objdump -t (或等价用法obdump --syms)列含义. 每一列用空格分开.

## 符号的地址

必有

## 符号的作用域

大多数情况必有

* l = local
* g = global
* u = global unique
* (空格) = 既不是global也不是local
* ! = 既是global又是local

一般l, g, u是最常见的.

## 符号的强弱

应用开发少见, 库开发或嵌入式开发常见

**首先要解释一下弱符号存在的意义** [参考: https://my.oschina.net/senmole/blog/50887] :

若两个或两个以上全局符号 (函数或变量名) 名字一样, 而其中之一声明为weak symbol (弱符号), 则这些全局符号不会引发重定义错误. 链接器会忽略弱符号, 去使用普通的全局符号来解析所有对这些符号的引用.

但当普通的全局符号不可用时, 链接器会使用弱符号.

* w = 符号为弱符号
* (空格) = 符号为默认情况, 即强符号

我的理解就是弱符号可以被override.

## 符号是否为构造函数

C++

* C = 是
* (空格) = 不是

## 符号是否为警告

罕见

* W = 是
* (空格) = 不是

## 符号是否为间接引用

罕见

* i = a function to be evaluated during reloc processing (不懂, 我的直觉判断这跟虚函数有关)
* I = indirect reference to another symbol (不懂, 难道是alias?)
* (空格) = 普通符号

## 是否为调试符号

罕见, 不懂

* d = 调试符号
* D = 动态符号
* (空格) = 普通符号

## 符号类型

常见

* F = 函数
* f = 文件
* O = 对象
* (空格) = 普通符号

## 符号所在内存区段

必有

* 如.bss, .text, .stack, .heap等就不细说了
* ABS区段: 区段的位置和长度是绝对寻址的
* UND区段: 说明符号是一个extern符号, 没有在当前文件中定义

## 符号的对齐或大小

manpage原文:

After the section name comes another field, a number, which for common symbols is the alignment and for other symbol is the size.

这里有两个问题我不懂:

* 如何界定common symbol和other symbol?
* 如果是common symbol, 它的alignment为什么是8个0, 什么意思?  
  推测:8个0难道是8字节对齐??

## 符号名称