108 0x08048108 0x00020 0x00020 R   0x4

 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp

02     .interp .note.ABI-tag .hash .dynsym .dynstr .gnu.version .gnu.version_r .rel.dyn .rel.plt .in
it .plt .text .fini .rodata
   03     .data .eh_frame .dynamic .ctors .dtors .jcr .got .bss
   04     .dynamic
   05     .note.ABI-tag
```

Session Header

接着我们来分析有关 session 的内容。同样首先来看看 session 头的数据结构：

```
typedef struct {
  Elf32_Word    sh_name;
  Elf32_Word    sh_type;
  Elf32_Word    sh_flags;
  Elf32_Addr    sh_addr;
  Elf32_Off     sh_offset;
  Elf32_Word    sh_size;
  Elf32_Word    sh_link;
  Elf32_Word    sh_info;
  Elf32_Word    sh_addralign;
  Elf32_Word    sh_entsize;
} Elf32_Shdr;
```

各字段含义：

sh_name: 指定了这个 section 的名字。它的值是 section 报头字符表 section 的索引。

sh_type: 该成员把 sections 按内容和意义分类。

Name	Value
SHT_NULL	0
SHT_PROGBITS	1
SHT_SYMTAB	2
SHT_STRTAB	3
SHT_RELA	4
SHT_HASH	5
SHT_DYNAMIC	6
SHT_NOTE	7
SHT_NOBITS	8
SHT_REL	9
SHT_SHLIB	10
SHT_DYNSYM	11
SHT_LOPROC	0x70000000
SHT_HIPROC	0x7fffffff
SHT_LOUSER	0x80000000
SHT_HIUSER	0xffffffff

sh_flags: sections 支持位的标记，用来描述多个属性。

Name	Value
SHF_WRITE	0x1
SHF_ALLOC	0x2
SHF_EXECINSTR	0x4
SHF_MASKPROC	0xf0000000

sh_addr: 若该 session 将出现在进程的内存映象中，该成员给出了该 section 在内存中的位置。

sh_offset: 给出了该 section 的字节偏移量(从文件开始计数)。

sh_size: 该成员给你了 section 的字节大小。

sh_link: 该成员保存了一个 section 报头表的索引连接，它的解释依靠该 section 的类型。

sh_info: 该成员保存着额外的信息，它的解释依靠该 section 的类型。

在 section 报头中，两个成员 sh_link 和 sh_info 的解释依靠该 section 的类型:

sh_type	sh_link	sh_info
SHT_DYNAMIC	The section header index of the string table used by entries in the section.	0
SHT_HASH	The section header index of the symbol table to which the hash table applies.	0
SHT_REL, SHT_RELA	The section header index of the associated symbol table.	The section header index of the section to which the relocation applies.
SHT_SYMTAB, SHT_DYNSYM	The section header index of the associated string table.	One greater than the symbol table index of the last local symbol (binding STB_LOCAL).
other	SHN_UNDEF	0

sh_addralign : 地址对齐的约束。

sh_entsize: 一些 sections 保存着一张固定大小入口的表，就象符号表。对于这样一个 section 来说，该成员给出了每个入口的字节大小。

---

sizeof(Elf32_Shdr)=40(0x28)。从 ELF 头中可知，session 头在文件中的偏移为：.20C8 字节，大小为 40 字节，34 个，可知所有节头表在 20C8 和 2618 之间(20C8+28*22=2618)。

session 头的名字字符串表的偏移为：20C8+1F*28=25A0(1F：e_shstrndx；0x28：session 大小)。由于有 34 个 session header table，都进行分析一来篇幅较大二来看起来罗嗦，所以我们不再从偏移 20C8 处开始一一说明，只挑一些较为重要的进行分析，其余的若感兴趣，可自行进行分析。首先，我们就分析在 ELF 有个索引的.shstrtab：

1. .shstrtab

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x25A0 –n 40 –C hello >> shstrtab.txt
<onlyforos>[/home/onlyforos/hello]%cat shstrtab.txt
000025a0    11 00 00 00 03 00 00 00    00 00 00 00 00 00 00 00    |................|
000025b0    9a 1f 00 00 2b 01 00 00    00 00 00 00 00 00 00 00    |....+...........|
000025c0    01 00 00 00 00 00 00 00                               |........|
```

sh_name = 0x11: 索引为 0x11.

sh_type = SHT_STRTAB: 该 section 保存着一个字符串表。

sh_flags = 0: 未明的属性设为 0。

sh_addr = 0: 不出现在进程内存映像空间内

sh_offset = 1f9a: 在文件偏移为 0x1f9a 处。

sh_size = 12b: 大小为 0x12b

sh_link = 0: SHN_UNDEF，即为 0

sh_info = 0: 0

sh_addralign = 0x1000000: 4K 对齐

sh_entsize = 0: 该 section 没有保存着一张固定大小入口的表，该成员就为 0。

那么，在偏移为 0x1f9a 处，大小为 0x12b 的.shstrtab 的内容是什么呢?

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x1f9a –n 299 –C hello >> .shstrtab.txt
<onlyforos>[/home/onlyforos/hello]%cat .shstrtab.txt
00001f9a    00 2e 73 79 6d 74 61 62    00 2e 73 74 72 74 61 62    |..symtab..strtab|
00001faa    00 2e 73 68 73 74 72 74    61 62 00 2e 69 6e 74 65    |..shstrtab..inte|
00001fba    72 70 00 2e 6e 6f 74 65    2e 41 42 49 2d 74 61 67    |rp..note.ABI-tag|
00001fca    00 2e 68 61 73 68 00 2e    64 79 6e 73 79 6d 00 2e    |..hash..dynsym..|
00001fda    64 79 6e 73 74 72 00 2e    67 6e 75 2e 76 65 72 73    |dynstr..gnu.vers|
00001fea    69 6f 6e 00 2e 67 6e 75    2e 76 65 72 73 69 6f 6e    |ion..gnu.version|
00001ffa    5f 72 00 2e 72 65 6c 2e    64 79 6e 00 2e 72 65 6c    |_r..rel.dyn..rel|
0000200a    2e 70 6c 74 00 2e 69 6e    69 74 00 2e 74 65 78 74    |.plt..init..text|
0000201a    00 2e 66 69 6e 69 00 2e    72 6f 64 61 74 61 00 2e    |..fini..rodata..|
0000202a    64 61 74 61 00 2e 65 68    5f 66 72 61 6d 65 00 2e    |data..eh_frame..|
0000203a    64 79 6e 61 6d 69 63 00    2e 63 74 6f 72 73 00 2e    |dynamic..ctors..|
0000204a    64 74 6f 72 73 00 2e 6a    63 72 00 2e 67 6f 74 00    |dtors..jcr..got.|
0000205a    2e 62 73 73 00 2e 63 6f    6d 6d 65 6e 74 00 2e 64    |.bss..comment..d|
0000206a    65 62 75 67 5f 61 72 61    6e 67 65 73 00 2e 64 65    |ebug_aranges..de|
0000207a    62 75 67 5f 70 75 62 6e    61 6d 65 73 00 2e 64 65    |bug_pubnames..de|
0000208a    62 75 67 5f 69 6e 66 6f    00 2e 64 65 62 75 67 5f    |bug_info..debug_|
0000209a    61 62 62 72 65 76 00 2e    64 65 62 75 67 5f 6c 69    |abbrev..debug_li|
000020aa    6e 65 00 2e 64 65 62 75    67 5f 66 72 61 6d 65 00    |ne..debug_frame.|
000020ba    2e 64 65 62 75 67 5f 73    74 72 00                   |.debug_str.|
000020c5
```

可见在字符串表中索引为 sh_name = 0x11 即偏移 0x1f9a+0x11=0x1fab 处即为.shstrtab。

2. .dynsym .symtab

一个 elf 文件通常包含两个 symbol 表，一个是.dynsym，一个是.symtab，前者表示程序运行时候需要重新定位/加载的符号(比如函数等)，后一个表示系统所有的符号列表。曾经错误地以为只有程序使用了-g 选项时，才有符号表。实际上，每个可重定位的目标文件在.symtab 中有一张符号表。当然，和编译器中的符号表不同，其不包含局部变量的表目。

.dynsym 节头表位于偏移 0x2168 处，而.symtab 节头表位于偏移 0x25c8 处。.dynsym 在偏移为 0x150 处，大小为 0x50;.symtab 在偏移为 0x2618 处，大小为 0x480。

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x2168 –n 40  –C hello
00002168  37 00 00 00 0b 00 00 00  02 00 00 00 50 81 04 08  |7...........P...|
00002178  50 01 00 00 50 00 00 00  05 00 00 00 01 00 00 00  |P...P...........|
00002188  04 00 00 00 10 00 00 00  3f 00 00 00 03 00 00 00  |........?.......|
```

sh_name = 0x37: 索引为 0x37. 0x1f9a+0x37=0x1fd1 处的字符串为：.dynsym。

sh_type = SHT_DYNSYM: 该 section 保存着动态连接的信息。

sh_flags = SHF_ALLOC: 该 section 在进程执行过程中占据着内存。

sh_addr = 0x8048150: 加载地址

sh_offset = 150: 在文件偏移为 0x150 处。

sh_size = 50: 大小为 0x50

sh_link = 5: 与第 5 个节头表关联，即.dynstr

sh_info = 1: ？One greater than the symbol table index of the last local symbol (binding STB_LOCAL).

sh_addralign = 0x4000000: 16K 对齐

sh_entsize = 10: 该 section 每个入口的表的大小为 0x10。

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x25c8 –n 40 –C hello
000025c8  01 00 00 00 02 00 00 00  00 00 00 00 00 00 00 00  |................|
000025d8  18 26 00 00 80 04 00 00  21 00 00 00 37 00 00 00  |.&......!...7...|
000025e8  04 00 00 00 10 00 00 00  09 00 00 00 03 00 00 00  |................|
```

sh_name = 0x1: 索引为 0x1. 0x1f9a+0x1=0x1f9b 处的字符串为：.symtab。

sh_type = SHT_SYMTAB: 该 section 保存着符号表的信息。

sh_flags = 0: 未明的属性设为 0。

sh_addr = 0: 不出现在进程内存映像空间内

sh_offset = 2618: 在文件偏移为 0x2618 处。

sh_size = 480: 大小为 0x480

sh_link = 21: 与第 33 个节头表关联，即.strtab

sh_info = 37: ？

sh_addralign = 0x4000000: 16K 对齐

sh_entsize = 10: 该 section 每个入口的表的大小为 0x10。

节头表描述了该节的基本信息如节名，偏移，大小等。那么位于该相应偏移处的节的具体内容是什么呢？对于符号表来说，数据结构为：

typedef struct elf32_sym{

```
   Elf32_Word      st_name;
   Elf32_Addr      st_value;
   Elf32_Word      st_size;
   unsigned char   st_info;
   unsigned char   st_other;
   Elf32_Half      st_shndx;
} Elf32_Sym;
```

---

st_name ：符号字符串表入口的索引

st_value : 相应的符号值

st_size ：符号的大小

st_info ：符号的类型和相应的属性(char type:4, binding:4) type 通常指函数或是数据，binding 通常指全局还是本地。

st_other : 0 ，没有含义。

st_shndx : 保存了相关的 section 头索引。

由前可知，对于.dynsym 节，从偏移 0x150 开始，大小为 0x50，每个入口表大小为 0x10，所以共有 5 个入口表。下面我们分析一下该节的内容：

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x150 –n 80 –C hello
00000150    00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 00    |................|
00000160    21 00 00 00 58 82 04 08    d8 00 00 00 12 00 00 00    |!...X...........|
00000170    0b 00 00 00 68 82 04 08    39 00 00 00 12 00 00 00    |....h...9.......|
00000180    12 00 00 00 94 83 04 08    04 00 00 00 11 00 0e 00    |................|
00000190    33 00 00 00 00 00 00 00    00 00 00 00 20 00 00 00    |3........... ...|
```

---

st_name :21 与.dynsym 关联和字符串表为.dynstr。.dynstr 的偏移为 0x1a0+21=0x1c1，0x1c1 偏移处的字符串为：__libc_start_main

st_value : 加载地址为 0x8048258。

st_size ：大小为 0xd8

st_info : 0

st_other : 0

st_shndx : 0x12， 与第 18 个节头表内容即.ctors 节关联。

对于其他节分析方式相同，.symtab 中有 72 个入口，在此就一一分析了。当然，对于这些内容其实不必如此费劲来分析，readelf 命令可以轻松给你答案：

```
<onlyforos>[/home/onlyforos/hello]%readelf -s hello
Symbol table '.dynsym' contains 5 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 08048258   216 FUNC    GLOBAL DEFAULT  UND
__libc_start_main@GLIBC_2.0 (2)
     2: 08048268    57 FUNC    GLOBAL DEFAULT  UND printf@GLIBC_2.0 (2)
     3: 08048394     4 OBJECT  GLOBAL DEFAULT   14 _IO_stdin_used
     4: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__

Symbol table '.symtab' contains 72 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 080480f4     0 SECTION LOCAL  DEFAULT    1
     2: 08048108     0 SECTION LOCAL  DEFAULT    2
```

```
  3: 08048128     0 SECTION LOCAL  DEFAULT     3
  4: 08048150     0 SECTION LOCAL  DEFAULT     4
  5: 080481a0     0 SECTION LOCAL  DEFAULT     5
  6: 080481ec     0 SECTION LOCAL  DEFAULT     6
  7: 080481f8     0 SECTION LOCAL  DEFAULT     7
  8: 08048218     0 SECTION LOCAL  DEFAULT     8
  9: 08048220     0 SECTION LOCAL  DEFAULT     9
 10: 08048230     0 SECTION LOCAL  DEFAULT    10
 11: 08048248     0 SECTION LOCAL  DEFAULT    11
 12: 08048278     0 SECTION LOCAL  DEFAULT    12
 13: 08048374     0 SECTION LOCAL  DEFAULT    13
 14: 08048390     0 SECTION LOCAL  DEFAULT    14
 15: 080493a8     0 SECTION LOCAL  DEFAULT    15
 16: 080493b4     0 SECTION LOCAL  DEFAULT    16
 17: 080493b8     0 SECTION LOCAL  DEFAULT    17
 18: 08049480     0 SECTION LOCAL  DEFAULT    18
 19: 08049488     0 SECTION LOCAL  DEFAULT    19
 20: 08049490     0 SECTION LOCAL  DEFAULT    20
 21: 08049494     0 SECTION LOCAL  DEFAULT    21
 22: 080494ac     0 SECTION LOCAL  DEFAULT    22
 23: 00000000     0 SECTION LOCAL  DEFAULT    23
 24: 00000000     0 SECTION LOCAL  DEFAULT    24
 25: 00000000     0 SECTION LOCAL  DEFAULT    25
 26: 00000000     0 SECTION LOCAL  DEFAULT    26
 27: 00000000     0 SECTION LOCAL  DEFAULT    27
 28: 00000000     0 SECTION LOCAL  DEFAULT    28
 29: 00000000     0 SECTION LOCAL  DEFAULT    29
 30: 00000000     0 SECTION LOCAL  DEFAULT    30
 31: 00000000     0 SECTION LOCAL  DEFAULT    31
 32: 00000000     0 SECTION LOCAL  DEFAULT    32
 33: 00000000     0 SECTION LOCAL  DEFAULT    33
 34: 00000000     0 FILE    LOCAL  DEFAULT   ABS init.c
 35: 00000000     0 FILE    LOCAL  DEFAULT   ABS initfini.c
 36: 0804829c     0 FUNC    LOCAL  DEFAULT    12 call_gmon_start
 37: 00000000     0 FILE    LOCAL  DEFAULT   ABS crtstuff.c
 38: 08049480     0 OBJECT  LOCAL  DEFAULT    18 __CTOR_LIST__
 39: 08049488     0 OBJECT  LOCAL  DEFAULT    19 __DTOR_LIST__
 40: 080493b4     0 OBJECT  LOCAL  DEFAULT    16 __EH_FRAME_BEGIN__
 41: 08049490     0 OBJECT  LOCAL  DEFAULT    20 __JCR_LIST__
 42: 080493b0     0 OBJECT  LOCAL  DEFAULT    15 p.0
 43: 080494ac     1 OBJECT  LOCAL  DEFAULT    22 completed.1
 44: 080482c0     0 FUNC    LOCAL  DEFAULT    12 __do_global_dtors_aux
 45: 080482fc     0 FUNC    LOCAL  DEFAULT    12 frame_dummy
 46: 00000000     0 FILE    LOCAL  DEFAULT   ABS crtstuff.c
 47: 08049484     0 OBJECT  LOCAL  DEFAULT    18 __CTOR_END__
 48: 0804948c     0 OBJECT  LOCAL  DEFAULT    19 __DTOR_END__
 49: 080493b4     0 OBJECT  LOCAL  DEFAULT    16 __FRAME_END__
 50: 08049490     0 OBJECT  LOCAL  DEFAULT    20 __JCR_END__
 51: 08048350     0 FUNC    LOCAL  DEFAULT    12 __do_global_ctors_aux
 52: 00000000     0 FILE    LOCAL  DEFAULT   ABS initfini.c
 53: 00000000     0 FILE    LOCAL  DEFAULT   ABS hello.c
 54: 080493ac     0 OBJECT  LOCAL  HIDDEN     15 __dso_handle
 55: 080493b8     0 OBJECT  GLOBAL DEFAULT    17 _DYNAMIC
 56: 08048390     4 OBJECT  GLOBAL DEFAULT    14 _fp_hw
 57: 08048230     0 FUNC    GLOBAL DEFAULT    10 _init
 58: 08048278     0 FUNC    GLOBAL DEFAULT    12 _start
 59: 080494ac     0 NOTYPE  GLOBAL DEFAULT   ABS __bss_start
```

```
    60: 08048328     39 FUNC    GLOBAL DEFAULT    12 main
    61:    08048258          216    FUNC                GLOBAL    DEFAULT         UND
__libc_start_main@@GLIBC_
    62: 080493a8     0 NOTYPE   WEAK    DEFAULT    15 data_start
    63: 08048268     57 FUNC    GLOBAL DEFAULT   UND printf@@GLIBC_2.0
    64: 08048374     0 FUNC     GLOBAL DEFAULT    13 _fini
    65: 080494ac     0 NOTYPE   GLOBAL DEFAULT   ABS _edata
    66:    08049494                0    OBJECT         GLOBAL    DEFAULT             21
_GLOBAL_OFFSET_TABLE_
    67: 080494b0     0 NOTYPE   GLOBAL DEFAULT   ABS _end
    68: 08048394     4 OBJECT   GLOBAL DEFAULT    14 _IO_stdin_used
    69: 080493a8     0 NOTYPE   GLOBAL DEFAULT    15 __data_start
    70: 00000000     0 NOTYPE   WEAK    DEFAULT   UND _Jv_RegisterClasses
    71: 00000000     0 NOTYPE   WEAK    DEFAULT   UND __gmon_start__
```

3．.dynamic

该节的数据结构如下：

```
typedef struct dynamic{
  Elf32_Sword d_tag;
  union{
    Elf32_Sword   d_val;
    Elf32_Addr    d_ptr;
  } d_un;
} Elf32_Dyn;
```

对每一个有该类型的 object，d_tag 控制着 d_un 的解释。d_val: 具有不同解释的整形变量；d_ptr: 程序的虚拟地址。

Name	Value	d_un	Executable	Shared Object
DT_NULL	0	ignored	mandatory	mandatory
DT_NEEDED	1	d_val	optional	optional
DT_PLTRELSZ	2	d_val	optional	optional
DT_PLTGOT	3	d_ptr	optional	optional
DT_HASH	4	d_ptr	mandatory	mandatory
DT_STRTAB	5	d_ptr	mandatory	mandatory
DT_SYMTAB	6	d_ptr	mandatory	mandatory
DT_RELA	7	d_ptr	mandatory	optional
DT_RELASZ	8	d_val	mandatory	optional
DT_RELAENT	9	d_val	mandatory	optional
DT_STRSZ	10	d_val	mandatory	mandatory
DT_SYMENT	11	d_val	mandatory	mandatory
DT_INIT	12	d_ptr	optional	optional
DT_FINI	13	d_ptr	optional	optional
DT_SONAME	14	d_val	ignored	optional
DT_RPATH	15	d_val	optional	ignored
DT_SYMBOLIC	16	ignored	ignored	optional
DT_REL	17	d_ptr	mandatory	optional
DT_RELSZ	18	d_val	mandatory	optional
DT_RELENT	19	d_val	mandatory	optional
DT_PLTREL	20	d_val	optional	optional
DT_DEBUG	21	d_ptr	optional	ignored
DT_TEXTREL	22	ignored	optional	optional

DT_JMPREL	23	d_ptr	optional	optional
DT_LOPROC	0x70000000	unspecified	unspecified	unspecified
DT_HIPROC	0x7fffffff	unspecified	unspecified	unspecified

在前面的程序头表中描述了该节的内容从偏移 0x3B8 开始，大小为 0xC8。其每项内容的大小为上面的数据结构的大小即为 8 字节。所以该节其有 0xC8/8=25 个入口。但从其内容可以看到从 0x450-0x480 的 6 个入口均为 0，所以其真正用到的入口只有 20 个(末尾 0 入口为结束)。

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x3B8 –n 200 -C hello
000003b8    01 00 00 00 01 00 00 00    0c 00 00 00 30 82 04 08    |............0...|
000003c8    0d 00 00 00 74 83 04 08    04 00 00 00 28 81 04 08    |....t.......(...|
000003d8    05 00 00 00 a0 81 04 08    06 00 00 00 50 81 04 08    |............P...|
000003e8    0a 00 00 00 4c 00 00 00    0b 00 00 00 10 00 00 00    |....L...........|
000003f8    15 00 00 00 00 00 00 00    03 00 00 00 94 94 04 08    |................|
00000408    02 00 00 00 10 00 00 00    14 00 00 00 11 00 00 00    |................|
00000418    17 00 00 00 20 82 04 08    11 00 00 00 18 82 04 08    |.... ...........|
00000428    12 00 00 00 08 00 00 00    13 00 00 00 08 00 00 00    |................|
00000438    fe ff ff 6f f8 81 04 08    ff ff ff 6f 01 00 00 00    |...o.......o....|
00000448    f0 ff ff 6f ec 81 04 08    00 00 00 00 00 00 00 00    |...o............|
00000458    00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 00    |................|
*
00000478    00 00 00 00 00 00 00 00                                |........|
```

我们就以第 1 个入口为例，分析一下. dynamic 节的内容。

d_tag = DT_NEEDED：

d_val = 1：这个元素保存着以 NULL 结尾的字符串表的偏移量，那些字符串是所需库的名字。.dynstr 节为 1 的索引，即为：libc.so.6

readelf 可以列出这 20 个. dynamic 节的入口：

```
<onlyforos>[/home/onlyforos/hello]%readelf -d hello
Dynamic segment at offset 0x3b8 contains 20 entries:
  Tag        Type                         Name/Value
 0x00000001 (NEEDED)                     Shared library: [libc.so.6]
 0x0000000c (INIT)                       0x8048230
 0x0000000d (FINI)                       0x8048374
 0x00000004 (HASH)                       0x8048128
 0x00000005 (STRTAB)                     0x80481a0
 0x00000006 (SYMTAB)                     0x8048150
 0x0000000a (STRSZ)                      76 (bytes)
 0x0000000b (SYMENT)                     16 (bytes)
 0x00000015 (DEBUG)                      0x0
 0x00000003 (PLTGOT)                     0x8049494
 0x00000002 (PLTRELSZ)                   16 (bytes)
 0x00000014 (PLTREL)                     REL
 0x00000017 (JMPREL)                     0x8048220
 0x00000011 (REL)                        0x8048218
 0x00000012 (RELSZ)                      8 (bytes)
 0x00000013 (RELENT)                     8 (bytes)
 0x6ffffffe (VERNEED)                    0x80481f8
 0x6fffffff (VERNEEDNUM)                 1
 0x6ffffff0 (VERSYM)                     0x80481ec
 0x00000000 (NULL)                       0x0
```

4. .rel.dyn .rel.plt

重定位是连接符号引用和符号定义的过程。重定位文件应当包含有如何修改他们的 section 内容的信息，从而允许可执行文件或共享目标文件为一个进程的程序映像保存正确的信息。重定位入口就是这样的数据：

```
typedef struct elf32_rel {
   Elf32_Addr        r_offset;
   Elf32_Word        r_info;
} Elf32_Rel;
```

---

r_offset ：给出了应用重定位行为的地址

r_info：该成员给出了具有受重定位影响因素的符号表索引和重定位应用的类型。该字段高 24 位为相关联符号表的索引，低 8 位为类型。类型如下：

Name	Value	Field	Calculation
R_386_NONE	0	none	none
R_386_32	1	word32	S + A
R_386_PC32	2	word32	S + A - P
R_386_GOT32	3	word32	G + A - P
R_386_PLT32	4	word32	L + A - P
R_386_COPY	5	none	none
R_386_GLOB_DAT	6	word32	S
R_386_JMP_SLOT	7	word32	S
R_386_RELATIVE	8	word32	B + A
R_386_GOTOFF	9	word32	S + A - GOT
R_386_GOTPC	10	word32	GOT + A - P

以.rel.plt 为例，从节头表中，我们可以得知其位于偏移 0x220，大小为 0x10。与索引为 4 的节头表即.dynsym 关联，因为每个入口大小为 8，所以此节共有 2 个入口。可自己去确认一下在.shstrtab 中索引为 6C 是什么？

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x2230 –n 40 -C hello
00002230    6c 00 00 00 09 00 00 00    02 00 00 00 20 82 04 08    |l..... ...|
00002240    20 02 00 00 10 00 00 00    04 00 00 00 0b 00 00 00    | ..............|
00002250    04 00 00 00 08 00 00 00                               |........|
```

跟踪至 0x220 处看看其内容：

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x220 –n 8 -C hello
00000220    a0 94 04 08 07 01 00 00    a4 94 04 08 07 02 00 00    |..............|
```

含义为：

---

r_offset = 0x80494a0: 重定位地址为：0x80494a0

r_info = 107: 类型为 7，即为 R_386_JMP_SLOT；在.dynsym 中的索引为 1，即为：__libc_start_main

r_offset = 0x80494a4: 重定位地址为：0x80494a4

r_info = 207: 类型为 7，即为 R_386_JMP_SLOT；在.dynsym 中的索引为 2，即为：printf

---

使用 readelf 查看结果：

```
<onlyforos>[/home/onlyforos/hello]%readelf -r hello
Relocation section '.rel.dyn' at offset 0x218 contains 1 entries:
```

```
 Offset     Info    Type            Sym.Value  Sym. Name
080494a8  00000406 R_386_GLOB_DAT    00000000   __gmon_start__

Relocation section '.rel.plt' at offset 0x220 contains 2 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
080494a0  00000107 R_386_JUMP_SLOT   08048258   __libc_start_main
080494a4  00000207 R_386_JUMP_SLOT   08048268   printf
```

5．.got

一般情况下，位置无关的代码不包含绝对的虚拟地址。全局偏移量表在私有数据中保存着绝对地址，所以应该使地址可用的，而不是和位置无关性和程序代码段共享能力妥协。一个程序引用它的 GOT(全局偏移量表)来使用位置无关的地址并且提取绝对的变量，所以重定位位置无关的参考到绝对的位置。

GOT 是一个数组，存在 ELF image 的数据段中，他们是一些指向 objects 的指针(通常是数据 objects).动态连接器将重新修改那些编译时还没有确定下来地址的符号的 GOT 入口。所以说 GOT 在 i386 动态连接中扮演着重要的角色。

通过简单的描述是不好理解该节的用途的，在有关链接的文档中再详细论述该节的含义。

从节头表可知，该节从偏移 0x494 处开始的 0x18 字节。先看看该节的内容：

```
<onlyforos>[/home/onlyforos/hello]%hexdump –s 0x494 –n 24 -C hello
00000494    b8 93 04 08 00 00 00 00    00 00 00 00 5e 82 04 08    |............^...|
000004a4    6e 82 04 08 00 00 00 00                               |n.......|
```

可见，GOT[0]=0x80493b8; GOT[1]=0; GOT[2]=0; GOT[3]=0x804825e; GOT[4]=0x804926e; GOT[5]=0;那么，0x804825e 及 0x804826e 是什么地址呢？查一下，原来是.plt 节中的地址。

6．.plt

PLT 是一个这样的结构，它的 entries 包含了一些代码片段用来传输控制到外部的过程。该节与.got 紧密相连。从节头表中可知，该节从偏移为 0x248 开始的 0x30 字节。那么该节的内容为：

```
<onlyforos>[/home/onlyforos/hello]%hexdump –d –j .plt hello
hello:          file format elf32-i386

Disassembly of section .plt:

08048248 <.plt>:
 8048248:       ff 35 98 94 04 08          pushl   0x8049498
 804824e:       ff 25 9c 94 04 08          jmp     *0x804949c
 8048254:       00 00                       add     %al,(%eax)
 8048256:       00 00                       add     %al,(%eax)
 8048258:       ff 25 a0 94 04 08          jmp     *0x80494a0
 804825e:       68 00 00 00 00             push    $0x0
 8048263:       e9 e0 ff ff ff             jmp     8048248 <_init+0x18>
 8048268:       ff 25 a4 94 04 08          jmp     *0x80494a4
 804826e:       68 08 00 00 00             push    $0x8
```

```
8048273:       e9 d0 ff ff ff                  jmp       8048248 <_init+0x18>
```

从中可以看到 0x804825e 及 0x804826e 这两个地址的内容。

7. All section header

使用 readelf 命令可以看到所有 section header 的内容及分布情况：

```
<onlyforos>[/home/onlyforos/hello]%readelf –S hello
There are 34 section headers, starting at offset 0x20c8:

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        080480f4 0000f4 000013 00   A  0   0  1
  [ 2] .note.ABI-tag     NOTE            08048108 000108 000020 00   A  0   0  4
  [ 3] .hash             HASH            08048128 000128 000028 04   A  4   0  4
  [ 4] .dynsym           DYNSYM          08048150 000150 000050 10   A  5   1
4
  [ 5] .dynstr           STRTAB          080481a0 0001a0 00004c 00   A  0   0  1
  [ 6] .gnu.version      VERSYM          080481ec 0001ec 00000a 02   A  4   0  2
  [ 7] .gnu.version_r    VERNEED         080481f8 0001f8 000020 00   A  5   1  4
  [ 8] .rel.dyn          REL             08048218 000218 000008 08   A  4   0  4
  [ 9] .rel.plt          REL             08048220 000220 000010 08   A  4   b  4
  [10] .init             PROGBITS        08048230 000230 000018 00  AX  0   0  4
  [11] .plt              PROGBITS        08048248 000248 000030 04  AX  0   0  4
  [12] .text             PROGBITS        08048278 000278 0000fc 00  AX  0   0  4
  [13] .fini             PROGBITS        08048374 000374 00001c 00  AX  0   0  4
  [14] .rodata           PROGBITS        08048390 000390 000015 00   A  0   0  4
  [15] .data             PROGBITS        080493a8 0003a8 00000c 00  WA  0   0  4
  [16] .eh_frame         PROGBITS        080493b4 0003b4 000004 00  WA  0   0  4
  [17] .dynamic          DYNAMIC         080493b8 0003b8 0000c8 08  WA  5   0
4
  [18] .ctors            PROGBITS        08049480 000480 000008 00  WA  0   0  4
  [19] .dtors            PROGBITS        08049488 000488 000008 00  WA  0   0  4
  [20] .jcr              PROGBITS        08049490 000490 000004 00  WA  0   0  4
  [21] .got              PROGBITS        08049494 000494 000018 04  WA  0   0  4
  [22] .bss              NOBITS          080494ac 0004ac 000004 00  WA  0   0  4
  [23] .comment          PROGBITS        00000000 0004ac 000132 00      0   0  1
  [24] .debug_aranges    PROGBITS        00000000 0005e0 000058 00      0   0  8
  [25] .debug_pubnames   PROGBITS        00000000 000638 000025 00      0   0  1
  [26] .debug_info       PROGBITS        00000000 00065d 000c85 00      0   0  1
  [27] .debug_abbrev     PROGBITS        00000000 0012e2 000127 00      0   0  1
  [28] .debug_line       PROGBITS        00000000 001409 0001f2 00      0   0  1
  [29] .debug_frame      PROGBITS        00000000 0015fc 000014 00      0   0  4
  [30] .debug_str        PROGBITS        00000000 001610 00098a 01  MS  0   0  1
  [31] .shstrtab         STRTAB          00000000 001f9a 00012b 00      0   0  1
  [32] .symtab           SYMTAB          00000000 002618 000480 10     33  37  4
  [33] .strtab           STRTAB          00000000 002a98 0001ca 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)
```

How to load the ELF

Appendix

A. hexdump –C hello

```
00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  02 00 03 00 01 00 00 00  78 82 04 08 34 00 00 00  |........x...4...|
00000020  c8 20 00 00 00 00 00 00  34 00 20 00 06 00 28 00  |. ......4. ...(.|
00000030  22 00 1f 00 06 00 00 00  34 00 00 00 34 80 04 08  |".......4...4...|
00000040  34 80 04 08 c0 00 00 00  c0 00 00 00 05 00 00 00  |4...............|
00000050  04 00 00 00 03 00 00 00  f4 00 00 00 f4 80 04 08  |................|
00000060  f4 80 04 08 13 00 00 00  13 00 00 00 04 00 00 00  |................|
00000070  01 00 00 00 01 00 00 00  00 00 00 00 00 80 04 08  |................|
00000080  00 80 04 08 a5 03 00 00  a5 03 00 00 05 00 00 00  |................|
00000090  00 10 00 00 01 00 00 00  a8 03 00 00 a8 93 04 08  |................|
000000a0  a8 93 04 08 04 01 00 00  08 01 00 00 06 00 00 00  |................|
000000b0  00 10 00 00 02 00 00 00  b8 03 00 00 b8 93 04 08  |................|
000000c0  b8 93 04 08 c8 00 00 00  c8 00 00 00 06 00 00 00  |................|
000000d0  04 00 00 00 04 00 00 00  08 01 00 00 08 81 04 08  |................|
000000e0  08 81 04 08 20 00 00 00  20 00 00 00 04 00 00 00  |.... ... .......|
000000f0  04 00 00 00 2f 6c 69 62  2f 6c 64 2d 6c 69 6e 75  |..../lib/ld-linu|
00000100  78 2e 73 6f 2e 32 00 00  04 00 00 00 10 00 00 00  |x.so.2..........|
00000110  01 00 00 00 47 4e 55 00  00 00 00 00 02 00 00 00  |....GNU.........|
00000120  02 00 00 00 05 00 00 00  03 00 00 00 05 00 00 00  |................|
00000130  04 00 00 00 01 00 00 00  03 00 00 00 00 00 00 00  |................|
00000140  00 00 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |................|
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000160  21 00 00 00 58 82 04 08  d8 00 00 00 12 00 00 00  |!...X...........|
00000170  0b 00 00 00 68 82 04 08  39 00 00 00 12 00 00 00  |....h...9.......|
00000180  12 00 00 00 94 83 04 08  04 00 00 00 11 00 0e 00  |................|
00000190  33 00 00 00 00 00 00 00  00 00 00 00 20 00 00 00  |3........... ...|
000001a0  00 6c 69 62 63 2e 73 6f  2e 36 00 70 72 69 6e 74  |.libc.so.6.print|
000001b0  66 00 5f 49 4f 5f 73 74  64 69 6e 5f 75 73 65 64  |f._IO_stdin_used|
000001c0  00 5f 5f 6c 69 62 63 5f  73 74 61 72 74 5f 6d 61  |.__libc_start_ma|
000001d0  69 6e 00 5f 5f 67 6d 6f  6e 5f 73 74 61 72 74 5f  |in.__gmon_start_|
000001e0  5f 00 47 4c 49 42 43 5f  32 2e 30 00 00 00 02 00  |_.GLIBC_2.0.....|
000001f0  02 00 01 00 00 00 00 00  01 00 01 00 01 00 00 00  |................|
00000200  10 00 00 00 00 00 00 00  10 69 69 0d 00 00 02 00  |.........ii.....|
00000210  42 00 00 00 00 00 00 00  a8 94 04 08 06 04 00 00  |B...............|
00000220  a0 94 04 08 07 01 00 00  a4 94 04 08 07 02 00 00  |................|
00000230  55 89 e5 83 ec 08 e8 61  00 00 00 90 e8 bb 00 00  |U......a........|
00000240  00 e8 0a 01 00 00 c9 c3  ff 35 98 94 04 08 ff 25  |.........5.....%|
00000250  9c 94 04 08 00 00 00 00  ff 25 a0 94 04 08 68 00  |.........%....h.|
00000260  00 00 00 e9 e0 ff ff ff  ff 25 a4 94 04 08 68 08  |.........%....h.|
00000270  00 00 00 e9 d0 ff ff ff  31 ed 5e 89 e1 83 e4 f0  |........1.^.....|
00000280  50 54 52 68 74 83 04 08  68 30 82 04 08 51 56 68  |PTRht...h0...QVh|
00000290  28 83 04 08 e8 bf ff ff  ff f4 90 90 55 89 e5 53  |(...........U..S|
000002a0  50 e8 00 00 00 00 5b 81  c3 ee 11 00 00 8b 83 14  |P.....[.........|
000002b0  00 00 00 85 c0 74 02 ff  d0 8b 5d fc c9 c3 90 90  |.....t....].....|
000002c0  55 89 e5 83 ec 08 80 3d  ac 94 04 08 00 75 29 a1  |U......=.....u).|
000002d0  b0 93 04 08 8b 10 85 d2  74 17 89 f6 83 c0 04 a3  |........t.......|
000002e0  b0 93 04 08 ff d2 a1 b0  93 04 08 8b 10 85 d2 75  |...............u|
```

```
000002f0    eb c6 05 ac 94 04 08 01    c9 c3 89 f6 55 89 e5 83    |............U...|
00000300    ec 08 a1 90 94 04 08 85    c0 74 19 b8 00 00 00 00    |.........t......|
00000310    85 c0 74 10 83 ec 0c 68    90 94 04 08 e8 df 7c fb    |..t....h......|.|
00000320    f7 83 c4 10 c9 c3 90 90    55 89 e5 83 ec 08 83 e4    |........U.......|
00000330    f0 b8 00 00 00 00 29 c4    83 ec 0c 68 98 83 04 08    |......)....h....|
00000340    e8 23 ff ff ff 83 c4 10    b8 00 00 00 00 c9 c3 90    |.#..............|
00000350    55 89 e5 53 52 a1 80 94    04 08 83 f8 ff bb 80 94    |U..SR...........|
00000360    04 08 74 0c 83 eb 04 ff    d0 8b 03 83 f8 ff 75 f4    |..t...........u.|
00000370    58 5b c9 c3 55 89 e5 53    52 e8 00 00 00 00 5b 81    |X[..U..SR.....[.|
00000380    c3 16 11 00 00 90 e8 35    ff ff ff 8b 5d fc c9 c3    |.......5....]...|
00000390    03 00 00 00 01 00 02 00    68 65 6c 6c 6f 2c 77 6f    |........hello,wo|
000003a0    72 6c 64 0a 00 00 00 00    00 00 00 00 00 00 00 00    |rld.............|
000003b0    8c 94 04 08 00 00 00 00    01 00 00 00 01 00 00 00    |................|
000003c0    0c 00 00 00 30 82 04 08    0d 00 00 00 74 83 04 08    |....0.......t...|
000003d0    04 00 00 00 28 81 04 08    05 00 00 00 a0 81 04 08    |....(...........|
000003e0    06 00 00 00 50 81 04 08    0a 00 00 00 4c 00 00 00    |....P.......L...|
000003f0    0b 00 00 00 10 00 00 00    15 00 00 00 00 00 00 00    |................|
00000400    03 00 00 00 94 94 04 08    02 00 00 00 10 00 00 00    |................|
00000410    14 00 00 00 11 00 00 00    17 00 00 00 20 82 04 08    |............ ...|
00000420    11 00 00 00 18 82 04 08    12 00 00 00 08 00 00 00    |................|
00000430    13 00 00 00 08 00 00 00    fe ff ff 6f f8 81 04 08    |...........o....|
00000440    ff ff ff 6f 01 00 00 00    f0 ff ff 6f ec 81 04 08    |...o.......o....|
00000450    00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 00    |................|
*
00000480    ff ff ff ff 00 00 00 00    ff ff ff ff 00 00 00 00    |................|
00000490    00 00 00 00 b8 93 04 08    00 00 00 00 00 00 00 00    |................|
000004a0    5e 82 04 08 6e 82 04 08    00 00 00 00 00 47 43 43    |^...n........GCC|
000004b0    3a 20 28 47 4e 55 29 20    33 2e 32 20 32 30 30 32    |: (GNU) 3.2 2002|
000004c0    30 39 30 33 20 28 52 65    64 20 48 61 74 20 4c 69    |0903 (Red Hat Li|
000004d0    6e 75 78 20 38 2e 30 20    33 2e 32 2d 37 29 00 00    |nux 8.0 3.2-7)..|
000004e0    47 43 43 3a 20 28 47 4e    55 29 20 33 2e 32 20 32    |GCC: (GNU) 3.2 2|
000004f0    30 30 32 30 39 30 33 20    28 52 65 64 20 48 61 74    |0020903 (Red Hat|
00000500    20 4c 69 6e 75 78 20 38    2e 30 20 33 2e 32 2d 37    | Linux 8.0 3.2-7|
00000510    29 00 00 47 43 43 3a 20    28 47 4e 55 29 20 33 2e    |)..GCC: (GNU) 3.|
00000520    32 20 32 30 30 32 30 39    30 33 20 28 52 65 64 20    |2 20020903 (Red |
00000530    48 61 74 20 4c 69 6e 75    78 20 38 2e 30 20 33 2e    |Hat Linux 8.0 3.|
00000540    32 2d 37 29 00 00 47 43    43 3a 20 28 47 4e 55 29    |2-7)..GCC: (GNU)|
00000550    20 33 2e 32 20 32 30 30    32 30 39 30 33 20 28 52    | 3.2 20020903 (R|
00000560    65 64 20 48 61 74 20 4c    69 6e 75 78 20 38 2e 30    |ed Hat Linux 8.0|
00000570    20 33 2e 32 2d 37 29 00    00 47 43 43 3a 20 28 47    | 3.2-7)..GCC: (G|
00000580    4e 55 29 20 33 2e 32 20    32 30 30 32 30 39 30 33    |NU) 3.2 20020903|
00000590    20 28 52 65 64 20 48 61    74 20 4c 69 6e 75 78 20    | (Red Hat Linux |
000005a0    38 2e 30 20 33 2e 32 2d    37 29 00 00 47 43 43 3a    |8.0 3.2-7)..GCC:|
000005b0    20 28 47 4e 55 29 20 33    2e 32 20 32 30 30 32 30    | (GNU) 3.2 20020|
000005c0    39 30 33 20 28 52 65 64    20 48 61 74 20 4c 69 6e    |903 (Red Hat Lin|
000005d0    75 78 20 38 2e 30 20 33    2e 32 2d 37 29 00 00 00    |ux 8.0 3.2-7)...|
000005e0    2c 00 00 00 02 00 d7 0b    00 00 04 00 00 00 00 00    |,...............|
000005f0    74 83 04 08 12 00 00 00    30 82 04 08 0c 00 00 00    |t.......0.......|
00000600    9c 82 04 08 23 00 00 00    00 00 00 00 00 00 00 00    |....#...........|
00000610    24 00 00 00 02 00 2e 0c    00 00 04 00 00 00 00 00    |$...............|
00000620    8b 83 04 08 05 00 00 00    46 82 04 08 02 00 00 00    |........F.......|
00000630    00 00 00 00 00 00 00 00    21 00 00 00 02 00 00 00    |........!.......|
00000640    00 00 d7 0b 00 00 bf 0b    00 00 5f 49 4f 5f 73 74    |..........._IO_st|
00000650    64 69 6e 5f 75 73 65 64    00 00 00 00 00 d3 0b 00    |din_used........|
00000660    00 02 00 00 00 00 00 04    01 00 00 00 00 9c 82 04    |................|
00000670    08 9c 82 04 08 6f 01 00    00 98 05 00 00 23 09 00    |.....o.......#..|
00000680    00 01 02 3c 00 00 00 08    02 4e 03 f4 00 00 00 02    |...<.....N......|
```

```
00000690  4d 3c 00 00 00 02 23 00  00 04 4c 00 00 00 53 00  |M<....#...L...S.|
000006a0  00 00 05 4c 00 00 00 01  00 06 10 04 00 00 04 07  |...L............|
000006b0  07 69 6e 74 00 04 05 08  75 00 00 00 b7 07 00 00  |.int....u.......|
000006c0  04 03 45 03 81 04 00 00  03 46 53 00 00 00 02 23  |..E......FS....#|
000006d0  00 00 08 9e 00 00 00 77  08 00 00 08 04 1b 03 9a  |.......w........|
000006e0  01 00 00 04 1c 9e 00 00  00 02 23 00 03 52 06 00  |..........#..R..|
000006f0  00 04 1d 53 00 00 00 02  23 04 00 06 d7 03 00 00  |...S....#.......|
00000700  04 05 08 30 01 00 00 87  02 00 00 24 04 2a 03 fa  |...0.......$.*..|
00000710  06 00 00 04 2b 53 00 00  00 02 23 00 03 89 08 00  |....+S....#.....|
00000720  00 04 2c 53 00 00 00 02  23 04 03 02 01 00 00 04  |..,S....#.......|
00000730  2d 5a 00 00 00 02 23 08  03 fe 02 00 00 04 2e 53  |-Z....#........S|
00000740  00 00 00 02 23 0c 03 e9  07 00 00 04 2f 53 00 00  |....#......./S..|
00000750  00 02 23 10 03 26 06 00  00 04 30 30 01 00 00 02  |..#..&....00....|
00000760  23 14 03 4e 00 00 00 04  31 53 00 00 00 02 23 18  |#..N....1S....#.|
00000770  03 12 08 00 00 04 32 37  01 00 00 02 23 1c 03 bc  |......27....#...|
00000780  00 00 00 04 33 30 01 00  00 02 23 20 00 06 5f 01  |....30....# .._.|
00000790  00 00 04 07 09 04 02 5e  01 00 00 0c 04 3c 03 93  |.......^.....<..|
000007a0  06 00 00 04 3a 75 00 00  00 02 23 00 03 4d 05 00  |....:u....#..M..|
000007b0  00 04 3b 64 01 00 00 02  23 08 00 0a 54 04 00 00  |..;d....#...T...|
000007c0  01 0b 04 5e 01 00 00 02  81 01 00 00 04 04 43 03  |...^..........C.|
000007d0  af 02 00 00 04 42 53 00  00 00 02 23 00 00 02 d0  |.....BS....#....|
000007e0  01 00 00 18 04 53 03 a1  04 00 00 04 4e 53 00 00  |.....S......NS..|
000007f0  00 02 23 00 03 9d 03 00  00 04 4f 53 00 00 00 02  |..#.......OS....|
00000800  23 04 03 40 07 00 00 04  50 64 01 00 00 02 23 08  |#..@....Pd....#.|
00000810  03 e5 08 00 00 04 51 53  00 00 00 02 23 0c 03 09  |......QS....#...|
00000820  02 00 00 04 52 75 00 00  00 02 23 10 00 02 e7 01  |....Ru....#.....|
00000830  00 00 04 04 5a 03 74 09  00 00 04 59 53 00 00 00  |....Z.t....YS...|
00000840  02 23 00 00 08 56 02 00  00 62 09 00 00 20 04 64  |.#...V...b... .d|
00000850  03 7d 02 00 00 04 65 75  00 00 00 02 23 00 03 a1  |.}....eu....#...|
00000860  00 00 00 04 66 53 00 00  00 02 23 08 03 fd 01 00  |....fS....#.....|
00000870  00 04 67 64 01 00 00 02  23 0c 03 50 09 00 00 04  |..gd....#..P....|
00000880  68 64 01 00 00 02 23 10  03 16 03 00 00 04 69 64  |hd....#.......id|
00000890  01 00 00 02 23 14 03 14  00 00 00 04 6a 53 00 00  |....#.......jS..|
000008a0  00 02 23 18 03 fd 07 00  00 04 6b 53 00 00 00 02  |..#.......kS....|
000008b0  23 1c 00 02 7b 02 00 00  08 04 74 03 3f 02 00 00  |#...{.....t.?...|
000008c0  04 72 53 00 00 00 02 23  00 03 9d 02 00 00 04 73  |.rS....#.......s|
000008d0  53 00 00 00 02 23 04 00  02 bc 02 00 00 14 04 81  |S....#..........|
000008e0  03 6c 07 00 00 04 7d 75  00 00 00 02 23 00 03 6c  |.l....}u....#..l|
000008f0  06 00 00 04 7e 53 00 00  00 02 23 08 03 1e 07 00  |....~S....#.....|
00000900  00 04 7f 53 00 00 00 02  23 0c 03 dc 07 00 00 04  |...S....#.......|
00000910  80 64 01 00 00 02 23 10  00 02 d3 02 00 00 04 04  |.d....#.........|
00000920  86 03 9d 02 00 00 04 85  53 00 00 00 02 23 00 00  |........S....#..|
00000930  0c f2 02 00 00 04 05 4e  0d 38 05 00 00 05 4c f2  |.......N.8....L.|
00000940  02 00 00 0d 25 04 00 00  05 4d f9 02 00 00 00 06  |....%....M......|
00000950  4d 04 00 00 04 07 04 09  03 00 00 09 03 00 00 05  |M...............|
00000960  4c 00 00 00 03 00 06 b7  00 00 00 01 06 02 35 03  |L.............5.|
00000970  00 00 08 05 4f 03 ed 02  00 00 05 49 53 00 00 00  |....O......IS...|
00000980  02 23 00 03 fa 00 00 00  05 4e d3 02 00 00 02 23  |.#.......N.....#|
00000990  04 00 02 5a 03 00 00 0c  06 1e 03 3c 06 00 00 06  |...Z.......<....|
000009a0  1c 5a 03 00 00 02 23 00  03 2a 08 00 00 06 1d 10  |.Z....#..*......|
000009b0  03 00 00 02 23 04 00 06  48 01 00 00 04 05 02 86  |....#...H.......|
000009c0  03 00 00 10 06 23 03 3c  06 00 00 06 21 86 03 00  |.....#.<....!...|
000009d0  00 02 23 00 03 2a 08 00  00 06 22 10 03 00 00 02  |..#..*...."".....|
000009e0  23 08 00 06 58 07 00 00  08 05 0e d2 03 00 00 04  |#...X...........|
000009f0  07 26 0f 7f 07 00 00 00  0f b7 02 00 00 01 0f ab  |.&..............|
00000a00  06 00 00 02 0f 4a 07 00  00 03 0f 84 05 00 00 04  |.....J..........|
00000a10  0f 29 03 00 00 05 0f 29  02 00 00 06 0f 7a 06 00  |.).....).....z..|
```

```
00000a20  00 07 0f 79 00 00 00 08  0f c5 07 00 00 09 00 0e  |...y............|
00000a30  e7 03 00 00 04 07 38 0f  dd 05 00 00 01 0f 61 08  |......8.......a.|
00000a40  00 00 02 00 08 3a 04 00  00 e0 03 00 00 14 07 42  |.....:.........B|
00000a50  03 23 05 00 00 07 67 f1  05 00 00 02 23 00 03 0f  |.#....g.....#...|
00000a60  05 00 00 07 68 1b 06 00  00 02 23 04 03 ed 01 00  |....h.....#.....|
00000a70  00 07 69 2d 06 00 00 02  23 08 03 ee 08 00 00 07  |..i-....#.......|
00000a80  6a 37 01 00 00 02 23 0c  03 5a 08 00 00 07 6b 33  |j7....#..Z....k3|
00000a90  06 00 00 02 23 10 00 10  6d 04 00 00 01 53 00 00  |....#...m....S..|
00000aa0  00 11 6d 04 00 00 11 44  05 00 00 11 37 01 00 00  |..m....D....7...|
00000ab0  11 c7 05 00 00 11 d9 05  00 00 11 c7 05 00 00 11  |................|
00000ac0  df 05 00 00 11 eb 05 00  00 00 0b 04 73 04 00 00  |............s...|
00000ad0  08 44 05 00 00 90 03 00  00 38 07 3f 03 3e 05 00  |.D.......8.?.>..|
00000ae0  00 07 72 3f 06 00 00 02  23 00 03 05 05 00 00 07  |..r?....#.......|
00000af0  73 45 06 00 00 02 23 04  03 d5 00 00 00 07 75 53  |sE....#.......uS|
00000b00  00 00 00 02 23 08 03 b5  08 00 00 07 77 50 06 00  |....#.......wP..|
00000b10  00 02 23 0c 03 4a 02 00  00 07 78 50 06 00 00 02  |..#..J....xP....|
00000b20  23 10 03 ee 00 00 00 07  7a 89 06 00 00 02 23 14  |#.......z.....#.|
00000b30  03 fa 04 00 00 07 7b 9f  06 00 00 02 23 18 03 54  |......{.....#..T|
00000b40  02 00 00 07 7c b1 06 00  00 02 23 1c 03 c1 08 00  |....|.....#.....|
00000b50  00 07 80 53 00 00 00 02  23 20 03 23 01 00 00 07  |...S....# .#....|
00000b60  81 53 00 00 00 02 23 24  03 a7 03 00 00 07 82 53  |.S....#$.......S|
00000b70  00 00 00 02 23 28 03 3d  03 00 00 07 83 53 00 00  |....#(.=.....S..|
00000b80  00 02 23 2c 03 57 03 00  00 07 86 53 00 00 00 02  |..#,.W.....S....|
00000b90  23 30 03 ee 08 00 00 07  88 37 01 00 00 02 23 34  |#0.......7....#4|
00000ba0  00 0b 04 4a 05 00 00 08  c7 05 00 00 d0 06 00 00  |...J............|
00000bb0  24 07 40 03 fd 03 00 00  07 8f e5 05 00 00 02 23  |$.@............#|
00000bc0  00 03 cc 04 00 00 07 90  e5 05 00 00 02 23 04 03  |.............#..|
00000bd0  c4 04 00 00 07 94 53 00  00 00 02 23 08 03 c6 02  |......S....#....|
00000be0  00 00 07 98 53 00 00 00  02 23 0c 03 9c 06 00 00  |....S....#......|
00000bf0  07 9c 53 00 00 00 02 23  10 03 66 01 00 00 07 9e  |..S....#..f.....|
00000c00  b7 06 00 00 02 23 14 03  2a 08 00 00 07 9f 10 03  |.....#..*.......|
00000c10  00 00 02 23 18 03 97 08  00 00 07 a3 33 06 00 00  |...#........3...|
00000c20  02 23 20 00 0b 04 cd 05  00 00 12 d2 05 00 00 06  |.# ............|
00000c30  ae 00 00 00 01 08 0b 04  c7 05 00 00 0b 04 e5 05  |................|
00000c40  00 00 0b 04 d2 05 00 00  0b 04 30 01 00 00 0b 04  |..........0.....|
00000c50  3a 04 00 00 10 1b 06 00  00 01 53 00 00 00 11 37  |:.........S....7|
00000c60  01 00 00 11 c7 05 00 00  11 c7 05 00 00 11 e5 05  |................|
00000c70  00 00 11 e5 05 00 00 00  0b 04 f7 05 00 00 13 2d  |...............-|
00000c80  06 00 00 01 11 37 01 00  00 00 0b 04 21 06 00 00  |.....7......!...|
00000c90  0b 04 e7 03 00 00 0a ae  01 00 00 01 0b 04 39 06  |..............9.|
00000ca0  00 00 0b 04 4b 06 00 00  12 09 03 00 00 0b 04 09  |....K...........|
00000cb0  03 00 00 10 89 06 00 00  01 53 00 00 00 11 6d 04  |.........S....m.|
00000cc0  00 00 11 44 05 00 00 11  d9 05 00 00 11 c7 05 00  |...D............|
00000cd0  00 11 df 05 00 00 11 eb  05 00 00 11 53 00 00 00  |............S...|
00000ce0  11 53 00 00 00 00 0b 04  56 06 00 00 10 9f 06 00  |.S......V.......|
00000cf0  00 01 53 00 00 00 11 6d  04 00 00 00 0b 04 8f 06  |..S....m........|
00000d00  00 00 13 b1 06 00 00 01  11 6d 04 00 00 00 0b 04  |.........m......|
00000d10  a5 06 00 00 0b 04 10 03  00 00 08 f4 06 00 00 94  |................|
00000d20  00 00 00 08 07 a9 03 0f  01 00 00 07 aa 30 01 00  |.............0..|
00000d30  00 02 23 00 03 83 01 00  00 07 ab 6d 04 00 00 02  |..#........m....|
00000d40  23 04 03 ee 08 00 00 07  ac f4 06 00 00 02 23 08  |#.............#.|
00000d50  00 04 03 07 00 00 4a 05  00 00 14 4c 00 00 00 00  |......J....L....|
00000d60  02 28 07 00 00 2c 06 34  03 98 02 00 00 06 32 bd  |.(...,.4......2.|
00000d70  06 00 00 02 23 00 03 ee  08 00 00 06 33 4a 05 00  |....#.......3J..|
00000d80  00 02 23 08 00 0c 47 07  00 00 2c 06 35 0d 98 02  |..#...G...,.5...|
00000d90  00 00 06 2f bd 06 00 00  0d d8 04 00 00 06 34 03  |.../..........4.|
00000da0  07 00 00 00 15 5f 01 00  00 08 c9 52 07 00 00 06  |....._.....R....|
```

```
00000db0  10 04 00 00 04 07 15 87  03 00 00 02 20 d2 05 00  |............ ...|
00000dc0  00 15 e3 02 00 00 02 21  6f 07 00 00 06 c4 01 00  |.......!o.......|
00000dd0  00 02 07 15 21 02 00 00  02 22 52 07 00 00 15 b7  |....!...."R.....|
00000de0  03 00 00 02 23 8c 07 00  00 06 0b 04 00 00 04 07  |....#...........|
00000df0  15 6a 04 00 00 02 25 9e  07 00 00 06 06 04 00 00  |.j....%.........|
00000e00  08 07 15 f5 02 00 00 02  26 b0 07 00 00 06 d2 03  |........&.......|
00000e10  00 00 08 05 15 76 07 00  00 02 31 c2 07 00 00 06  |.....v....1.....|
00000e20  b0 00 00 00 01 06 15 9f  08 00 00 02 32 d2 05 00  |............2...|
00000e30  00 15 80 09 00 00 02 33  df 07 00 00 06 c0 03 00  |.......3........|
00000e40  00 02 05 15 2c 04 00 00  02 34 6f 07 00 00 15 28  |....,....4o....(|
00000e50  00 00 00 02 35 53 00 00  00 15 62 03 00 00 02 36  |....5S....b....6|
00000e60  52 07 00 00 15 14 07 00  00 02 38 b0 07 00 00 15  |R.........8.....|
00000e70  5e 02 00 00 02 39 9e 07  00 00 15 32 06 00 00 02  |^....9.....2....|
00000e80  3b 28 08 00 00 0b 04 a5  07 00 00 15 ca 03 00 00  |;(..............|
00000e90  02 3d 93 07 00 00 15 1d  04 00 00 02 3e 76 07 00  |.=..........>v..|
00000ea0  00 15 0a 08 00 00 02 3f  76 07 00 00 15 a7 02 00  |.......?v.......|
00000eb0  00 02 40 81 07 00 00 15  09 06 00 00 02 41 76 07  |..@..........Av.|
00000ec0  00 00 15 a3 07 00 00 02  42 76 07 00 00 15 48 01  |........Bv....H.|
00000ed0  00 00 02 43 9e 00 00 00  15 1a 09 00 00 02 44 a5  |...C..........D.|
00000ee0  07 00 00 15 db 02 00 00  02 45 53 00 00 00 15 50  |.........ES....P|
00000ef0  01 00 00 02 46 53 00 00  00 15 2f 05 00 00 02 47  |....FS..../....G|
00000f00  81 07 00 00 15 5e 00 00  00 02 48 93 07 00 00 15  |.....^....H.....|
00000f10  76 02 00 00 02 49 76 07  00 00 15 0d 03 00 00 02  |v....Iv.........|
00000f20  4e 25 00 00 00 15 1e 00  00 00 02 51 53 00 00 00  |N%.........QS...|
00000f30  15 04 09 00 00 02 52 50  06 00 00 15 0b 00 00 00  |......RP........|
00000f40  02 53 9e 00 00 00 15 c8  00 00 00 02 54 52 07 00  |.S..........TR..|
00000f50  00 15 64 05 00 00 02 55  9e 00 00 00 15 ad 07 00  |..d....U........|
00000f60  00 02 56 9e 00 00 00 15  b8 06 00 00 02 58 9e 00  |..V..........X..|
00000f70  00 00 15 1e 08 00 00 02  5b 53 00 00 00 15 4d 03  |........[S....M.|
00000f80  00 00 02 5e 53 00 00 00  15 7f 03 00 00 02 65 53  |...^S.........eS|
00000f90  00 00 00 15 75 04 00 00  02 68 6f 07 00 00 15 5a  |....u....ho....Z|
00000fa0  01 00 00 02 6c 9e 00 00  00 15 a3 01 00 00 02 71  |....l..........q|
00000fb0  9e 00 00 00 15 ed 05 00  00 02 72 a5 07 00 00 15  |..........r.....|
00000fc0  8a 07 00 00 02 75 81 07  00 00 15 fa 05 00 00 02  |.....u..........|
00000fd0  76 93 07 00 00 15 76 01  00 00 02 79 81 07 00 00  |v.....v....y....|
00000fe0  15 8b 01 00 00 02 7a 93  07 00 00 15 72 05 00 00  |......z.....r...|
00000ff0  02 7d 93 07 00 00 15 58  07 00 00 02 80 7b 08 00  |.}.....X.....{..|
00001000  00 15 69 02 00 00 02 83  9e 00 00 00 15 32 08 00  |..i..........2..|
00001010  00 02 84 8c 07 00 00 15  59 05 00 00 02 87 53 00  |........Y.....S.|
00001020  00 00 15 08 07 00 00 02  8a 52 07 00 00 15 df 00  |.........R......|
00001030  00 00 04 23 db 09 00 00  0b 04 5e 01 00 00 15 f5  |...#......^.....|
00001040  08 00 00 04 34 a5 00 00  00 15 12 02 00 00 04 3c  |....4..........<|
00001050  39 01 00 00 15 35 01 00  00 04 43 6a 01 00 00 15  |9....5....Cj....|
00001060  c2 06 00 00 04 46 52 07  00 00 15 69 00 00 00 04  |.....FR....i....|
00001070  53 81 01 00 00 15 12 06  00 00 04 5a d0 01 00 00  |S..........Z....|
00001080  15 92 04 00 00 04 5e 53  00 00 00 15 63 09 00 00  |......^S....c...|
00001090  04 6c e7 01 00 00 15 2b  07 00 00 04 74 56 02 00  |.l.....+....tV..|
000010a0  00 15 ca 05 00 00 04 79  4f 0a 00 00 16 53 00 00  |.......yO....S..|
000010b0  00 15 6d 03 00 00 04 81  7b 02 00 00 15 37 04 00  |..m.....{....7..|
000010c0  00 04 86 bc 02 00 00 15  f3 03 00 00 04 8c 8c 07  |................|
000010d0  00 00 17 7c 05 00 00 08  26 01 9e 00 00 00 17 4d  |...|....&......M|
000010e0  04 00 00 08 41 01 52 07  00 00 15 f1 07 00 00 05  |....A.R.........|
000010f0  4f 10 03 00 00 15 62 07  00 00 06 1e 35 03 00 00  |O.....b.....5...|
00001100  15 d7 01 00 00 06 23 61  03 00 00 15 a9 08 00 00  |......#a........|
00001110  07 48 b9 0a 00 00 0b 04  56 06 00 00 15 32 00 00  |.H......V....2..|
00001120  00 07 4b ca 0a 00 00 0b  04 8f 06 00 00 15 42 06  |..K...........B.|
00001130  00 00 07 4c db 0a 00 00  0b 04 a5 06 00 00 15 d3  |...L............|
```

```
00001140  08 00 00 07 55 ec 0a 00  00 0b 04 3a 04 00 00 15  |....U......:....|
00001150  40 08 00 00 07 5a fd 0a  00 00 0b 04 f7 05 00 00  |@....Z..........|
00001160  15 e2 06 00 00 07 5e 0e  0b 00 00 0b 04 14 0b 00  |......^.........|
00001170  00 10 2e 0b 00 00 01 53  00 00 00 11 45 06 00 00  |.......S....E...|
00001180  11 2e 0b 00 00 11 eb 05  00 00 00 0b 04 34 0b 00  |.............4..|
00001190  00 0b 04 45 06 00 00 15  e3 04 00 00 07 61 45 0b  |...E.........aE.|
000011a0  00 00 0b 04 4b 0b 00 00  10 60 0b 00 00 01 53 00  |....K....`....S.|
000011b0  00 00 11 60 0b 00 00 11  45 06 00 00 00 0b 04 37  |...`....E......7|
000011c0  01 00 00 15 ae 04 00 00  07 62 71 0b 00 00 0b 04  |.........bq.....|
000011d0  21 06 00 00 15 e3 01 00  00 07 ad 82 0b 00 00 0b  |!...............|
000011e0  04 bd 06 00 00 15 43 00  00 00 06 35 28 07 00 00  |......C....5(...|
000011f0  15 18 01 00 00 06 37 df  07 00 00 15 00 00 00 00  |......7.........|
00001200  06 38 53 00 00 00 15 0e  09 00 00 06 39 6f 07 00  |.8S.........9o..|
00001210  00 15 97 07 00 00 06 3a  52 07 00 00 18 5d 06 00  |.......:R....]..|
00001220  00 01 19 d1 0b 00 00 01  05 03 94 83 04 08 12 53  |...............S|
00001230  00 00 00 00 53 00 00 00  02 00 0b 01 00 00 04 01  |....S...........|
00001240  33 01 00 00 2f 75 73 72  2f 73 72 63 2f 62 75 69  |3.../usr/src/bui|
00001250  6c 64 2f 31 34 38 36 32  30 2d 69 33 38 36 2f 42  |ld/148620-i386/B|
00001260  55 49 4c 44 2f 67 6c 69  62 63 2d 32 2e 32 2e 39  |UILD/glibc-2.2.9|
00001270  33 2f 63 73 75 00 47 4e  55 20 41 53 20 32 2e 31  |3/csu.GNU AS 2.1|
00001280  33 2e 39 30 2e 30 2e 32  00 01 80 53 00 00 00 02  |3.90.0.2...S....|
00001290  00 19 01 00 00 04 01 a4  01 00 00 2f 75 73 72 2f  |.........../usr/|
000012a0  73 72 63 2f 62 75 69 6c  64 2f 31 34 38 36 32 30  |src/build/148620|
000012b0  2d 69 33 38 36 2f 42 55  49 4c 44 2f 67 6c 69 62  |-i386/BUILD/glib|
000012c0  63 2d 32 2e 32 2e 39 33  2f 63 73 75 00 47 4e 55  |c-2.2.93/csu.GNU|
000012d0  20 41 53 20 32 2e 31 33  2e 39 30 2e 30 2e 32 00  | AS 2.13.90.0.2.|
000012e0  01 80 01 11 01 10 06 12  01 11 01 03 0e 1b 0e 25  |...............%|
000012f0  0e 13 0b 00 00 02 13 01  01 13 0b 0b 3a 0b 3b 0b  |............:.;.|
00001300  00 00 03 0d 00 03 0e 3a  0b 3b 0b 49 13 38 0a 00  |.......:.;.I.8..|
00001310  00 04 01 01 01 13 49 13  00 00 05 21 00 49 13 2f  |......I....!.I./|
00001320  0b 00 00 06 24 00 03 0e  0b 0b 3e 0b 00 00 07 24  |....$.....>....$|
00001330  00 03 08 0b 0b 3e 0b 00  00 08 13 01 01 13 03 0e  |.....>..........|
00001340  0b 0b 3a 0b 3b 0b 00 00  09 0f 00 0b 0b 00 00 0a  |..:.;...........|
00001350  13 00 03 0e 3c 0c 00 00  0b 0f 00 0b 0b 49 13 00  |....<........I..|
00001360  00 0c 17 01 01 13 0b 0b  3a 0b 3b 0b 00 00 0d 0d  |........:.;.....|
00001370  00 03 0e 3a 0b 3b 0b 49  13 00 00 0e 04 01 01 13  |...:.;.I........|
00001380  0b 0b 3a 0b 3b 0b 00 00  0f 28 00 03 0e 1c 0b 00  |..:.;....(......|
00001390  00 10 15 01 01 13 27 0c  49 13 00 00 11 05 00 49  |......'.I......I|
000013a0  13 00 00 12 26 00 49 13  00 00 13 15 01 01 13 27  |....&.I........'|
000013b0  0c 00 00 14 21 00 49 13  00 00 15 16 00 03 0e 3a  |....!.I........:|
000013c0  0b 3b 0b 49 13 00 00 16  35 00 49 13 00 00 17 16  |.;.I....5.I.....|
000013d0  00 03 0e 3a 0b 3b 05 49  13 00 00 18 34 00 03 0e  |...:.;.I....4...|
000013e0  3a 0b 3b 0b 49 13 3f 0c  02 0a 00 00 00 01 11 00  |:.;.I.?.........|
000013f0  10 06 1b 08 25 08 13 05  00 00 00 01 11 00 10 06  |....%...........|
00001400  1b 08 25 08 13 05 00 00  00 2f 01 00 00 02 00 29  |..%....../.....)|
00001410  01 00 00 01 01 fb 0e 0a  00 01 01 01 01 00 00 00  |................|
00001420  01 00 69 6e 69 74 2e 63  00 00 00 00 2e 2e 2f 73  |..init.c....../s|
00001430  79 73 64 65 70 73 2f 75  6e 69 78 2f 73 79 73 76  |ysdeps/unix/sysv|
00001440  2f 6c 69 6e 75 78 2f 62  69 74 73 2f 74 79 70 65  |/linux/bits/type|
00001450  73 2e 68 00 00 00 00 2e  2e 2f 73 79 73 64 65 70  |s.h....../sysdep|
00001460  73 2f 75 6e 69 78 2f 73  79 73 76 2f 6c 69 6e 75  |s/unix/sysv/linu|
00001470  78 2f 62 69 74 73 2f 73  63 68 65 64 2e 68 00 00  |x/bits/sched.h..|
00001480  00 00 2e 2e 2f 6c 69 6e  75 78 74 68 72 65 61 64  |..../linuxthread|
00001490  73 2f 73 79 73 64 65 70  73 2f 70 74 68 72 65 61  |s/sysdeps/pthrea|
000014a0  64 2f 62 69 74 73 2f 70  74 68 72 65 61 64 74 79  |d/bits/pthreadty|
000014b0  70 65 73 2e 68 00 00 00  00 2e 2e 2f 77 63 73 6d  |pes.h....../wcsm|
000014c0  62 73 2f 77 63 68 61 72  2e 68 00 00 00 00 2e 2e  |bs/wchar.h......|
```

```
000014d0  2f 73 79 73 64 65 70 73  2f 67 6e 75 2f 5f 47 5f  |/sysdeps/gnu/_G_|
000014e0  63 6f 6e 66 69 67 2e 68  00 00 00 00 2e 2e 2f 69  |config.h....../i|
000014f0  63 6f 6e 76 2f 67 63 6f  6e 76 2e 68 00 00 00 00  |conv/gconv.h....|
00001500  2f 75 73 72 2f 6c 69 62  2f 67 63 63 2d 6c 69 62  |/usr/lib/gcc-lib|
00001510  2f 69 33 38 36 2d 72 65  64 68 61 74 2d 6c 69 6e  |/i386-redhat-lin|
00001520  75 78 2f 33 2e 32 2f 69  6e 63 6c 75 64 65 2f 73  |ux/3.2/include/s|
00001530  74 64 64 65 66 2e 68 00  00 00 00 00 6d 00 00 00  |tddef.h.....m...|
00001540  02 00 23 00 00 00 01 01  fb 0e 0a 00 01 01 01 01  |..#.............|
00001550  00 00 00 01 00 2f 74 6d  70 2f 63 63 6e 68 46 63  |...../tmp/ccnhFc|
00001560  69 6c 2e 73 00 00 00 00  00 00 05 02 74 83 04 08  |il.s........t...|
00001570  03 c4 00 01 1e 2c 1e 1e  57 1e 02 07 00 01 01 00  |.....,..W.......|
00001580  05 02 30 82 04 08 03 33  01 1e 2c 3a 02 06 00 01  |..0....3..,:....|
00001590  01 00 05 02 9c 82 04 08  03 1b 01 1e 2c 1e 1e 57  |............,..W|
000015a0  1e 64 64 2c 2c 2d 3a 1e  02 02 00 01 01 4a 00 00  |.dd,,-:......J..|
000015b0  00 02 00 23 00 00 00 01  01 fb 0e 0a 00 01 01 01  |...#............|
000015c0  01 00 00 00 01 00 2f 74  6d 70 2f 63 63 71 66 4a  |....../tmp/ccqfJ|
000015d0  4e 56 6b 2e 73 00 00 00  00 00 00 05 02 8b 83 04  |NVk.s...........|
000015e0  08 03 21 01 3a 1e 02 01  00 01 01 00 05 02 46 82  |..!.:.........F.|
000015f0  04 08 03 18 01 1e 02 01  00 01 01 00 10 00 00 00  |................|
00001600  ff ff ff ff 01 00 01 7c  08 0c 04 04 88 01 00 00  |.......|........|
00001610  5f 47 5f 69 6e 74 33 32  5f 74 00 5f 5f 74 69 6d  |_G_int32_t.__tim|
00001620  65 5f 74 00 5f 5f 72 77  5f 6b 69 6e 64 00 5f 5f  |e_t.__rw_kind.__|
00001630  64 61 64 64 72 5f 74 00  5f 5f 69 6e 74 33 32 5f  |daddr_t.__int32_|
00001640  74 00 5f 5f 67 63 6f 6e  76 5f 69 6e 69 74 5f 66  |t.__gconv_init_f|
00001650  63 74 00 5f 47 5f 69 63  6f 6e 76 5f 74 00 5f 5f  |ct._G_iconv_t.__|
00001660  73 74 61 63 6b 61 64 64  72 5f 73 65 74 00 5f 5f  |stackaddr_set.__|
00001670  72 6c 69 6d 36 34 5f 74  00 70 74 68 72 65 61 64  |rlim64_t.pthread|
00001680  5f 6d 75 74 65 78 5f 74  00 5f 5f 47 43 4f 4e 56  |_mutex_t.__GCONV|
00001690  5f 49 4c 4c 45 47 41 4c  5f 44 45 53 43 52 49 50  |_ILLEGAL_DESCRIP|
000016a0  54 4f 52 00 5f 5f 67 63  6f 6e 76 5f 69 6e 66 6f  |TOR.__gconv_info|
000016b0  00 5f 5f 72 77 5f 72 65  61 64 65 72 73 00 75 6e  |.__rw_readers.un|
000016c0  73 69 67 6e 65 64 20 63  68 61 72 00 5f 5f 73 74  |signed char.__st|
000016d0  61 63 6b 73 69 7a 65 00  5f 5f 75 73 65 63 6f 6e  |acksize.__usecon|
000016e0  64 73 5f 74 00 5f 5f 63  6f 75 6e 74 65 72 00 5f  |ds_t.__counter._|
000016f0  70 74 68 72 65 61 64 5f  64 65 73 63 72 00 5f 5f  |pthread_descr.__|
00001700  66 63 74 00 5f 5f 76 61  6c 00 5f 5f 76 61 6c 75  |fct.__val.__valu|
00001710  65 00 5f 5f 73 63 68 65  64 70 61 72 61 6d 00 5f  |e.__schedparam._|
00001720  5f 6e 73 74 65 70 73 00  5f 47 5f 69 6e 74 31 36  |_nsteps._G_int16|
00001730  5f 74 00 5f 5f 6d 61 78  5f 6e 65 65 64 65 64 5f  |_t.__max_needed_|
00001740  66 72 6f 6d 00 70 74 68  72 65 61 64 5f 63 6f 6e  |from.pthread_con|
00001750  64 61 74 74 72 5f 74 00  5f 5f 6f 66 66 5f 74 00  |dattr_t.__off_t.|
00001760  5f 5f 73 73 69 7a 65 5f  74 00 5f 5f 62 6c 6b 73  |__ssize_t.__blks|
00001770  69 7a 65 5f 74 00 5f 5f  73 74 61 74 65 70 00 69  |ize_t.__statep.i|
00001780  6e 69 74 2e 63 00 5f 5f  66 73 66 69 6c 63 6e 74  |nit.c.__fsfilcnt|
00001790  5f 74 00 5f 5f 73 74 65  70 73 00 5f 5f 66 73 66  |_t.__steps.__fsf|
000017a0  69 6c 63 6e 74 36 34 5f  74 00 5f 5f 73 74 61 74  |ilcnt64_t.__stat|
000017b0  75 73 00 5f 5f 62 6c 6b  63 6e 74 5f 74 00 5f 5f  |us.__blkcnt_t.__|
000017c0  67 63 6f 6e 76 5f 6c 6f  61 64 65 64 5f 6f 62 6a  |gconv_loaded_obj|
000017d0  65 63 74 00 73 68 6f 72  74 20 75 6e 73 69 67 6e  |ect.short unsign|
000017e0  65 64 20 69 6e 74 00 5f  47 5f 66 70 6f 73 36 34  |ed int._G_fpos64|
000017f0  5f 74 00 5f 5f 67 63 6f  6e 76 5f 74 00 5f 5f 74  |_t.__gconv_t.__t|
00001800  72 61 6e 73 5f 65 6e 64  5f 66 63 74 00 5f 5f 72  |rans_end_fct.__r|
00001810  77 5f 77 72 69 74 65 72  00 5f 5f 6d 5f 6c 6f 63  |w_writer.__m_loc|
00001820  6b 00 70 74 68 72 65 61  64 5f 63 6f 6e 64 5f 74  |k.pthread_cond_t|
00001830  00 5f 5f 75 5f 69 6e 74  00 5f 5f 47 43 4f 4e 56  |.__u_int.__GCONV|
00001840  5f 49 4c 4c 45 47 41 4c  5f 49 4e 50 55 54 00 5f  |_ILLEGAL_INPUT._|
00001850  5f 6c 6f 63 6b 6b 69 6e  64 00 5f 5f 74 6f 5f 6e  |_lockkind.__to_n|
```

```
00001860  61 6d 65 00 5f 5f 65 6e  64 5f 66 63 74 00 5f 5f  |ame.__end_fct.__|
00001870  75 69 6e 74 36 34 5f 74  00 5f 5f 74 5f 73 63 61  |uint64_t.__t_sca|
00001880  6c 61 72 5f 74 00 5f 5f  69 64 5f 74 00 5f 5f 72  |lar_t.__id_t.__r|
00001890  77 5f 6c 6f 63 6b 00 5f  5f 70 74 68 72 65 61 64  |w_lock.__pthread|
000018a0  5f 61 74 74 72 5f 73 00  5f 5f 63 64 00 5f 5f 70  |_attr_s.__cd.__p|
000018b0  73 68 61 72 65 64 00 5f  5f 69 6e 6f 5f 74 00 5f  |shared.__ino_t._|
000018c0  5f 64 75 6d 6d 79 00 5f  5f 47 43 4f 4e 56 5f 4e  |_dummy.__GCONV_N|
000018d0  4f 43 4f 4e 56 00 5f 5f  69 6e 76 6f 63 61 74 69  |OCONV.__invocati|
000018e0  6f 6e 5f 63 6f 75 6e 74  65 72 00 5f 5f 70 69 64  |on_counter.__pid|
000018f0  5f 74 00 5f 5f 75 5f 73  68 6f 72 74 00 5f 5f 63  |_t.__u_short.__c|
00001900  6f 75 6e 74 00 5f 5f 71  75 61 64 5f 74 00 5f 5f  |ount.__quad_t.__|
00001910  69 6e 68 65 72 69 74 73  63 68 65 64 00 5f 5f 66  |inheritsched.__f|
00001920  73 69 64 5f 74 00 5f 5f  72 77 5f 77 72 69 74 65  |sid_t.__rw_write|
00001930  5f 77 61 69 74 69 6e 67  00 5f 5f 47 43 4f 4e 56  |_waiting.__GCONV|
00001940  5f 46 55 4c 4c 5f 4f 55  54 50 55 54 00 5f 5f 6d  |_FULL_OUTPUT.__m|
00001950  61 78 5f 6e 65 65 64 65  64 5f 74 6f 00 5f 5f 74  |ax_needed_to.__t|
00001960  69 6d 65 72 5f 74 00 5f  5f 73 74 61 74 65 66 75  |imer_t.__statefu|
00001970  6c 00 5f 5f 75 69 6e 74  33 32 5f 74 00 70 74 68  |l.__uint32_t.pth|
00001980  72 65 61 64 5f 62 61 72  72 69 65 72 5f 74 00 5f  |read_barrier_t._|
00001990  5f 6b 65 79 5f 74 00 5f  5f 75 5f 63 68 61 72 00  |_key_t.__u_char.|
000019a0  5f 5f 67 63 6f 6e 76 5f  73 74 65 70 00 5f 5f 6d  |__gconv_step.__m|
000019b0  5f 63 6f 75 6e 74 00 5f  5f 6d 69 6e 5f 6e 65 65  |_count.__min_nee|
000019c0  64 65 64 5f 74 6f 00 5f  5f 75 5f 6c 6f 6e 67 00  |ded_to.__u_long.|
000019d0  73 68 6f 72 74 20 69 6e  74 00 5f 5f 64 65 76 5f  |short int.__dev_|
000019e0  74 00 6c 6f 6e 67 20 6c  6f 6e 67 20 69 6e 74 00  |t.long long int.|
000019f0  5f 5f 67 63 6f 6e 76 5f  74 72 61 6e 73 5f 64 61  |__gconv_trans_da|
00001a00  74 61 00 70 74 68 72 65  61 64 5f 74 00 5f 5f 6f  |ta.pthread_t.__o|
00001a10  75 74 62 75 66 00 6c 6f  6e 67 20 6c 6f 6e 67 20  |utbuf.long long |
00001a20  75 6e 73 69 67 6e 65 64  20 69 6e 74 00 5f 5f 75  |unsigned int.__u|
00001a30  69 64 5f 74 00 5f 5f 77  63 68 62 00 5f 5f 75 69  |id_t.__wchb.__ui|
00001a40  6e 74 31 36 5f 74 00 70  74 68 72 65 61 64 5f 62  |nt16_t.pthread_b|
00001a50  61 72 72 69 65 72 61 74  74 72 5f 74 00 77 69 6e  |arrierattr_t.win|
00001a60  74 5f 74 00 5f 70 74 68  72 65 61 64 5f 64 65 73  |t_t._pthread_des|
00001a70  63 72 5f 73 74 72 75 63  74 00 5f 5f 75 5f 71 75  |cr_struct.__u_qu|
00001a80  61 64 5f 74 00 5f 5f 69  70 63 5f 70 69 64 5f 74  |ad_t.__ipc_pid_t|
00001a90  00 5f 5f 73 63 68 65 64  5f 70 72 69 6f 72 69 74  |.__sched_priorit|
00001aa0  79 00 70 74 68 72 65 61  64 5f 6f 6e 63 65 5f 74  |y.pthread_once_t|
00001ab0  00 5f 5f 6d 5f 72 65 73  65 72 76 65 64 00 5f 5f  |.__m_reserved.__|
00001ac0  67 63 6f 6e 76 5f 74 72  61 6e 73 5f 65 6e 64 5f  |gconv_trans_end_|
00001ad0  66 63 74 00 5f 5f 66 6c  61 67 73 00 5f 5f 6f 75  |fct.__flags.__ou|
00001ae0  74 62 75 66 65 6e 64 00  5f 5f 63 6f 6d 62 69 6e  |tbufend.__combin|
00001af0  65 64 00 5f 5f 67 63 6f  6e 76 5f 74 72 61 6e 73  |ed.__gconv_trans|
00001b00  5f 69 6e 69 74 5f 66 63  74 00 5f 5f 69 6e 69 74  |_init_fct.__init|
00001b10  5f 66 63 74 00 5f 5f 6d  6f 64 6e 61 6d 65 00 5f  |_fct.__modname._|
00001b20  5f 74 72 61 6e 73 5f 63  6f 6e 74 65 78 74 5f 66  |_trans_context_f|
00001b30  63 74 00 5f 5f 74 72 61  6e 73 5f 66 63 74 00 5f  |ct.__trans_fct._|
00001b40  5f 72 6c 69 6d 5f 74 00  5f 5f 77 63 68 00 5f 5f  |_rlim_t.__wch.__|
00001b50  73 68 6c 69 62 5f 68 61  6e 64 6c 65 00 5f 5f 63  |shlib_handle.__c|
00001b60  5f 77 61 69 74 69 6e 67  00 5f 5f 69 6e 74 70 74  |_waiting.__intpt|
00001b70  72 5f 74 00 5f 5f 73 75  73 65 63 6f 6e 64 73 5f  |r_t.__suseconds_|
00001b80  74 00 5f 5f 69 6e 6f 36  34 5f 74 00 77 63 68 61  |t.__ino64_t.wcha|
00001b90  72 5f 74 00 5f 5f 47 43  4f 4e 56 5f 45 4d 50 54  |r_t.__GCONV_EMPT|
00001ba0  59 5f 49 4e 50 55 54 00  2f 75 73 72 2f 73 72 63  |Y_INPUT./usr/src|
00001bb0  2f 62 75 69 6c 64 2f 31  34 38 36 32 30 2d 69 33  |/build/148620-i3|
00001bc0  38 36 2f 42 55 49 4c 44  2f 67 6c 69 62 63 2d 32  |86/BUILD/glibc-2|
00001bd0  2e 32 2e 39 33 2f 63 73  75 00 70 74 68 72 65 61  |.2.93/csu.pthrea|
00001be0  64 5f 73 70 69 6e 6c 6f  63 6b 5f 74 00 5f 5f 47  |d_spinlock_t.__G|
```

```
00001bf0  43 4f 4e 56 5f 49 53 5f  4c 41 53 54 00 5f 5f 62  |CONV_IS_LAST.__b|
00001c00  6c 6b 63 6e 74 36 34 5f  74 00 5f 5f 66 73 62 6c  |lkcnt64_t.__fsbl|
00001c10  6b 63 6e 74 36 34 5f 74  00 5f 5f 6d 6f 64 65 5f  |kcnt64_t.__mode_|
00001c20  74 00 70 74 68 72 65 61  64 5f 6d 75 74 65 78 61  |t.pthread_mutexa|
00001c30  74 74 72 5f 74 00 5f 5f  67 75 61 72 64 73 69 7a  |ttr_t.__guardsiz|
00001c40  65 00 5f 5f 71 61 64 64  72 5f 74 00 5f 5f 70 6f  |e.__qaddr_t.__po|
00001c50  73 00 5f 5f 67 63 6f 6e  76 5f 65 6e 64 5f 66 63  |s.__gconv_end_fc|
00001c60  74 00 5f 5f 73 70 69 6e  6c 6f 63 6b 00 5f 49 4f  |t.__spinlock._IO|
00001c70  5f 73 74 64 69 6e 5f 75  73 65 64 00 5f 5f 62 61  |_stdin_used.__ba|
00001c80  5f 72 65 71 75 69 72 65  64 00 5f 5f 47 43 4f 4e  |_required.__GCON|
00001c90  56 5f 49 4e 43 4f 4d 50  4c 45 54 45 5f 49 4e 50  |V_INCOMPLETE_INP|
00001ca0  55 54 00 5f 5f 63 5f 6c  6f 63 6b 00 5f 5f 69 6e  |UT.__c_lock.__in|
00001cb0  74 65 72 6e 61 6c 5f 75  73 65 00 5f 5f 47 43 4f  |ternal_use.__GCO|
00001cc0  4e 56 5f 4e 4f 44 42 00  5f 5f 63 6c 6f 63 6b 5f  |NV_NODB.__clock_|
00001cd0  74 00 70 74 68 72 65 61  64 5f 6b 65 79 5f 74 00  |t.pthread_key_t.|
00001ce0  5f 5f 67 63 6f 6e 76 5f  73 74 65 70 5f 64 61 74  |__gconv_step_dat|
00001cf0  61 00 5f 5f 67 63 6f 6e  76 5f 74 72 61 6e 73 5f  |a.__gconv_trans_|
00001d00  71 75 65 72 79 5f 66 63  74 00 5f 5f 64 65 74 61  |query_fct.__deta|
00001d10  63 68 73 74 61 74 65 00  5f 5f 73 6f 63 6b 6c 65  |chstate.__sockle|
00001d20  6e 5f 74 00 5f 5f 69 6e  74 36 34 5f 74 00 5f 5f  |n_t.__int64_t.__|
00001d30  62 61 5f 70 72 65 73 65  6e 74 00 70 74 68 72 65  |ba_present.pthre|
00001d40  61 64 5f 72 77 6c 6f 63  6b 61 74 74 72 5f 74 00  |ad_rwlockattr_t.|
00001d50  5f 5f 6d 5f 6f 77 6e 65  72 00 5f 5f 47 43 4f 4e  |__m_owner.__GCON|
00001d60  56 5f 4e 4f 4d 45 4d 00  5f 5f 6f 66 66 36 34 5f  |V_NOMEM.__off64_|
00001d70  74 00 5f 47 5f 66 70 6f  73 5f 74 00 5f 5f 62 61  |t._G_fpos_t.__ba|
00001d80  5f 6c 6f 63 6b 00 5f 5f  69 6e 74 38 5f 74 00 5f  |_lock.__int8_t._|
00001d90  5f 47 43 4f 4e 56 5f 4f  4b 00 5f 5f 66 73 62 6c  |_GCONV_OK.__fsbl|
00001da0  6b 63 6e 74 5f 74 00 5f  47 5f 75 69 6e 74 33 32  |kcnt_t._G_uint32|
00001db0  5f 74 00 5f 5f 6e 6c 69  6e 6b 5f 74 00 5f 5f 73  |_t.__nlink_t.__s|
00001dc0  77 62 6c 6b 5f 74 00 5f  5f 73 63 68 65 64 5f 70  |wblk_t.__sched_p|
00001dd0  61 72 61 6d 00 5f 5f 47  43 4f 4e 56 5f 49 4e 54  |aram.__GCONV_INT|
00001de0  45 52 4e 41 4c 5f 45 52  52 4f 52 00 5f 5f 62 61  |ERNAL_ERROR.__ba|
00001df0  5f 77 61 69 74 69 6e 67  00 5f 5f 73 63 6f 70 65  |_waiting.__scope|
00001e00  00 5f 5f 6d 62 73 74 61  74 65 5f 74 00 5f 5f 72  |.__mbstate_t.__r|
00001e10  77 5f 70 73 68 61 72 65  64 00 5f 5f 67 69 64 5f  |w_pshared.__gid_|
00001e20  74 00 5f 5f 73 74 61 63  6b 61 64 64 72 00 5f 5f  |t.__stackaddr.__|
00001e30  63 6c 6f 63 6b 69 64 5f  74 00 5f 5f 73 74 61 74  |clockid_t.__stat|
00001e40  65 00 5f 5f 74 5f 75 73  63 61 6c 61 72 5f 74 00  |e.__t_uscalar_t.|
00001e50  5f 5f 67 63 6f 6e 76 5f  74 72 61 6e 73 5f 63 6f  |__gconv_trans_co|
00001e60  6e 74 65 78 74 5f 66 63  74 00 5f 5f 6e 65 78 74  |ntext_fct.__next|
00001e70  00 5f 5f 47 43 4f 4e 56  5f 49 47 4e 4f 52 45 5f  |.__GCONV_IGNORE_|
00001e80  45 52 52 4f 52 53 00 5f  70 74 68 72 65 61 64 5f  |ERRORS._pthread_|
00001e90  66 61 73 74 6c 6f 63 6b  00 5f 5f 73 63 68 65 64  |fastlock.__sched|
00001ea0  70 6f 6c 69 63 79 00 5f  5f 74 72 61 6e 73 00 5f  |policy.__trans._|
00001eb0  5f 75 69 6e 74 38 5f 74  00 5f 5f 67 63 6f 6e 76  |_uint8_t.__gconv|
00001ec0  5f 66 63 74 00 5f 5f 66  72 6f 6d 5f 6e 61 6d 65  |_fct.__from_name|
00001ed0  00 5f 5f 6d 69 6e 5f 6e  65 65 64 65 64 5f 66 72  |.__min_needed_fr|
00001ee0  6f 6d 00 5f 5f 67 63 6f  6e 76 5f 74 72 61 6e 73  |om.__gconv_trans|
00001ef0  5f 66 63 74 00 5f 5f 6d  5f 6b 69 6e 64 00 5f 5f  |_fct.__m_kind.__|
00001f00  64 61 74 61 00 70 74 68  72 65 61 64 5f 61 74 74  |data.pthread_att|
00001f10  72 5f 74 00 5f 5f 63 61  64 64 72 5f 74 00 5f 47  |r_t.__caddr_t._G|
00001f20  5f 75 69 6e 74 31 36 5f  74 00 5f 5f 6c 6f 66 66  |_uint16_t.__loff|
00001f30  5f 74 00 47 4e 55 20 43  20 33 2e 32 20 32 30 30  |_t.GNU C 3.2 200|
00001f40  32 30 39 30 33 20 28 52  65 64 20 48 61 74 20 4c  |20903 (Red Hat L|
00001f50  69 6e 75 78 20 38 2e 30  20 33 2e 32 2d 37 29 00  |inux 8.0 3.2-7).|
00001f60  5f 5f 72 77 5f 72 65 61  64 5f 77 61 69 74 69 6e  |__rw_read_waitin|
00001f70  67 00 5f 70 74 68 72 65  61 64 5f 72 77 6c 6f 63  |g._pthread_rwloc|
```

```
00001f80    6b 5f 74 00 5f 5f 6d 75    74 65 78 6b 69 6e 64 00    |k_t.__mutexkind.|
00001f90    5f 5f 69 6e 74 31 36 5f    74 00 00 2e 73 79 6d 74    |__int16_t...symt|
00001fa0    61 62 00 2e 73 74 72 74    61 62 00 2e 73 68 73 74    |ab..strtab..shst|
00001fb0    72 74 61 62 00 2e 69 6e    74 65 72 70 00 2e 6e 6f    |rtab..interp..no|
00001fc0    74 65 2e 41 42 49 2d 74    61 67 00 2e 68 61 73 68    |te.ABI-tag..hash|
00001fd0    00 2e 64 79 6e 73 79 6d    00 2e 64 79 6e 73 74 72    |..dynsym..dynstr|
00001fe0    00 2e 67 6e 75 2e 76 65    72 73 69 6f 6e 00 2e 67    |..gnu.version..g|
00001ff0    6e 75 2e 76 65 72 73 69    6f 6e 5f 72 00 2e 72 65    |nu.version_r..re|
00002000    6c 2e 64 79 6e 00 2e 72    65 6c 2e 70 6c 74 00 2e    |l.dyn..rel.plt..|
00002010    69 6e 69 74 00 2e 74 65    78 74 00 2e 66 69 6e 69    |init..text..fini|
00002020    00 2e 72 6f 64 61 74 61    00 2e 64 61 74 61 00 2e    |..rodata..data..|
00002030    65 68 5f 66 72 61 6d 65    00 2e 64 79 6e 61 6d 69    |eh_frame..dynami|
00002040    63 00 2e 63 74 6f 72 73    00 2e 64 74 6f 72 73 00    |c..ctors..dtors.|
00002050    2e 6a 63 72 00 2e 67 6f    74 00 2e 62 73 73 00 2e    |.jcr..got..bss..|
00002060    63 6f 6d 6d 65 6e 74 00    2e 64 65 62 75 67 5f 61    |comment..debug_a|
00002070    72 61 6e 67 65 73 00 2e    64 65 62 75 67 5f 70 75    |ranges..debug_pu|
00002080    62 6e 61 6d 65 73 00 2e    64 65 62 75 67 5f 69 6e    |bnames..debug_in|
00002090    66 6f 00 2e 64 65 62 75    67 5f 61 62 62 72 65 76    |fo..debug_abbrev|
000020a0    00 2e 64 65 62 75 67 5f    6c 69 6e 65 00 2e 64 65    |..debug_line..de|
000020b0    62 75 67 5f 66 72 61 6d    65 00 2e 64 65 62 75 67    |bug_frame..debug|
000020c0    5f 73 74 72 00 00 00 00    00 00 00 00 00 00 00 00    |_str............|
000020d0    00 00 00 00 00 00 00 00    00 00 00 00 00 00 00 00    |................|
*
000020f0    1b 00 00 00 01 00 00 00    02 00 00 00 f4 80 04 08    |................|
00002100    f4 00 00 00 13 00 00 00    00 00 00 00 00 00 00 00    |................|
00002110    01 00 00 00 00 00 00 00    23 00 00 00 07 00 00 00    |........#.......|
00002120    02 00 00 00 08 81 04 08    08 01 00 00 20 00 00 00    |............ ...|
00002130    00 00 00 00 00 00 00 00    04 00 00 00 00 00 00 00    |................|
00002140    31 00 00 00 05 00 00 00    02 00 00 00 28 81 04 08    |1...........(...|
00002150    28 01 00 00 28 00 00 00    04 00 00 00 00 00 00 00    |(...(...........|
00002160    04 00 00 00 04 00 00 00    37 00 00 00 0b 00 00 00    |........7.......|
00002170    02 00 00 00 50 81 04 08    50 01 00 00 50 00 00 00    |....P...P...P...|
00002180    05 00 00 00 01 00 00 00    04 00 00 00 10 00 00 00    |................|
00002190    3f 00 00 00 03 00 00 00    02 00 00 00 a0 81 04 08    |?...............|
000021a0    a0 01 00 00 4c 00 00 00    00 00 00 00 00 00 00 00    |....L...........|
000021b0    01 00 00 00 00 00 00 00    47 00 00 00 ff ff ff 6f    |........G......o|
000021c0    02 00 00 00 ec 81 04 08    ec 01 00 00 0a 00 00 00    |................|
000021d0    04 00 00 00 00 00 00 00    02 00 00 00 02 00 00 00    |................|
000021e0    54 00 00 00 fe ff ff 6f    02 00 00 00 f8 81 04 08    |T......o........|
000021f0    f8 01 00 00 20 00 00 00    05 00 00 00 01 00 00 00    |.... ...........|
00002200    04 00 00 00 00 00 00 00    63 00 00 00 09 00 00 00    |........c.......|
00002210    02 00 00 00 18 82 04 08    18 02 00 00 08 00 00 00    |................|
00002220    04 00 00 00 00 00 00 00    04 00 00 00 08 00 00 00    |................|
00002230    6c 00 00 00 09 00 00 00    02 00 00 00 20 82 04 08    |l........... ...|
00002240    20 02 00 00 10 00 00 00    04 00 00 00 0b 00 00 00    | ...............|
00002250    04 00 00 00 08 00 00 00    75 00 00 00 01 00 00 00    |........u.......|
00002260    06 00 00 00 30 82 04 08    30 02 00 00 18 00 00 00    |....0...0.......|
00002270    00 00 00 00 00 00 00 00    04 00 00 00 00 00 00 00    |................|
00002280    70 00 00 00 01 00 00 00    06 00 00 00 48 82 04 08    |p...........H...|
00002290    48 02 00 00 30 00 00 00    00 00 00 00 00 00 00 00    |H...0...........|
000022a0    04 00 00 00 04 00 00 00    7b 00 00 00 01 00 00 00    |........{.......|
000022b0    06 00 00 00 78 82 04 08    78 02 00 00 fc 00 00 00    |....x...x.......|
000022c0    00 00 00 00 00 00 00 00    04 00 00 00 00 00 00 00    |................|
000022d0    81 00 00 00 01 00 00 00    06 00 00 00 74 83 04 08    |............t...|
000022e0    74 03 00 00 1c 00 00 00    00 00 00 00 00 00 00 00    |t...............|
000022f0    04 00 00 00 00 00 00 00    87 00 00 00 01 00 00 00    |................|
00002300    02 00 00 00 90 83 04 08    90 03 00 00 15 00 00 00    |................|
```

```
00002310  00 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  |................|
00002320  8f 00 00 00 01 00 00 00  03 00 00 00 a8 93 04 08  |................|
00002330  a8 03 00 00 0c 00 00 00  00 00 00 00 00 00 00 00  |................|
00002340  04 00 00 00 00 00 00 00  95 00 00 00 01 00 00 00  |................|
00002350  03 00 00 00 b4 93 04 08  b4 03 00 00 04 00 00 00  |................|
00002360  00 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  |................|
00002370  9f 00 00 00 06 00 00 00  03 00 00 00 b8 93 04 08  |................|
00002380  b8 03 00 00 c8 00 00 00  05 00 00 00 00 00 00 00  |................|
00002390  04 00 00 00 08 00 00 00  a8 00 00 00 01 00 00 00  |................|
000023a0  03 00 00 00 80 94 04 08  80 04 00 00 08 00 00 00  |................|
000023b0  00 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  |................|
000023c0  af 00 00 00 01 00 00 00  03 00 00 00 88 94 04 08  |................|
000023d0  88 04 00 00 08 00 00 00  00 00 00 00 00 00 00 00  |................|
000023e0  04 00 00 00 00 00 00 00  b6 00 00 00 01 00 00 00  |................|
000023f0  03 00 00 00 90 94 04 08  90 04 00 00 04 00 00 00  |................|
00002400  00 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  |................|
00002410  bb 00 00 00 01 00 00 00  03 00 00 00 94 94 04 08  |................|
00002420  94 04 00 00 18 00 00 00  00 00 00 00 00 00 00 00  |................|
00002430  04 00 00 00 04 00 00 00  c0 00 00 00 08 00 00 00  |................|
00002440  03 00 00 00 ac 94 04 08  ac 04 00 00 04 00 00 00  |................|
00002450  00 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00  |................|
00002460  c5 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
00002470  ac 04 00 00 32 01 00 00  00 00 00 00 00 00 00 00  |....2...........|
00002480  01 00 00 00 00 00 00 00  ce 00 00 00 01 00 00 00  |................|
00002490  00 00 00 00 00 00 00 00  e0 05 00 00 58 00 00 00  |............X...|
000024a0  00 00 00 00 00 00 00 00  08 00 00 00 00 00 00 00  |................|
000024b0  dd 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
000024c0  38 06 00 00 25 00 00 00  00 00 00 00 00 00 00 00  |8...%...........|
000024d0  01 00 00 00 00 00 00 00  ed 00 00 00 01 00 00 00  |................|
000024e0  00 00 00 00 00 00 00 00  5d 06 00 00 85 0c 00 00  |........].......|
000024f0  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
00002500  f9 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
00002510  e2 12 00 00 27 01 00 00  00 00 00 00 00 00 00 00  |....'...........|
00002520  01 00 00 00 00 00 00 00  07 01 00 00 01 00 00 00  |................|
00002530  00 00 00 00 00 00 00 00  09 14 00 00 f2 01 00 00  |................|
00002540  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
00002550  13 01 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
00002560  fc 15 00 00 14 00 00 00  00 00 00 00 00 00 00 00  |................|
00002570  04 00 00 00 00 00 00 00  20 01 00 00 01 00 00 00  |........ .......|
00002580  30 00 00 00 00 00 00 00  10 16 00 00 8a 09 00 00  |0...............|
00002590  00 00 00 00 00 00 00 00  01 00 00 00 01 00 00 00  |................|
000025a0  11 00 00 00 03 00 00 00  00 00 00 00 00 00 00 00  |................|
000025b0  9a 1f 00 00 2b 01 00 00  00 00 00 00 00 00 00 00  |....+...........|
000025c0  01 00 00 00 00 00 00 00  01 00 00 00 02 00 00 00  |................|
000025d0  00 00 00 00 00 00 00 00  18 26 00 00 80 04 00 00  |.........&......|
000025e0  21 00 00 00 37 00 00 00  04 00 00 00 10 00 00 00  |!...7...........|
000025f0  09 00 00 00 03 00 00 00  00 00 00 00 00 00 00 00  |................|
00002600  98 2a 00 00 ca 01 00 00  00 00 00 00 00 00 00 00  |.*..............|
00002610  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00002620  00 00 00 00 00 00 00 00  00 00 00 00 f4 80 04 08  |................|
00002630  00 00 00 00 03 00 01 00  00 00 00 00 08 81 04 08  |................|
00002640  00 00 00 00 03 00 02 00  00 00 00 00 28 81 04 08  |............(...|
00002650  00 00 00 00 03 00 03 00  00 00 00 00 50 81 04 08  |............P...|
00002660  00 00 00 00 03 00 04 00  00 00 00 00 a0 81 04 08  |................|
00002670  00 00 00 00 03 00 05 00  00 00 00 00 ec 81 04 08  |................|
00002680  00 00 00 00 03 00 06 00  00 00 00 00 f8 81 04 08  |................|
00002690  00 00 00 00 03 00 07 00  00 00 00 00 18 82 04 08  |................|
```

```
000026a0  00 00 00 00 03 00 08 00   00 00 00 00 20 82 04 08  |............ ...|
000026b0  00 00 00 00 03 00 09 00   00 00 00 00 30 82 04 08  |............0...|
000026c0  00 00 00 00 03 00 0a 00   00 00 00 00 48 82 04 08  |............H...|
000026d0  00 00 00 00 03 00 0b 00   00 00 00 00 78 82 04 08  |............x...|
000026e0  00 00 00 00 03 00 0c 00   00 00 00 00 74 83 04 08  |............t...|
000026f0  00 00 00 00 03 00 0d 00   00 00 00 00 90 83 04 08  |................|
00002700  00 00 00 00 03 00 0e 00   00 00 00 00 a8 93 04 08  |................|
00002710  00 00 00 00 03 00 0f 00   00 00 00 00 b4 93 04 08  |................|
00002720  00 00 00 00 03 00 10 00   00 00 00 00 b8 93 04 08  |................|
00002730  00 00 00 00 03 00 11 00   00 00 00 00 80 94 04 08  |................|
00002740  00 00 00 00 03 00 12 00   00 00 00 00 88 94 04 08  |................|
00002750  00 00 00 00 03 00 13 00   00 00 00 00 90 94 04 08  |................|
00002760  00 00 00 00 03 00 14 00   00 00 00 00 94 94 04 08  |................|
00002770  00 00 00 00 03 00 15 00   00 00 00 00 ac 94 04 08  |................|
00002780  00 00 00 00 03 00 16 00   00 00 00 00 00 00 00 00  |................|
00002790  00 00 00 00 03 00 17 00   00 00 00 00 00 00 00 00  |................|
000027a0  00 00 00 00 03 00 18 00   00 00 00 00 00 00 00 00  |................|
000027b0  00 00 00 00 03 00 19 00   00 00 00 00 00 00 00 00  |................|
000027c0  00 00 00 00 03 00 1a 00   00 00 00 00 00 00 00 00  |................|
000027d0  00 00 00 00 03 00 1b 00   00 00 00 00 00 00 00 00  |................|
000027e0  00 00 00 00 03 00 1c 00   00 00 00 00 00 00 00 00  |................|
000027f0  00 00 00 00 03 00 1d 00   00 00 00 00 00 00 00 00  |................|
00002800  00 00 00 00 03 00 1e 00   00 00 00 00 00 00 00 00  |................|
00002810  00 00 00 00 03 00 1f 00   00 00 00 00 00 00 00 00  |................|
00002820  00 00 00 00 03 00 20 00   00 00 00 00 00 00 00 00  |...... .........|
00002830  00 00 00 00 03 00 21 00   01 00 00 00 00 00 00 00  |......!.........|
00002840  00 00 00 00 04 00 f1 ff   08 00 00 00 00 00 00 00  |................|
00002850  00 00 00 00 04 00 f1 ff   13 00 00 00 9c 82 04 08  |................|
00002860  00 00 00 00 02 00 0c 00   23 00 00 00 00 00 00 00  |........#.......|
00002870  00 00 00 00 04 00 f1 ff   2e 00 00 00 80 94 04 08  |................|
00002880  00 00 00 00 01 00 12 00   3c 00 00 00 88 94 04 08  |........<.......|
00002890  00 00 00 00 01 00 13 00   4a 00 00 00 b4 93 04 08  |........J.......|
000028a0  00 00 00 00 01 00 10 00   5d 00 00 00 90 94 04 08  |........].......|
000028b0  00 00 00 00 01 00 14 00   6a 00 00 00 b0 93 04 08  |........j.......|
000028c0  00 00 00 00 01 00 0f 00   6e 00 00 00 ac 94 04 08  |........n.......|
000028d0  01 00 00 00 01 00 16 00   7a 00 00 00 c0 82 04 08  |........z.......|
000028e0  00 00 00 00 02 00 0c 00   90 00 00 00 fc 82 04 08  |................|
000028f0  00 00 00 00 02 00 0c 00   23 00 00 00 00 00 00 00  |........#.......|
00002900  00 00 00 00 04 00 f1 ff   9c 00 00 00 84 94 04 08  |................|
00002910  00 00 00 00 01 00 12 00   a9 00 00 00 8c 94 04 08  |................|
00002920  00 00 00 00 01 00 13 00   b6 00 00 00 b4 93 04 08  |................|
00002930  00 00 00 00 01 00 10 00   c4 00 00 00 90 94 04 08  |................|
00002940  00 00 00 00 01 00 14 00   d0 00 00 00 50 83 04 08  |............P...|
00002950  00 00 00 00 02 00 0c 00   08 00 00 00 00 00 00 00  |................|
00002960  00 00 00 00 04 00 f1 ff   e6 00 00 00 00 00 00 00  |................|
00002970  00 00 00 00 04 00 f1 ff   ee 00 00 00 ac 93 04 08  |................|
00002980  00 00 00 00 01 02 0f 00   fb 00 00 00 b8 93 04 08  |................|
00002990  00 00 00 00 11 00 11 00   04 01 00 00 90 83 04 08  |................|
000029a0  04 00 00 00 11 00 0e 00   0b 01 00 00 30 82 04 08  |............0...|
000029b0  00 00 00 00 12 00 0a 00   11 01 00 00 78 82 04 08  |............x...|
000029c0  00 00 00 00 12 00 0c 00   18 01 00 00 ac 94 04 08  |................|
000029d0  00 00 00 00 10 00 f1 ff   24 01 00 00 28 83 04 08  |........$...(...|
000029e0  27 00 00 00 12 00 0c 00   29 01 00 00 58 82 04 08  |'.......)...X...|
000029f0  d8 00 00 00 12 00 00 00   46 01 00 00 a8 93 04 08  |........F.......|
00002a00  00 00 00 00 20 00 0f 00   51 01 00 00 68 82 04 08  |.... ...Q...h...|
00002a10  39 00 00 00 12 00 00 00   63 01 00 00 74 83 04 08  |9.......c...t...|
00002a20  00 00 00 00 12 00 0d 00   69 01 00 00 ac 94 04 08  |........i.......|
```

```
00002a30    00 00 00 00 10 00 f1 ff    70 01 00 00 94 94 04 08    |........p.......|
00002a40    00 00 00 00 11 00 15 00    86 01 00 00 b0 94 04 08    |................|
00002a50    00 00 00 00 10 00 f1 ff    8b 01 00 00 94 83 04 08    |................|
00002a60    04 00 00 00 11 00 0e 00    9a 01 00 00 a8 93 04 08    |................|
00002a70    00 00 00 00 10 00 0f 00    a7 01 00 00 00 00 00 00    |................|
00002a80    00 00 00 00 20 00 00 00    bb 01 00 00 00 00 00 00    |.... ...........|
00002a90    00 00 00 00 20 00 00 00    00 69 6e 69 74 2e 63 00    |.... ....init.c.|
00002aa0    69 6e 69 74 66 69 6e 69    2e 63 00 63 61 6c 6c 5f    |initfini.c.call_|
00002ab0    67 6d 6f 6e 5f 73 74 61    72 74 00 63 72 74 73 74    |gmon_start.crtst|
00002ac0    75 66 66 2e 63 00 5f 5f    43 54 4f 52 5f 4c 49 53    |uff.c.__CTOR_LIS|
00002ad0    54 5f 5f 00 5f 5f 44 54    4f 52 5f 4c 49 53 54 5f    |T__.__DTOR_LIST_|
00002ae0    5f 00 5f 5f 45 48 5f 46    52 41 4d 45 5f 42 45 47    |_.__EH_FRAME_BEG|
00002af0    49 4e 5f 5f 00 5f 5f 4a    43 52 5f 4c 49 53 54 5f    |IN__.__JCR_LIST_|
00002b00    5f 00 70 2e 30 00 63 6f    6d 70 6c 65 74 65 64 2e    |_.p.0.completed.|
00002b10    31 00 5f 5f 64 6f 5f 67    6c 6f 62 61 6c 5f 64 74    |1.__do_global_dt|
00002b20    6f 72 73 5f 61 75 78 00    66 72 61 6d 65 5f 64 75    |ors_aux.frame_du|
00002b30    6d 6d 79 00 5f 5f 43 54    4f 52 5f 45 4e 44 5f 5f    |mmy.__CTOR_END__|
00002b40    00 5f 5f 44 54 4f 52 5f    45 4e 44 5f 5f 00 5f 5f    |.__DTOR_END__.__|
00002b50    46 52 41 4d 45 5f 45 4e    44 5f 5f 00 5f 5f 4a 43    |FRAME_END__.__JC|
00002b60    52 5f 45 4e 44 5f 5f 00    5f 5f 64 6f 5f 67 6c 6f    |R_END__.__do_glo|
00002b70    62 61 6c 5f 63 74 6f 72    73 5f 61 75 78 00 68 65    |bal_ctors_aux.he|
00002b80    6c 6c 6f 2e 63 00 5f 5f    64 73 6f 5f 68 61 6e 64    |llo.c.__dso_hand|
00002b90    6c 65 00 5f 44 59 4e 41    4d 49 43 00 5f 66 70 5f    |le._DYNAMIC._fp_|
00002ba0    68 77 00 5f 69 6e 69 74    00 5f 73 74 61 72 74 00    |hw._init._start.|
00002bb0    5f 5f 62 73 73 5f 73 74    61 72 74 00 6d 61 69 6e    |__bss_start.main|
00002bc0    00 5f 5f 6c 69 62 63 5f    73 74 61 72 74 5f 6d 61    |.__libc_start_ma|
00002bd0    69 6e 40 40 47 4c 49 42    43 5f 32 2e 30 00 64 61    |in@@GLIBC_2.0.da|
00002be0    74 61 5f 73 74 61 72 74    00 70 72 69 6e 74 66 40    |ta_start.printf@|
00002bf0    40 47 4c 49 42 43 5f 32    2e 30 00 5f 66 69 6e 69    |@GLIBC_2.0._fini|
00002c00    00 5f 65 64 61 74 61 00    5f 47 4c 4f 42 41 4c 5f    |._edata._GLOBAL_|
00002c10    4f 46 46 53 45 54 5f 54    41 42 4c 45 5f 00 5f 65    |OFFSET_TABLE_._e|
00002c20    6e 64 00 5f 49 4f 5f 73    74 64 69 6e 5f 75 73 65    |nd._IO_stdin_use|
00002c30    64 00 5f 5f 64 61 74 61    5f 73 74 61 72 74 00 5f    |d.__data_start._|
00002c40    4a 76 5f 52 65 67 69 73    74 65 72 43 6c 61 73 73    |Jv_RegisterClass|
00002c50    65 73 00 5f 5f 67 6d 6f    6e 5f 73 74 61 72 74 5f    |es.__gmon_start_|
00002c60    5f 00                                                 |_.|
00002c62
```

B. readelf –a hello

```
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:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x8048278
  Start of program headers:          52 (bytes into file)
  Start of section headers:          8392 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
```

```
  Number of program headers:         6
  Size of section headers:           40 (bytes)
  Number of section headers:         34
  Section header string table index: 31

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        080480f4 0000f4 000013 00   A  0   0  1
  [ 2] .note.ABI-tag     NOTE            08048108 000108 000020 00   A  0   0  4
  [ 3] .hash             HASH            08048128 000128 000028 04   A  4   0  4
  [ 4] .dynsym           DYNSYM          08048150 000150 000050 10   A  5   1
4
  [ 5] .dynstr           STRTAB          080481a0 0001a0 00004c 00   A  0   0  1
  [ 6] .gnu.version      VERSYM          080481ec 0001ec 00000a 02   A  4   0  2
  [ 7] .gnu.version_r    VERNEED         080481f8 0001f8 000020 00   A  5   1  4
  [ 8] .rel.dyn          REL             08048218 000218 000008 08   A  4   0  4
  [ 9] .rel.plt          REL             08048220 000220 000010 08   A  4   b  4
  [10] .init             PROGBITS        08048230 000230 000018 00  AX  0   0  4
  [11] .plt              PROGBITS        08048248 000248 000030 04  AX  0   0  4
  [12] .text             PROGBITS        08048278 000278 0000fc 00  AX  0   0  4
  [13] .fini             PROGBITS        08048374 000374 00001c 00  AX  0   0  4
  [14] .rodata           PROGBITS        08048390 000390 000015 00   A  0   0  4
  [15] .data             PROGBITS        080493a8 0003a8 00000c 00  WA  0   0  4
  [16] .eh_frame         PROGBITS        080493b4 0003b4 000004 00  WA  0   0  4
  [17] .dynamic          DYNAMIC         080493b8 0003b8 0000c8 08  WA  5   0
4
  [18] .ctors            PROGBITS        08049480 000480 000008 00  WA  0   0  4
  [19] .dtors            PROGBITS        08049488 000488 000008 00  WA  0   0  4
  [20] .jcr              PROGBITS        08049490 000490 000004 00  WA  0   0  4
  [21] .got              PROGBITS        08049494 000494 000018 04  WA  0   0  4
  [22] .bss              NOBITS          080494ac 0004ac 000004 00  WA  0   0  4
  [23] .comment          PROGBITS        00000000 0004ac 000132 00      0   0  1
  [24] .debug_aranges    PROGBITS        00000000 0005e0 000058 00      0   0  8
  [25] .debug_pubnames   PROGBITS        00000000 000638 000025 00      0   0  1
  [26] .debug_info       PROGBITS        00000000 00065d 000c85 00      0   0  1
  [27] .debug_abbrev     PROGBITS        00000000 0012e2 000127 00      0   0  1
  [28] .debug_line       PROGBITS        00000000 001409 0001f2 00      0   0  1
  [29] .debug_frame      PROGBITS        00000000 0015fc 000014 00      0   0  4
  [30] .debug_str        PROGBITS        00000000 001610 00098a 01  MS  0   0  1
  [31] .shstrtab         STRTAB          00000000 001f9a 00012b 00      0   0  1
  [32] .symtab           SYMTAB          00000000 002618 000480 10     33  37  4
  [33] .strtab           STRTAB          00000000 002a98 0001ca 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x08048034 0x08048034 0x000c0 0x000c0 R E 0x4
  INTERP         0x0000f4 0x080480f4 0x080480f4 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /lib/ld-linux.so.2]
  LOAD           0x000000 0x08048000 0x08048000 0x003a5 0x003a5 R E 0x1000
  LOAD           0x0003a8 0x080493a8 0x080493a8 0x00104 0x00108 RW  0x1000
  DYNAMIC        0x0003b8 0x080493b8 0x080493b8 0x000c8 0x000c8 RW  0x4
  NOTE           0x000108 0x08048108 0x08048108 0x00020 0x00020 R   0x4
```

```
 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp

02     .interp .note.ABI-tag .hash .dynsym .dynstr .gnu.version .gnu.version_r .rel.dyn .rel.plt .in
it .plt .text .fini .rodata
   03     .data .eh_frame .dynamic .ctors .dtors .jcr .got .bss
   04     .dynamic
   05     .note.ABI-tag

Dynamic segment at offset 0x3b8 contains 20 entries:
  Tag        Type                         Name/Value
 0x00000001 (NEEDED)                     Shared library: [libc.so.6]
 0x0000000c (INIT)                       0x8048230
 0x0000000d (FINI)                       0x8048374
 0x00000004 (HASH)                       0x8048128
 0x00000005 (STRTAB)                     0x80481a0
 0x00000006 (SYMTAB)                     0x8048150
 0x0000000a (STRSZ)                      76 (bytes)
 0x0000000b (SYMENT)                     16 (bytes)
 0x00000015 (DEBUG)                      0x0
 0x00000003 (PLTGOT)                     0x8049494
 0x00000002 (PLTRELSZ)                   16 (bytes)
 0x00000014 (PLTREL)                     REL
 0x00000017 (JMPREL)                     0x8048220
 0x00000011 (REL)                        0x8048218
 0x00000012 (RELSZ)                      8 (bytes)
 0x00000013 (RELENT)                     8 (bytes)
 0x6ffffffe (VERNEED)                    0x80481f8
 0x6fffffff (VERNEEDNUM)                 1
 0x6ffffff0 (VERSYM)                     0x80481ec
 0x00000000 (NULL)                       0x0

Relocation section '.rel.dyn' at offset 0x218 contains 1 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
080494a8  00000406 R_386_GLOB_DAT    00000000   __gmon_start__

Relocation section '.rel.plt' at offset 0x220 contains 2 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
080494a0  00000107 R_386_JUMP_SLOT   08048258   __libc_start_main
080494a4  00000207 R_386_JUMP_SLOT   08048268   printf

There are no unwind sections in this file.

Symbol table '.dynsym' contains 5 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 08048258   216 FUNC    GLOBAL DEFAULT  UND
__libc_start_main@GLIBC_2.0 (2)
     2: 08048268    57 FUNC    GLOBAL DEFAULT  UND printf@GLIBC_2.0 (2)
     3: 08048394     4 OBJECT  GLOBAL DEFAULT   14 _IO_stdin_used
     4: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__

Symbol table '.symtab' contains 72 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
```

```
  0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
  1: 080480f4     0 SECTION LOCAL  DEFAULT    1
  2: 08048108     0 SECTION LOCAL  DEFAULT    2
  3: 08048128     0 SECTION LOCAL  DEFAULT    3
  4: 08048150     0 SECTION LOCAL  DEFAULT    4
  5: 080481a0     0 SECTION LOCAL  DEFAULT    5
  6: 080481ec     0 SECTION LOCAL  DEFAULT    6
  7: 080481f8     0 SECTION LOCAL  DEFAULT    7
  8: 08048218     0 SECTION LOCAL  DEFAULT    8
  9: 08048220     0 SECTION LOCAL  DEFAULT    9
 10: 08048230     0 SECTION LOCAL  DEFAULT   10
 11: 08048248     0 SECTION LOCAL  DEFAULT   11
 12: 08048278     0 SECTION LOCAL  DEFAULT   12
 13: 08048374     0 SECTION LOCAL  DEFAULT   13
 14: 08048390     0 SECTION LOCAL  DEFAULT   14
 15: 080493a8     0 SECTION LOCAL  DEFAULT   15
 16: 080493b4     0 SECTION LOCAL  DEFAULT   16
 17: 080493b8     0 SECTION LOCAL  DEFAULT   17
 18: 08049480     0 SECTION LOCAL  DEFAULT   18
 19: 08049488     0 SECTION LOCAL  DEFAULT   19
 20: 08049490     0 SECTION LOCAL  DEFAULT   20
 21: 08049494     0 SECTION LOCAL  DEFAULT   21
 22: 080494ac     0 SECTION LOCAL  DEFAULT   22
 23: 00000000     0 SECTION LOCAL  DEFAULT   23
 24: 00000000     0 SECTION LOCAL  DEFAULT   24
 25: 00000000     0 SECTION LOCAL  DEFAULT   25
 26: 00000000     0 SECTION LOCAL  DEFAULT   26
 27: 00000000     0 SECTION LOCAL  DEFAULT   27
 28: 00000000     0 SECTION LOCAL  DEFAULT   28
 29: 00000000     0 SECTION LOCAL  DEFAULT   29
 30: 00000000     0 SECTION LOCAL  DEFAULT   30
 31: 00000000     0 SECTION LOCAL  DEFAULT   31
 32: 00000000     0 SECTION LOCAL  DEFAULT   32
 33: 00000000     0 SECTION LOCAL  DEFAULT   33
 34: 00000000     0 FILE    LOCAL  DEFAULT  ABS init.c
 35: 00000000     0 FILE    LOCAL  DEFAULT  ABS initfini.c
 36: 0804829c     0 FUNC    LOCAL  DEFAULT   12 call_gmon_start
 37: 00000000     0 FILE    LOCAL  DEFAULT  ABS crtstuff.c
 38: 08049480     0 OBJECT  LOCAL  DEFAULT   18 __CTOR_LIST__
 39: 08049488     0 OBJECT  LOCAL  DEFAULT   19 __DTOR_LIST__
 40: 080493b4     0 OBJECT  LOCAL  DEFAULT   16 __EH_FRAME_BEGIN__
 41: 08049490     0 OBJECT  LOCAL  DEFAULT   20 __JCR_LIST__
 42: 080493b0     0 OBJECT  LOCAL  DEFAULT   15 p.0
 43: 080494ac     1 OBJECT  LOCAL  DEFAULT   22 completed.1
 44: 080482c0     0 FUNC    LOCAL  DEFAULT   12 __do_global_dtors_aux
 45: 080482fc     0 FUNC    LOCAL  DEFAULT   12 frame_dummy
 46: 00000000     0 FILE    LOCAL  DEFAULT  ABS crtstuff.c
 47: 08049484     0 OBJECT  LOCAL  DEFAULT   18 __CTOR_END__
 48: 0804948c     0 OBJECT  LOCAL  DEFAULT   19 __DTOR_END__
 49: 080493b4     0 OBJECT  LOCAL  DEFAULT   16 __FRAME_END__
 50: 08049490     0 OBJECT  LOCAL  DEFAULT   20 __JCR_END__
 51: 08048350     0 FUNC    LOCAL  DEFAULT   12 __do_global_ctors_aux
 52: 00000000     0 FILE    LOCAL  DEFAULT  ABS initfini.c
 53: 00000000     0 FILE    LOCAL  DEFAULT  ABS hello.c
 54: 080493ac     0 OBJECT  LOCAL  HIDDEN   15 __dso_handle
 55: 080493b8     0 OBJECT  GLOBAL DEFAULT   17 _DYNAMIC
 56: 08048390     4 OBJECT  GLOBAL DEFAULT   14 _fp_hw
```

```
     57: 08048230     0 FUNC    GLOBAL DEFAULT    10 _init
     58: 08048278     0 FUNC    GLOBAL DEFAULT    12 _start
     59: 080494ac     0 NOTYPE  GLOBAL DEFAULT   ABS __bss_start
     60: 08048328    39 FUNC    GLOBAL DEFAULT    12 main
     61: 08048258   216 FUNC    GLOBAL DEFAULT   UND
__libc_start_main@@GLIBC_
     62: 080493a8     0 NOTYPE  WEAK   DEFAULT    15 data_start
     63: 08048268    57 FUNC    GLOBAL DEFAULT   UND printf@@GLIBC_2.0
     64: 08048374     0 FUNC    GLOBAL DEFAULT    13 _fini
     65: 080494ac     0 NOTYPE  GLOBAL DEFAULT   ABS _edata
     66: 08049494     0 OBJECT  GLOBAL DEFAULT   21
_GLOBAL_OFFSET_TABLE_
     67: 080494b0     0 NOTYPE  GLOBAL DEFAULT   ABS _end
     68: 08048394     4 OBJECT  GLOBAL DEFAULT    14 _IO_stdin_used
     69: 080493a8     0 NOTYPE  GLOBAL DEFAULT    15 __data_start
     70: 00000000     0 NOTYPE  WEAK   DEFAULT   UND _Jv_RegisterClasses
     71: 00000000     0 NOTYPE  WEAK   DEFAULT   UND __gmon_start__

Histogram for bucket list length (total of 3 buckets):
 Length  Number     % of total  Coverage
      0  0          (  0.0%)
      1  2          ( 66.7%)     50.0%
      2  1          ( 33.3%)    100.0%

Version symbols section '.gnu.version' contains 5 entries:
 Addr: 00000000080481ec  Offset: 0x0001ec  Link: 4 (.dynsym)
  000:   0 (*local*)       2 (GLIBC_2.0)       2 (GLIBC_2.0)       1 (*global*)
  004:   0 (*local*)

Version needs section '.gnu.version_r' contains 1 entries:
 Addr: 0x00000000080481f8  Offset: 0x0001f8  Link to section: 5 (.dynstr)
  000000: Version: 1  File: libc.so.6  Cnt: 1
  0x0010: Name: GLIBC_2.0  Flags: none  Version: 2
```

Reference