### ELF视图



图 1 - 1

ELF文件主要包含三部分组成：描述文件整体信息的elf头、描述elf文件加载相关信息的program头表和描述文件节区内容信息的section头表。在elf文件未加载链接前，program头表对链接过程无影响，链接器只需根据section头表中描述的字符串、符号、重定位信息即可对elf文件完成重定位修复、符号地址解析过程，生成最终目标文件；在elf加载执行时，section头表有无不会影响此过程，加载器根据program头表中描述的加载信息将对应的sections加载到对应的基址位置即可。即一个program头描述的segment可能由多个section区所组成。

### ELF文件整体结构

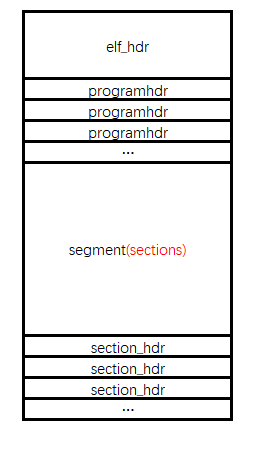


图 1 - 1

elf\_hdr中定义了描述文件整个结构的字段，如描述其他头部的：program\_hdr、section\_hdr的起始偏移，program\_hdr、section\_hdr的结构大小，program\_hdr、section\_hdr的包含的元素个数；如整个elf文件的类型；elf编译的目标平台，编译的字长位数大小等等。



### 文件头解析(Elfxx\_Ehdr)

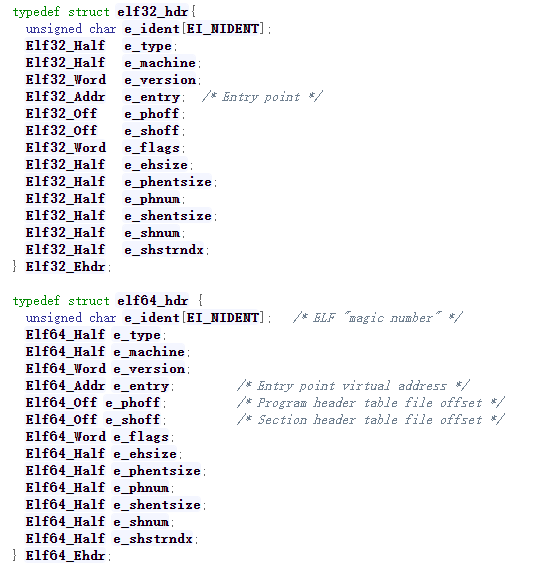
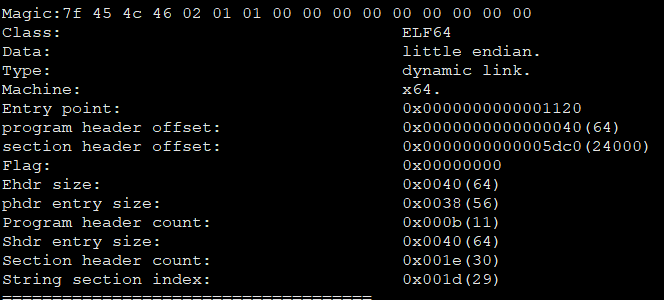


图3 - 1

ELF文件头结构如上，32位和64位系统上主要差异在于带注释的三个字段扩展到了64位字长，其主要描述ELF文件的整体格式，主要字段的含义见上一节具体描述。



### 程序头表解析(Elfxx\_Phdr)

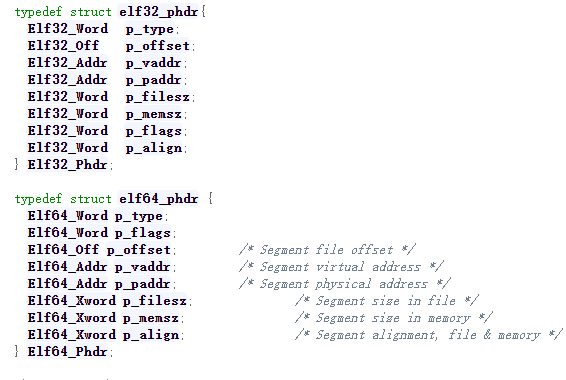
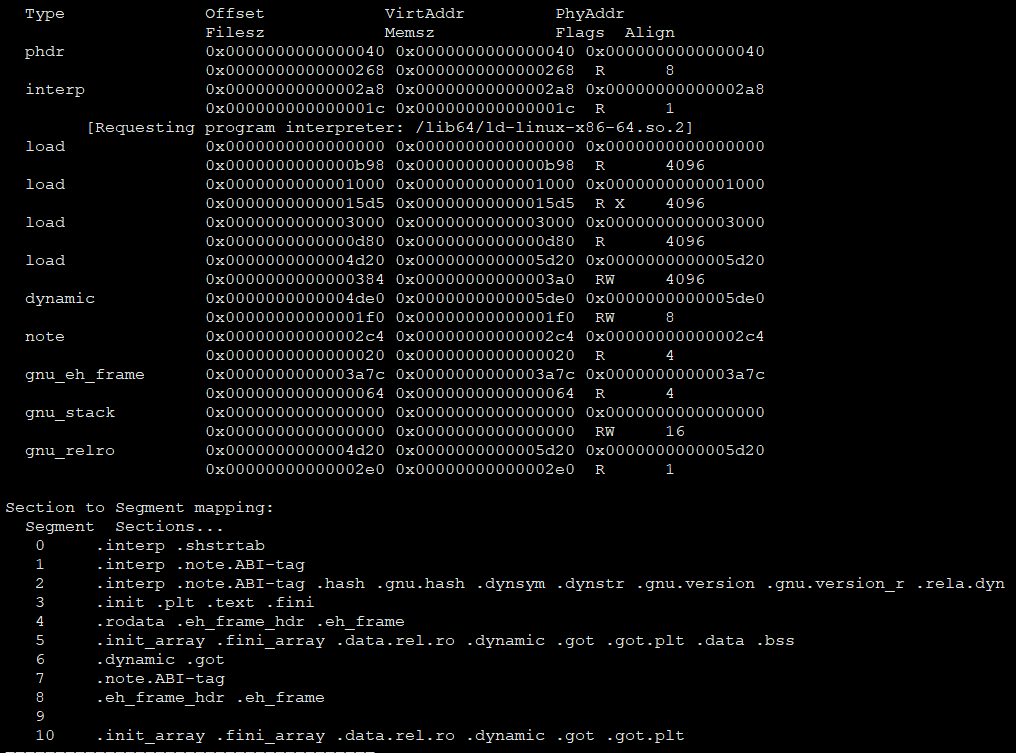
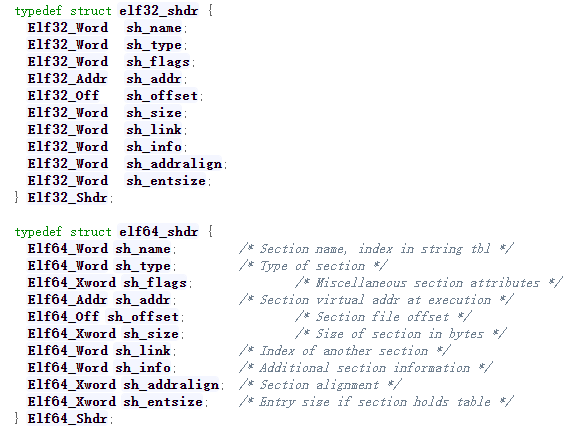


图4 - 1

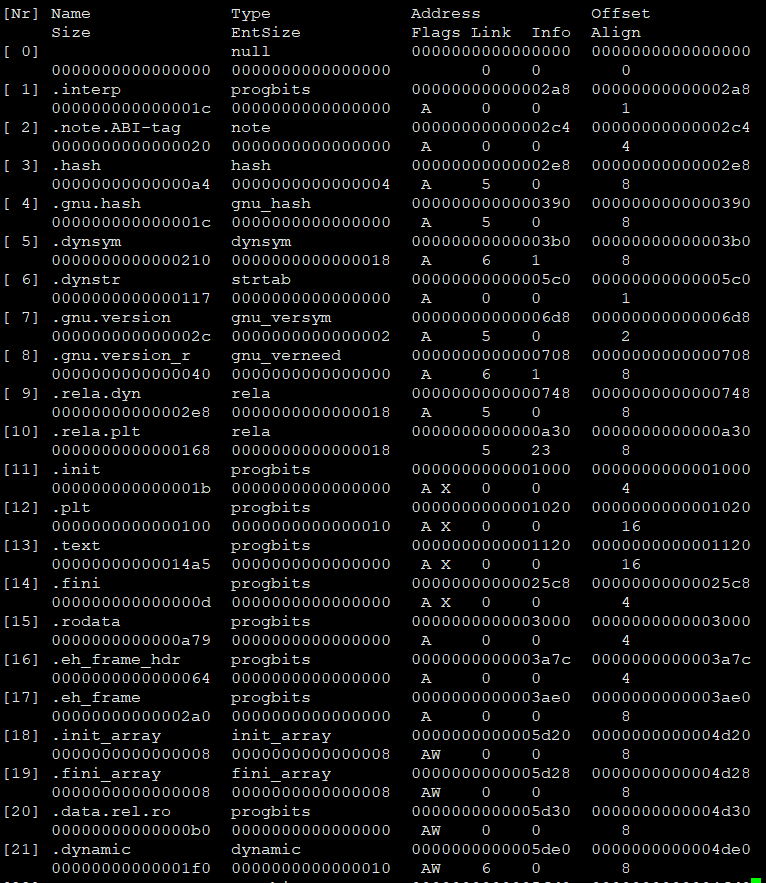
程序头表是一个程序头结构组成的数组，每一项用于描述程序动态加载时将要加载的部分、加载的位置和大小，以及相应加载区域的类型标记等信息，如下解析过程主要定位各个segment的描述信息以及segment于section映射关系。



### 节头表解析(Elfxx\_Shdr)



节头表是一个节头组成的数组，数组大小定义于ELF文件头中，每一项描述一个section相关信息，如节名称、节类型、节的偏移(相对于文件头基址)等等。



### 符号表解析(Elfxx\_Sym)

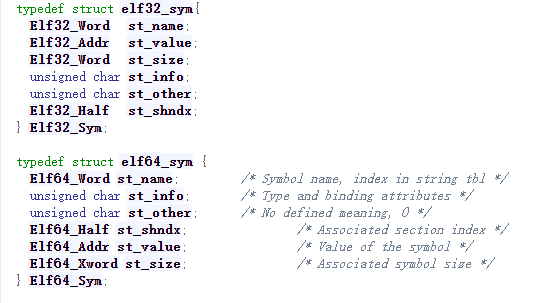
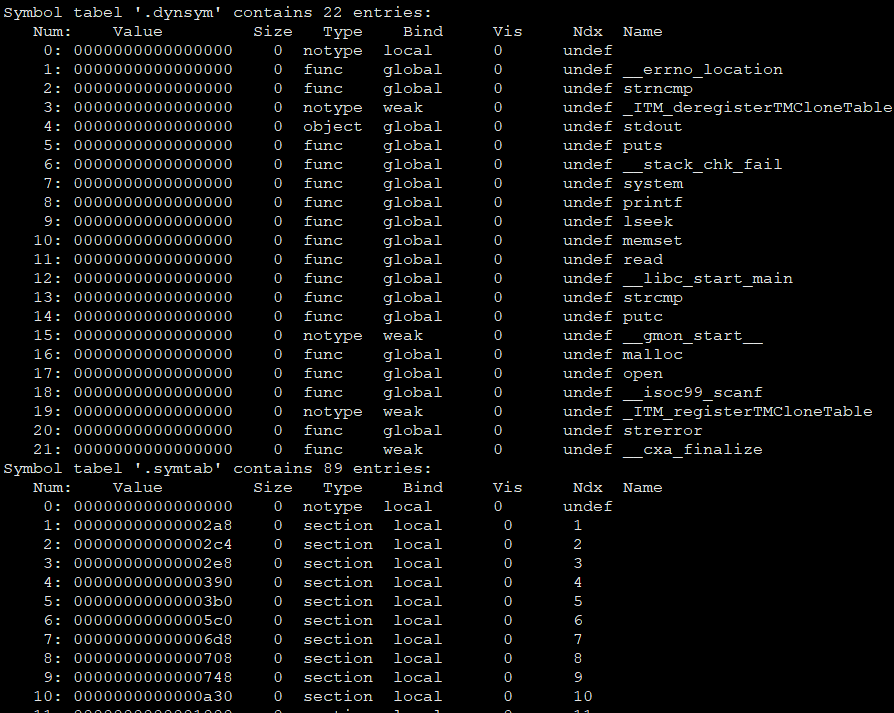


图6 - 1

当ELF文件节区名称为”.dynsym”或”.symtab”时表示此节为一个包含符号信息的节区， ELF文件正常情况下包含两类符号表(节)信息，节名称分别为”.dynsym”、”.symtab”，前者包含此ELF引入的外部符号，比如程序中调用一个库函数，此函数符号即视为外部引入符号，存放于此节。而后者是个包含前者的集合，包含外部符号和本地符号的全集。而只有dynsym节才是链接器解析符号需要的。.symtab节可删除并不会加载到内存中。通过节区中offset字段即可索引到符号表项的结构地址，每一个符号项描述一个符号信息，如符号名称的偏移、符号类型等信息。



### 重定位表解析(Elfxx\_Rela)

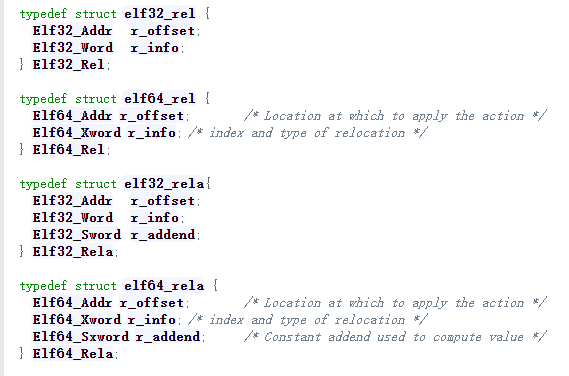
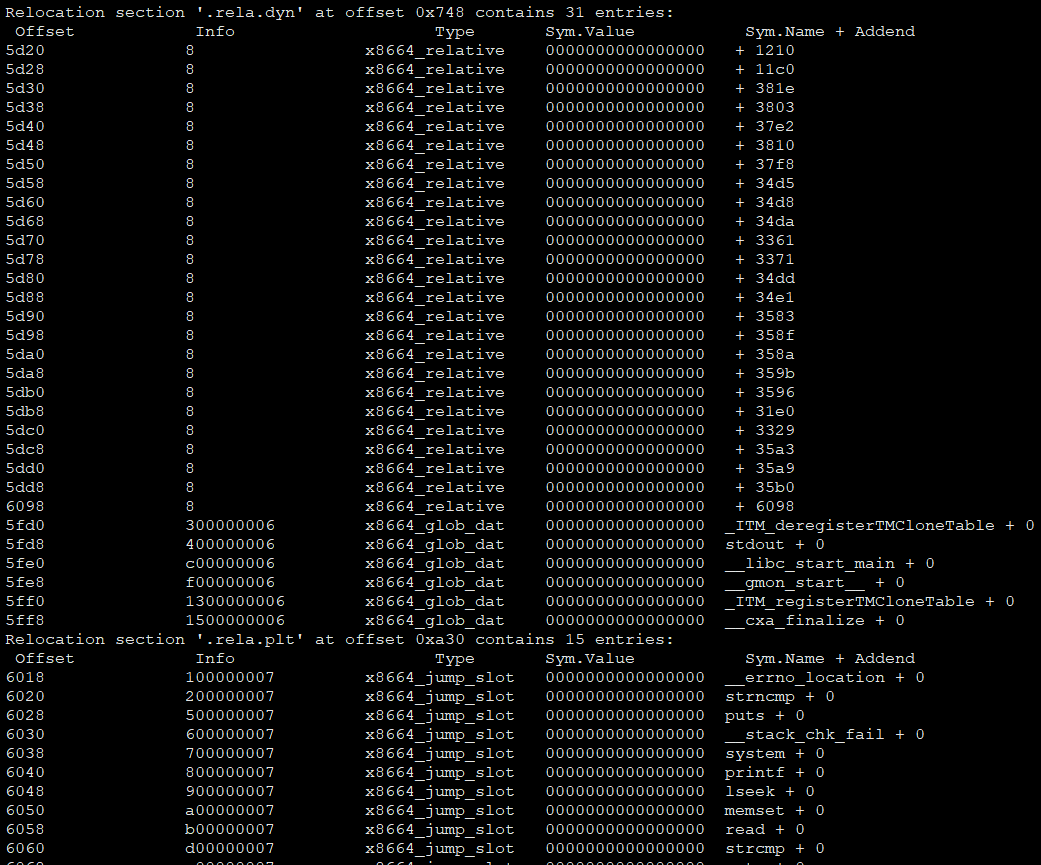


图7 - 1

同符号表一样，当节区名称以”.rel”开头时表示此节是一个描述重定位信息的节区，通过节区offset字段描述的偏移即可索引到重定位表的基址，每一个重定位项描述待重定位地址的偏移、重定位的符号名称，重定位类型等信息。

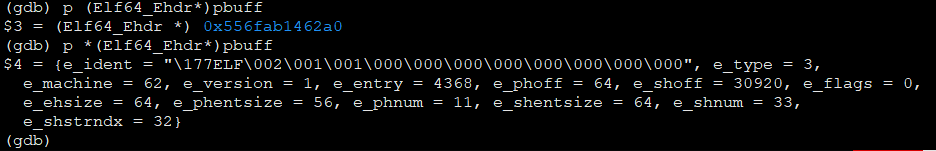


如图，常见的x8664\_relative是一种重定位方式，此种重定位类型的修复方式为(模块加载基址+Addend)，常见于知道符号相对地址，但模块加载基址尚未确定时的重定位。

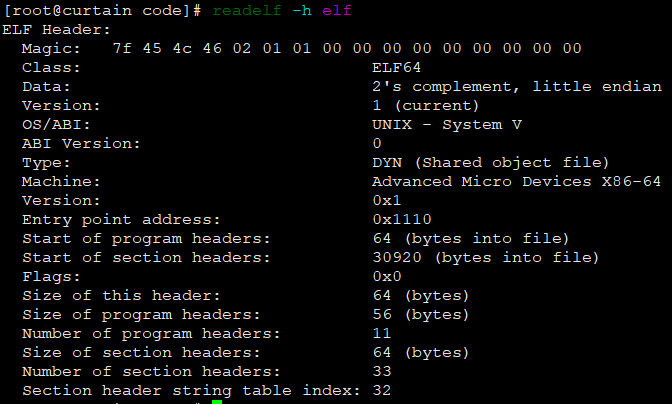
### 二进制分析

#### 8.1 文件头结构

动态：使用gdb attach到elf所在进程，在文件头解析函数下断，run起来后，查看对应结构内存如下：



使用已有工具解析对应文件头作对照：



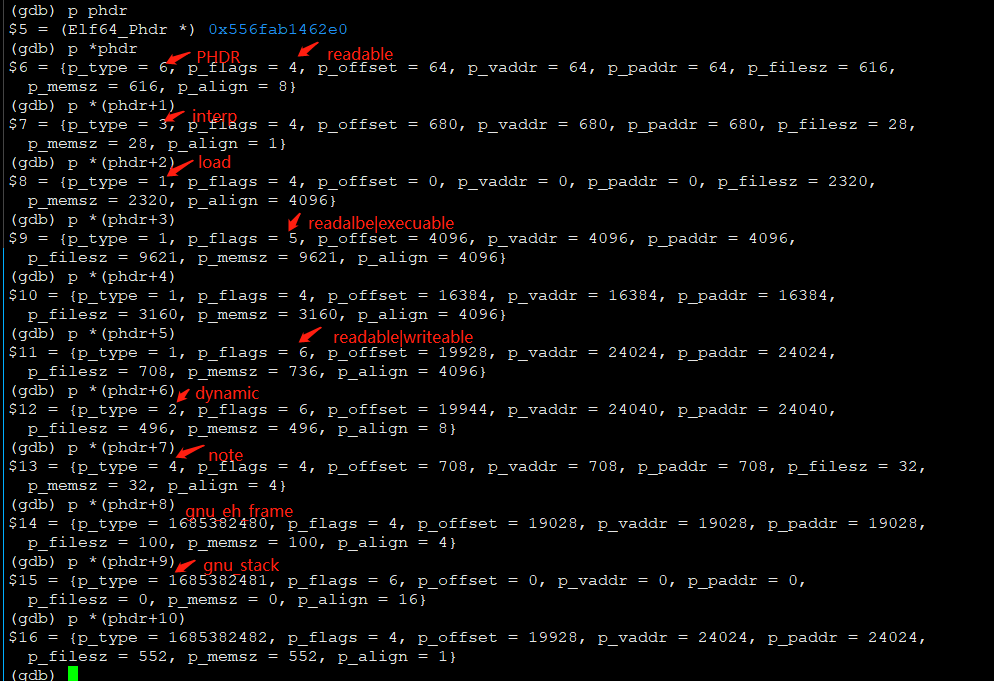
可看到二者解析内容一致。

静态：IDA加载之，查看对应函数反编译结果：

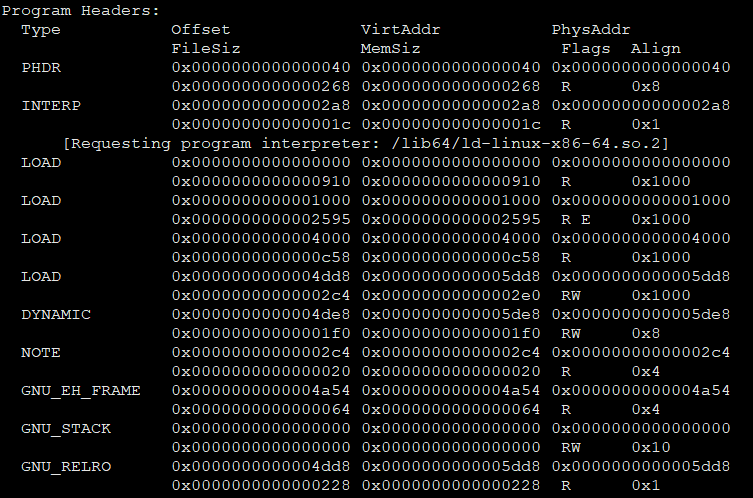


#### 8.2 程序头表结构

同样在程序头解析函数下断，run起来之后即可断到对应解析点。

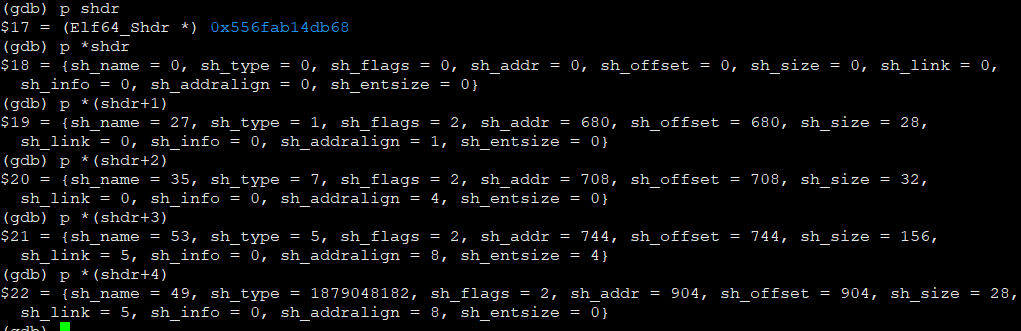


同理还是拿出一组对照

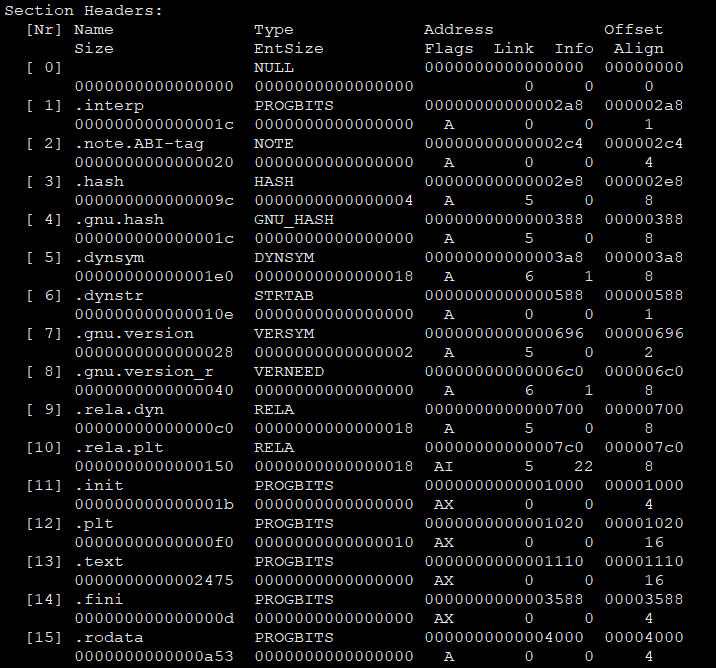


#### 8.3 节头表结构

同理，由于节表较多，此处仅给出范例。

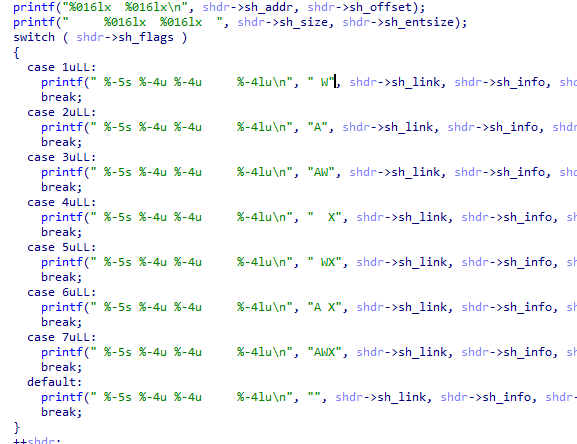


可看到sh\_name字段仅作为节表名称在字符串表中的偏移，节的类型和属性等。

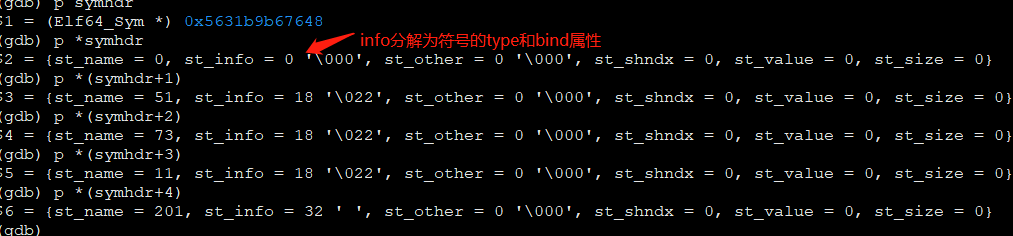


静态：反编译之后的结果基本跟源码差不多了。。。





#### 8.4 符号表结构





#### 8.5 重定位结构

