目录

可执行文件格式：ELF

- 最新版本： `v1.0.1`
- 更新时间： `20231004`

简介

介绍常见的可执行文件格式：ELF。主要是Linux和Android的常见格式。先是ELF概览，包括ELF文件类型和术语和概念；然后是ELF结构详解，包括两种视图、结构图、链接和执行过程以及section节和segment段；然后介绍ELF的解析工具，包括读取信息的和解析修改的；读取信息的有Linux通用的readelf、objdump、rabin2等和Android专用的JEB等。且都有详细的安装、用法和举例；解析工具包括LIEF；然后专门介绍Android中的ELF格式。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- crifan/exec_file_format_elf: 可执行文件格式：ELF

如何使用此HonKit源码去生成发布为电子书

详见：crifan/honkit_template: demo how to use crifan honkit template and demo

在线浏览

- 可执行文件格式：ELF book.crifan.org
- 可执行文件格式：ELF crifan.github.io

离线下载阅读

- 可执行文件格式：ELF PDF
- 可执行文件格式：ELF ePub
- 可执行文件格式：ELF Mobi

版权和用途说明

鸣谢

感谢我的老婆**陈雪**的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 `150+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

关于作者

关于作者更多介绍，详见：

关于CrifanLi李茂 – 在路上

ELF概览

- `ELF = Executable and Linking Format = 可执行和链接格式`
  - 是什么：一种文件格式file format
    - that defines how an object file is composed and organized
  - 用途：With this information, your kernel and the binary loader know how to load the file, where to look for the code, where to look the initialized data, which shared library that needs to be loaded and so on.
  - 主要历史
    - Linux和安卓通用的可执行文件格式： `ELF`
      - 最早：Linux通用可执行文件格式： `ELF`
      - 后来：Android是基于Linux的，所以也是沿用 `ELF`
        - 详见：Android中的ELF

ELF资料

- ELF资料
  - ELF格式详细定义
    - Executable and Linkable Format - Wikipedia
    - elf(5) — Linux manual pages (courier-mta.org)
    - ELF Header (sco.com)
  - 各个section节的含义
    - Special Sections (oracle.com)
    - Executable and Linkable Format (ELF) (netmeister.org)
  - elf内部过程
    - s.eresi-project.org/inc/articles/elf-rtld.txt
      - Understanding Linux ELF RTLD internals

ELF文件类型

- ELF文件类型
    - `Relocatable File` = `可重定位文件`
        - an object file that holds code and data suitable for linking with other object files to create an executable or a shared object file. In other word, you can say that relocatable file is a foundation for creating executables and libraries
        - 常见后缀
            - object file= `.o` 文件
                - 举例
                    - `gcc -c test.c`
                        - 生成： `test.o`
            - Kernel module = `.o` 或 `.ko`
    - `Executable File` = `可执行文件`
        - object file that holds a program suitable for execution
        - 常见后缀：无后缀
            - 二进制文件
                - `gcc -o test test.c`
                    - 生成（无后缀的）： `test`
    - `Shared Object File` = `共享对象文件` =SO文件= `Shared object` == `DYNamic link library`
        - A shared object file holds code and data suitable for linking in two contexts
            1. the link editor may process it with other relocatable and shared object files to create another object file
            2. the dynamic linker combines it with an executable file and other shared objects to create a process image
        - 常见后缀
            - `.so` 文件

举例说明

用readelf查看header中文件类型

- 举例1

可以用 `readelf` 查看header，确定一个文件的类型到底是什么： `Relocatable file` / `Executable file` / `Shared object file`

```
$ readelf -h /bin/ls
Type: EXEC (Executable file)

$ readelf -h /usr/lib/crt1.o
Type: REL (Relocatable file)

$ readelf-h /lib/libc-2.3.2.so
Type: DYN (Shared object file)
```

- 举例2

```
➜  arm64-v8a readelf -h libtacker.so
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
...
  Type:                              DYN (Shared object file)
```

->

- ARM64架构的 `Shared Object File` = `DYN` = `Dynamic Library`

ELF术语和概念

ELF相关术语

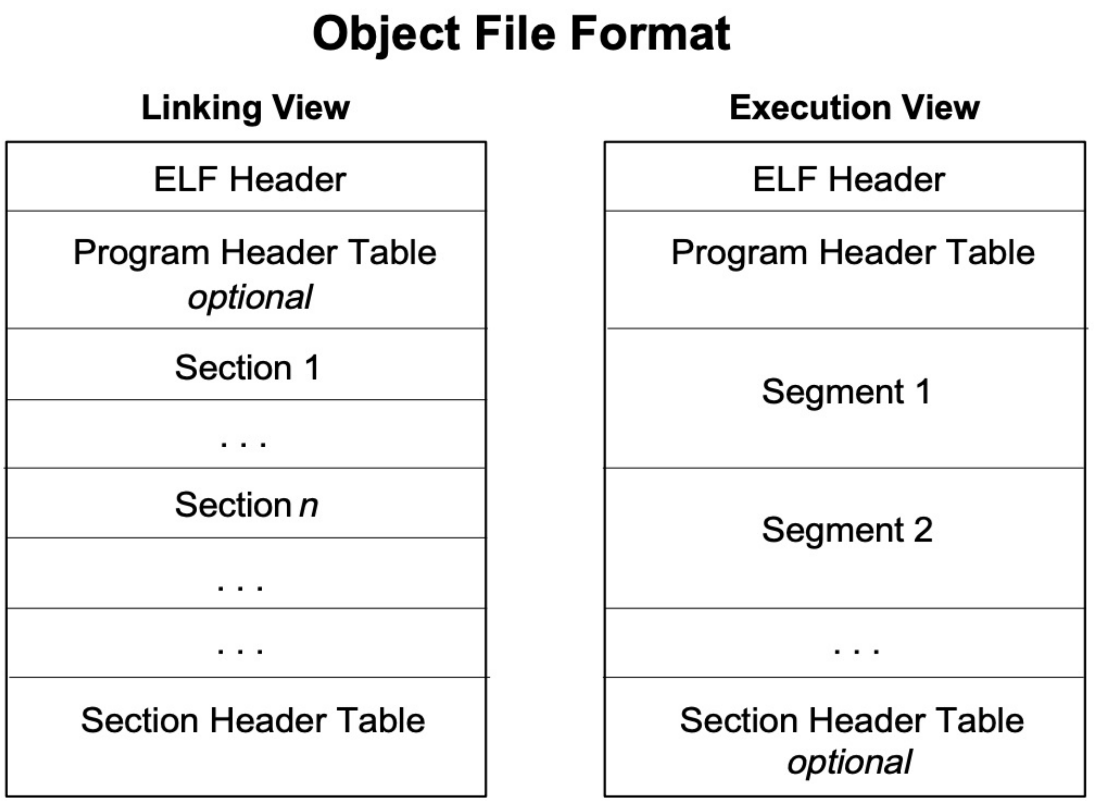
- ELF相关术语
  - Table表
    - GOT=Global Offset Table
    - SHT=Section Header Table
    - PLT=Procedure Linkage Table
    - PHT=Program Header Table
      - the kernel knows which section goes to which segment
  - 文件格式
    - COFF=Common object file format
    - 其他文件格式
      - Mach-O=Mach object file format
      - PE=Portable executable
  - BSS=Block Started by Symbol
    - The uninitialized data segment containing statically-allocated variables
  - DWARF
    - A standardized debugging data format
  - PC=Program counter
    - On x86, this is the same as IP (Instruction Pointer) register
  - section
    - SHF=Section header Flag
    - shstrtab = section header string table
  - 地址
    - RVA=Relative virtual address
    - VMA=Virtual Memory Area/Address
  - 加载
    - PIC=Position independent code
    - PIE=Position independent executable
    - REL=RELA=Relocation
  - TLS=Thread-Local Storage
    - DTV=Dynamic thread vector
    - access models
      - GD=Global Dynamic
        - dynamic TLS
      - IE=Initial Executable
        - static TLS with assigned offsets
      - LD=Local Dynamic
        - dynamic TLS of local symbols
      - LE=Local Executable
        - static TLS

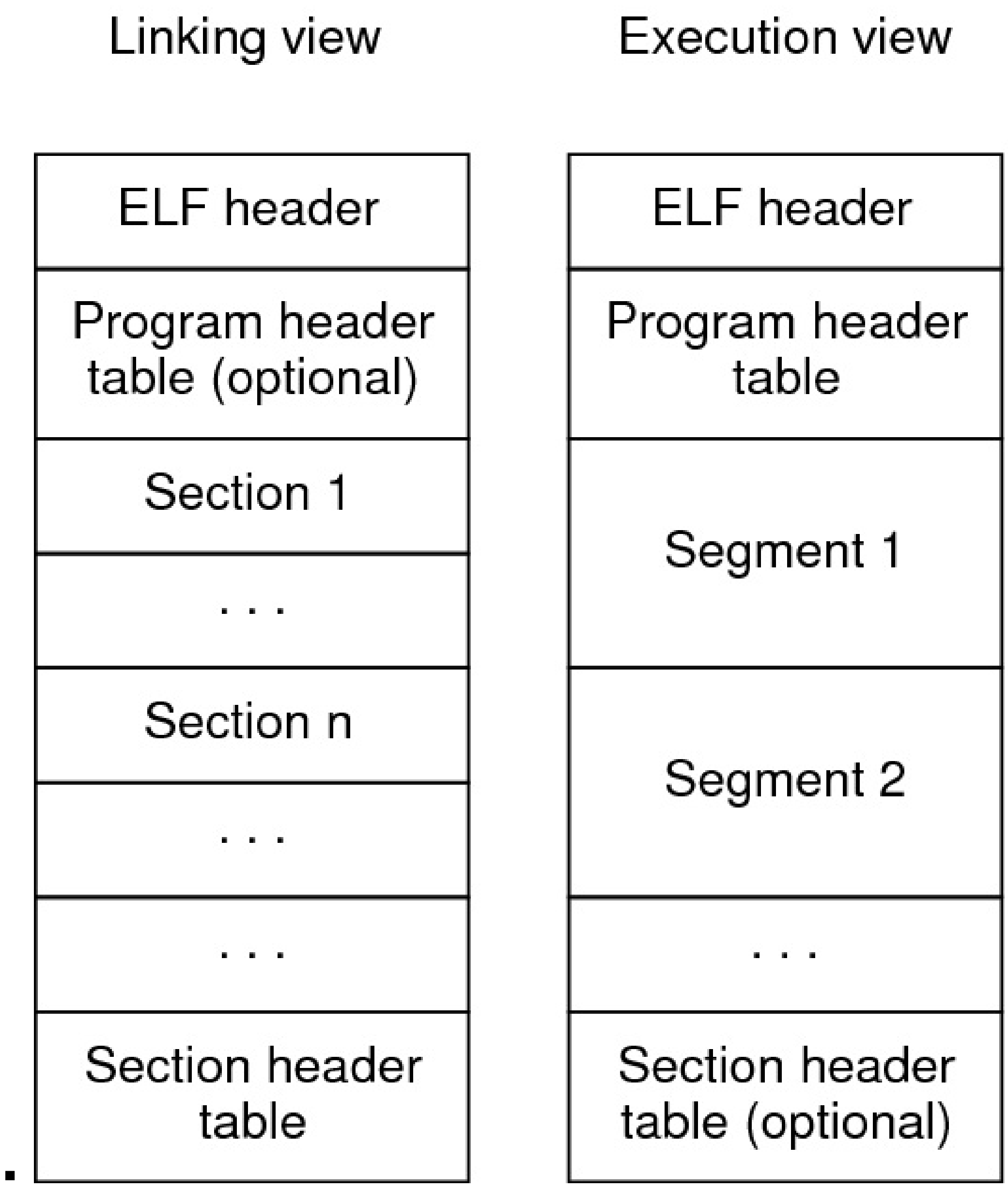
ELF相关概念

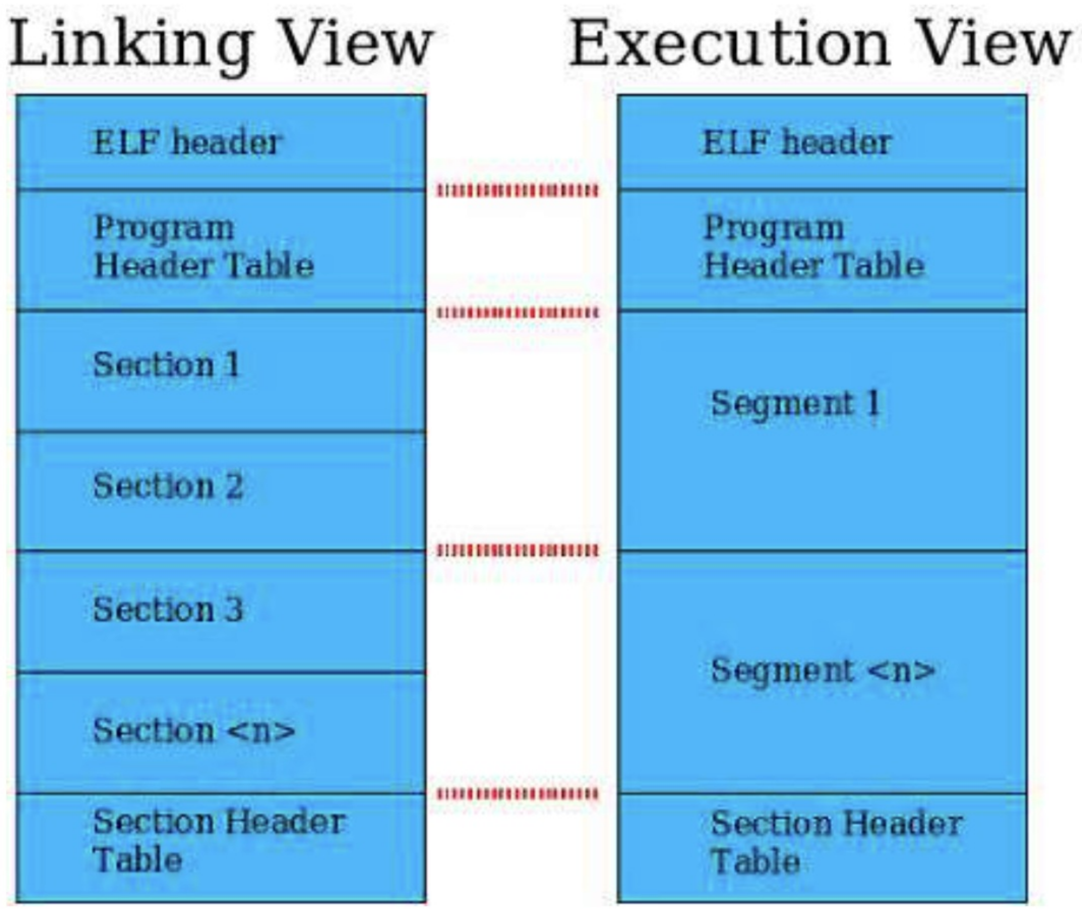
- ELF相关概念
  - section
    - 不同的section
      - .text=代码段
      - .data=数据段：全局变量
      - .bss：未初始化的数据值
        - 【整理】ELF相关：.bss节
  - segment ~= VMA
    - Linux内核内部的概念
    - contains virtually contiguous page frame

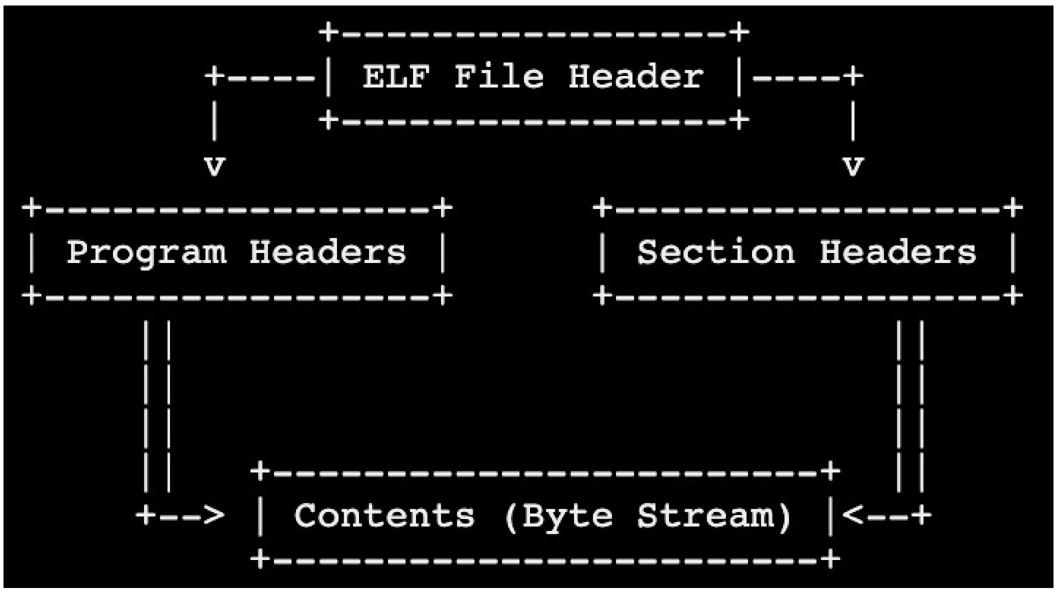
ELF结构详解

ELF的2种视图

- ELF的2种视图views
  - 概述
    - Linking View
      - Linking链接时：需要Section Header Table，不需要Program Header Table
    - Execution View
      - Execution执行时：需要Program Header Table，不需要Section Header Table
  - ELF的2种视图
    - 


Linking view
Execution view
ELF header
Program header table (optional)
Section 1
. . .
Section n
. . .
. . .
Section header table
ELF header
Program header table
Segment 1
Segment 2
. . .
Section header table (optional)

- 

- 文字版
  - 

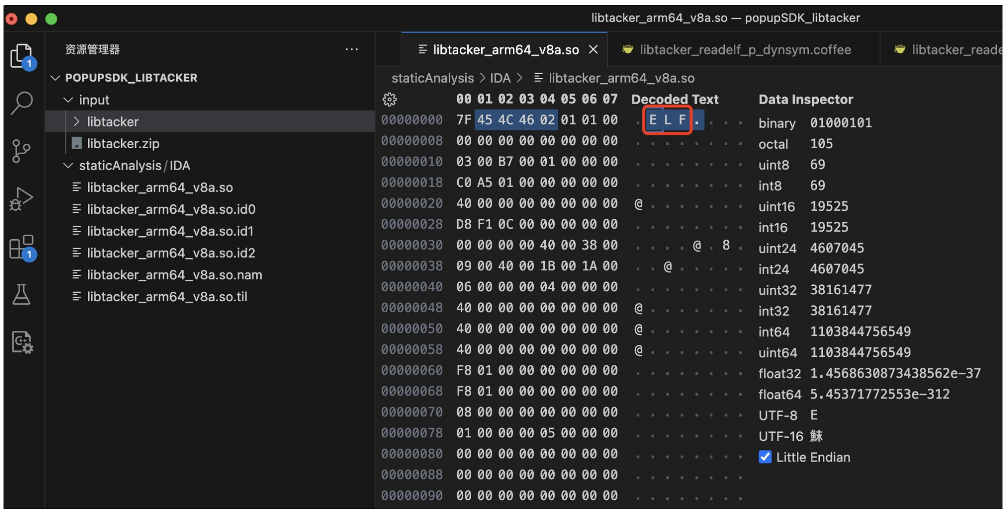
```
               +-----------------+
          +----| ELF File Header |----+
          |    +-----------------+    |
          v                           v
  +-----------------+       +-----------------+
  | Program Headers |       | Section Headers |
  +-----------------+       +-----------------+
         ||                         ||
         ||                         ||
         ||                         ||
         ||   +------------------------+   ||
         +--> | Contents (Byte Stream) |<--+
              +------------------------+
```

```
                +-----------------+
           +----| ELF File Header |----+
           |    +-----------------+    |
           v                           v
   +-----------------+       +-----------------+
   | Program Headers |       | Section Headers |
   +-----------------+       +-----------------+
          ||                         ||
          ||                         ||
          ||                         ||
        ||    +------------------------+  ||
       +-->  |  Contents (Byte Stream)  |<--+
              +------------------------+
```

ELF结构图

- ELF结构布局图=ELF layout
    -
    - 文字版

```
      +--------------------------------+
      | ELF File Header                |
      +--------------------------------+
      | Program Header for segment #1 |
      +--------------------------------+
      | Program Header for segment #2 |
      +--------------------------------+
      | ...                            |
      +--------------------------------+
      | Contents (Byte Stream)         |
      | ...                            |
      +--------------------------------+
      | Section Header for section #1 |
+--------------------------------+
| Section Header for section #2 |
+--------------------------------+
| ...                            |
+--------------------------------+
| ".shstrtab" section            |
+--------------------------------+
| ".symtab"   section            |
+--------------------------------+
| ".strtab"   section            |
+--------------------------------+
```

- 举例
  - 打开so可以看到顶部有ELF字样
    - 

libtacker_arm64_v8a.so — popupSDK_libtacker
资源管理器
POPUPSDK_LIBTACKER
input
libtacker
libtacker.zip
staticAnalysis / IDA
libtacker_arm64_v8a.so
libtacker_arm64_v8a.so.id0
libtacker_arm64_v8a.so.id1
libtacker_arm64_v8a.so.id2
libtacker_arm64_v8a.so.nam
libtacker_arm64_v8a.so.til
libtacker_arm64_v8a.so
libtacker_readelf_p_dynsym.coffee
libtacker_reade
staticAnalysis > IDA > libtacker_arm64_v8a.so

```
          00 01 02 03 04 05 06 07  Decoded Text
00000000  7F 45 4C 46 02 01 01 00  . E L F . . . .
00000008  00 00 00 00 00 00 00 00  . . . . . . . .
00000010  03 00 B7 00 01 00 00 00  . . . . . . . .
00000018  C0 A5 01 00 00 00 00 00  . . . . . . . .
00000020  40 00 00 00 00 00 00 00  @ . . . . . . .
00000028  D8 F1 0C 00 00 00 00 00  . . . . . . . .
00000030  00 00 00 00 40 00 38 00  . . . . @ . 8 .
00000038  09 00 40 00 1B 00 1A 00  . . @ . . . . .
00000040  06 00 00 00 04 00 00 00  . . . . . . . .
00000048  40 00 00 00 00 00 00 00  @ . . . . . . .
00000050  40 00 00 00 00 00 00 00  @ . . . . . . .
00000058  40 00 00 00 00 00 00 00  @ . . . . . . .
00000060  F8 01 00 00 00 00 00 00  . . . . . . . .
00000068  F8 01 00 00 00 00 00 00  . . . . . . . .
00000070  08 00 00 00 00 00 00 00  . . . . . . . .
00000078  01 00 00 00 05 00 00 00  . . . . . . . .
00000080  00 00 00 00 00 00 00 00  . . . . . . . .
00000088  00 00 00 00 00 00 00 00  . . . . . . . .
00000090  00 00 00 00 00 00 00 00  . . . . . . . .
```

```
Data Inspector
binary   01000101
octal    105
uint8    69
int8     69
uint16   19525
int16    19525
uint24   4607045
int24    4607045
uint32   38161477
int32    38161477
int64    1103844756549
uint64   1103844756549
float32  1.4568630873438562e-37
float64  5.45371772553e-312
UTF-8    E
UTF-16   鱅
✓ Little Endian
```

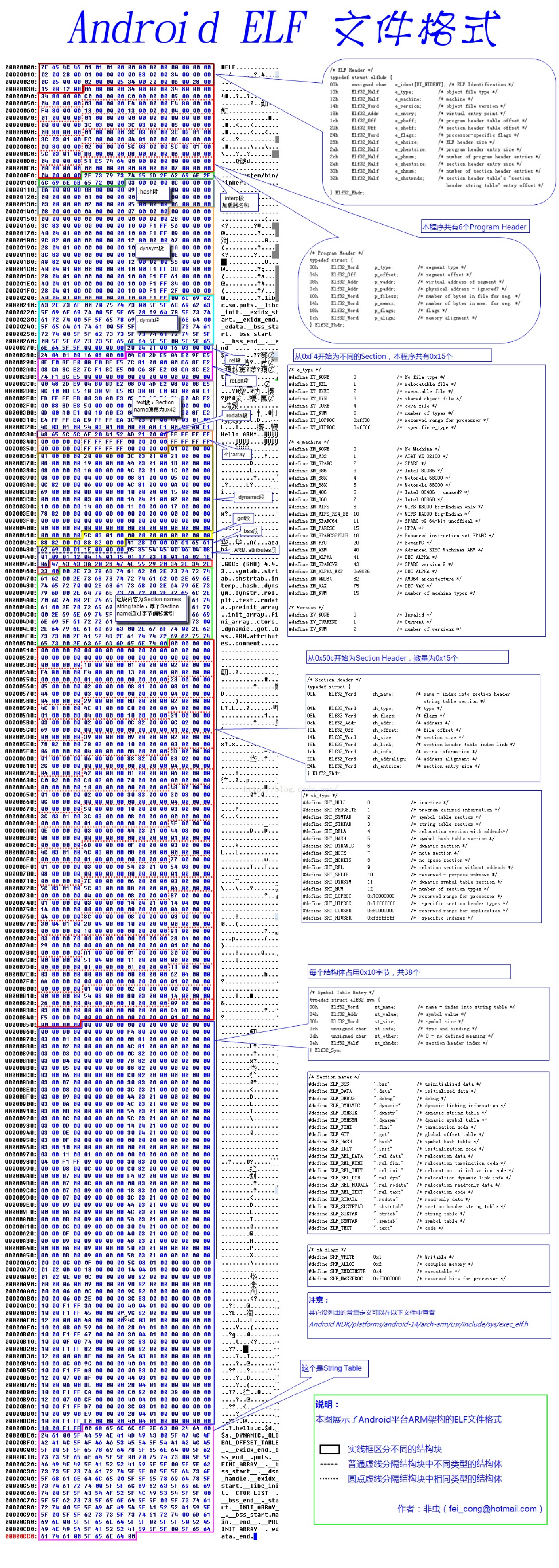
-

  -

举例

Android ELF 文件格式

- Android ELF 文件格式
  -



Android ELF 文件格式
interp段 加载器名称
hash段
dynsym段
dynstr段
rel段
rel.plt段
text段，Section name偏移为0x42
rodata段
4个array
dynamic段
got段
bss段
ARM attributes段
这段内容为Section names string table，每个Section name通过字节偏移索引
本程序共有6个Program Header
从0xF4开始为不同的Section，本程序共有0x15个
从0x50c开始为Section Header，数量为0x15个
每个结构体占用0x10字节，共38个
这个是String Table

```
/* ELF Header */
typedef struct elfhdr {
00h     unsigned char   e_ident[EI_NIDENT]; /* ELF Identification */
10h     Elf32_Half      e_type;         /* object file type */
12h     Elf32_Half      e_machine;      /* machine */
14h     Elf32_Word      e_version;      /* object file version */
18h     Elf32_Addr      e_entry;        /* virtual entry point */
1ch     Elf32_Off       e_phoff;        /* program header table offset */
20h     Elf32_Off       e_shoff;        /* section header table offset */
24h     Elf32_Word      e_flags;        /* processor-specific flags */
28h     Elf32_Half      e_ehsize;       /* ELF header size */
2ah     Elf32_Half      e_phentsize;    /* program header entry size */
2ch     Elf32_Half      e_phnum;        /* number of program header entries */
2eh     Elf32_Half      e_shentsize;    /* section header entry size */
30h     Elf32_Half      e_shnum;        /* number of section header entries */
32h     Elf32_Half      e_shstrndx;     /* section header table's "section
                                           header string table" entry offset */
} Elf32_Ehdr;
```

```
/* Program Header */
typedef struct {
00h     Elf32_Word      p_type;         /* segment type */
04h     Elf32_Off       p_offset;       /* segment offset */
08h     Elf32_Addr      p_vaddr;        /* virtual address of segment */
0ch     Elf32_Addr      p_paddr;        /* physical address - ignored? */
10h     Elf32_Word      p_filesz;       /* number of bytes in file for seg. */
14h     Elf32_Word      p_memsz;        /* number of bytes in mem. for seg. */
18h     Elf32_Word      p_flags;        /* flags */
1ch     Elf32_Word      p_align;        /* memory alignment */
} Elf32_Phdr;
```

```
/* e_type */
#define ET_NONE         0               /* No file type */
#define ET_REL          1               /* relocatable file */
#define ET_EXEC         2               /* executable file */
#define ET_DYN          3               /* shared object file */
#define ET_CORE         4               /* core file */
#define ET_NUM          5               /* number of types */
#define ET_LOPROC       0xff00          /* reserved range for processor */
#define ET_HIPROC       0xffff          /*  specific e_type */

/* e_machine */
#define EM_NONE         0               /* No Machine */
#define EM_M32          1               /* AT&T WE 32100 */
#define EM_SPARC        2               /* SPARC */
#define EM_386          3               /* Intel 80386 */
#define EM_68K          4               /* Motorola 68000 */
#define EM_88K          5               /* Motorola 88000 */
#define EM_486          6               /* Intel 80486 - unused? */
#define EM_860          7               /* Intel 80860 */
#define EM_MIPS         8               /* MIPS R3000 Big-Endian only */
#define EM_MIPS_RS4_BE  10              /* MIPS R4000 Big-Endian */
#define EM_SPARC64      11              /* SPARC v9 64-bit unoffical */
#define EM_PARISC       15              /* HPPA */
#define EM_SPARC32PLUS  18              /* Enhanced instruction set SPARC */
#define EM_PPC          20              /* PowerPC */
#define EM_ARM          40              /* Advanced RISC Machines ARM */
#define EM_ALPHA        41              /* DEC ALPHA */
#define EM_SPARCV9      43              /* SPARC version 9 */
#define EM_ALPHA_EXP    0x9026          /* DEC ALPHA */
#define EM_AMD64        62              /* AMD64 architecture */
#define EM_VAX          75              /* DEC VAX */
#define EM_NUM          15              /* number of machine types */

/* Version */
#define EV_NONE         0               /* Invalid */
#define EV_CURRENT      1               /* Current */
#define EV_NUM          2               /* number of versions */
```

```
/* Section Header */
typedef struct {
00h     Elf32_Word      sh_name;        /* name - index into section header
                                           string table section */
04h     Elf32_Word      sh_type;        /* type */
08h     Elf32_Word      sh_flags;       /* flags */
0ch     Elf32_Addr      sh_addr;        /* address */
10h     Elf32_Off       sh_offset;      /* file offset */
14h     Elf32_Word      sh_size;        /* section size */
18h     Elf32_Word      sh_link;        /* section header table index link */
1ch     Elf32_Word      sh_info;        /* extra information */
20h     Elf32_Word      sh_addralign;   /* address alignment */
24h     Elf32_Word      sh_entsize;     /* section entry size */
} Elf32_Shdr;
```

```
/* sh_type */
#define SHT_NULL        0               /* inactive */
#define SHT_PROGBITS    1               /* program defined information */
#define SHT_SYMTAB      2               /* symbol table section */
#define SHT_STRTAB      3               /* string table section */
#define SHT_RELA        4               /* relocation section with addends*/
#define SHT_HASH        5               /* symbol hash table section */
#define SHT_DYNAMIC     6               /* dynamic section */
#define SHT_NOTE        7               /* note section */
#define SHT_NOBITS      8               /* no space section */
#define SHT_REL         9               /* relation section without addends */
#define SHT_SHLIB       10              /* reserved - purpose unknown */
#define SHT_DYNSYM      11              /* dynamic symbol table section */
#define SHT_NUM         12              /* number of section types */
#define SHT_LOPROC      0x70000000      /* reserved range for processor */
#define SHT_HIPROC      0x7fffffff      /*  specific section header types */
#define SHT_LOUSER      0x80000000      /* reserved range for application */
#define SHT_HIUSER      0xffffffff      /*  specific indexes */
```

```
/* Symbol Table Entry */
typedef struct elf32_sym {
00h     Elf32_Word      st_name;        /* name - index into string table */
04h     Elf32_Addr      st_value;       /* symbol value */
08h     Elf32_Word      st_size;        /* symbol size */
0ch     unsigned char   st_info;        /* type and binding */
0dh     unsigned char   st_other;       /* 0 - no defined meaning */
0eh     Elf32_Half      st_shndx;       /* section header index */
} Elf32_Sym;
```

```
/* Section names */
#define ELF_BSS         ".bss"          /* uninitialized data */
#define ELF_DATA        ".data"         /* initialized data */
#define ELF_DEBUG       ".debug"        /* debug */
#define ELF_DYNAMIC     ".dynamic"      /* dynamic linking information */
#define ELF_DYNSTR      ".dynstr"       /* dynamic string table */
#define ELF_DYNSYM      ".dynsym"       /* dynamic symbol table */
#define ELF_FINI        ".fini"         /* termination code */
#define ELF_GOT         ".got"          /* global offset table */
#define ELF_HASH        ".hash"         /* symbol hash table */
#define ELF_INIT        ".init"         /* initialization code */
#define ELF_REL_DATA    ".rel.data"     /* relocation data */
#define ELF_REL_FINI    ".rel.fini"     /* relocation termination code */
#define ELF_REL_INIT    ".rel.init"     /* relocation initialization code */
#define ELF_REL_DYN     ".rel.dyn"      /* relocaltion dynamic link info */
#define ELF_REL_RODATA  ".rel.rodata"   /* relocation read-only data */
#define ELF_REL_TEXT    ".rel.text"     /* relocation code */
#define ELF_RODATA      ".rodata"       /* read-only data */
#define ELF_SHSTRTAB    ".shstrtab"     /* section header string table */
#define ELF_STRTAB      ".strtab"       /* string table */
#define ELF_SYMTAB      ".symtab"       /* symbol table */
#define ELF_TEXT        ".text"         /* code */
```

```
/* sh_flags */
#define SHF_WRITE       0x1             /* Writable */
#define SHF_ALLOC       0x2             /* occupies memory */
#define SHF_EXECINSTR   0x4             /* executable */
#define SHF_MASKPROC    0xf0000000      /* reserved bits for processor */
```

注意：

其它没列出的常量定义可以在以下文件中查看

Android NDK/platforms/android-14/arch-arm/usr/include/sys/exec_elf.h

说明：

本图展示了Android平台ARM架构的ELF文件格式

▭	实线框区分不同的结构块
-----	普通虚线分隔结构块中不同类型的结构体
·······	圆点虚线分隔结构块中相同类型的结构体

作者：非虫（fei_cong@hotmail.com）

ELF链接和执行过程

- ELF链接和执行过程 = ELF Executable and Linkable Format diagram
    -

section节

ELF中常见的section节

- `.bss` ： Uninitialized global data ("Block Started by Symbol")
- `.comment` ： A series of NULL-terminated strings containing compiler information.
- `.ctors` ： Pointers to functions which are marked as `__attribute__ ((constructor))` as well as static C++ objects' constructors. They will be used by `__libc_global_ctors` function.
- `.data` ： Initialized data
- `.data.rel.ro` ： Similar to `.data` section, but this section should be made Read-Only after relocation is done.
- `.debug_XXX` ： Debugging information (for the programs which are compiled with `-g` option) which is in the DWARF 2.0 format.
- `.dtors` ： Pointers to functions which are marked as `__attribute__ ((destructor))` as well as static C++ objects' destructors.
- `.dynamic` ： For dynamic binaries, this section holds dynamic linking information used by `ld.so`
- `.dynstr` ： NULL-terminated strings of names of symbols in `.dynsym` section.
  - One can use commands such as `readelf -p .dynstr a.out` to see these strings.
- `.dynsym` ： Runtime/Dynamic symbol table. For dynamic binaries, this section is the symbol table of globally visible symbols. For example, if a dynamic link library wants to export its symbols, these symbols will be stored here. On the other hand, if a dynamic executable binary uses symbols from a dynamic link library, then these symbols are stored here too.
  - The symbol names (as NULL-terminated strings) are stored in `.dynstr` section.
- `.eh_frame` / `eh_frame_hdr` ： Frame unwind information (`EH` = `Exception Handling`).
  - To see the content of .eh_frame section, use `readelf --debug-dump=frames-interp a.out`
- `.fini` ： Code which will be executed when program exits normally
- `.fini_array` ： Pointers to functions which will be executed when program exits normally
- `.GCC.command.line` ： A series of NULL-terminated strings containing GCC command-line (that is used to compile the code) options.
  - This feature is supported since GCC 4.5 and the program must be compiled with `-frecord-gcc-switches` option.
- `.gnu.hash` ： GNU's extension to hash table for symbols.
  - The link editor ld calls `bfd_elf_gnu_hash` in in GNU Binutil's source file `bfd/elf.c` to compute the hash value.
  - The runtime linker ld.so calls do_lookup_x in elf/dl-lookup.c to do the symbol look-up. The hash computing function here is dl_new_hash.
- `.gnu.linkonceXXX` ： GNU's extension. It means only a single copy of the section will be used in linking. This is used to by g++. g++ will emit each template expansion in its own section. The symbols will be defined as weak, so that multiple definitions are permitted.
- `.gnu.version` ： Versions of symbols.
- `.gnu.version_d` ： Version definitions of symbols.
- `.gnu.version_r` ： Version references (version needs) of symbols.
- `.got` ： For dynamic binaries, this `Global Offset Table` holds the addresses of variables which

are relocated upon loading
- `.got.plt`：For dynamic binaries, this `Global Offset Table` holds the addresses of functions in dynamic libraries. They are used by trampoline code in `.plt` section. If `.got.plt` section is present, it contains at least three entries, which have special meanings
- `.hash`：Hash table for symbols.
  - The link editor ld calls bfd_elf_hash in in GNU Binutil's source file bfd/elf.c to compute the hash value.
  - The runtime linker ld.so calls do_lookup_x in elf/dl-lookup.c to do the symbol look-up. The hash computing function here is _dl_elf_hash.
- `.init`：Code which will be executed when program initializes
- `.init_array`：Pointers to functions which will be executed when program starts
- `.interp`：For dynamic binaries, this holds the full pathname of runtime linker `ld.so`
- `.jcr`：Java class registration information.
  - Like `.ctors` section, it contains a list of addresses which will be used by `_Jv_RegisterClasses` function in `CRT` (`C Runtime`) startup files (see gcc/crtstuff.c in GCC's source tree)
- `.note.ABI-tag`：This Linux-specific section is structured as a note section in ELF specification
- `.note.gnu.build-id`：A unique build ID
- `.note.GNU-stack`：see Executable stack
- `.nvFatBinSegment`：This segment contains information of nVidia's `CUDA` fat binary container. Its format is described by struct `__cudaFatCudaBinaryRec` in `__cudaFatFormat.h`
- `.plt`：For dynamic binaries, this `Procedure Linkage Table` holds the trampoline/linkage code. See paragraphs below.
- `.preinit_array`：Similar to .init_array section
- `.rela.dyn`：Runtime/Dynamic relocation table.
  - For dynamic binaries, this relocation table holds information of variables which must be relocated upon loading. Each entry in this table is a struct `Elf64_Rela` (see `/usr/include/elf.h`) which has only three members:
    - offset (the variable's [usually position-independent] virtual memory address which holds the "patched" value during the relocation process)
    - info (Index into .dynsym section and Relocation Type)
    - addend
- `.rela.plt`：Runtime/Dynamic relocation table.
  - This relocation table is similar to the one in .rela.dyn section; the difference is this one is for functions, not variables.
  - The relocation type of entries in this table is R_386_JMP_SLOT or R_X86_64_JUMP_SLOT and the "offset" refers to memory addresses which are inside .got.plt section.
  - Simply put, this table holds information to relocate entries in .got.plt section.
- `.rel.text` / `rela.text`：Compile-time/Static relocation table.
  - For programs compiled with `-c` option, this section provides information to the link editor `ld` where and how to "patch" executable code in `.text` section.
  - The difference between `.rel.text` and `.rela.text` is entries in the former does not have addend member. (Compare struct `Elf64_Rel` with struct `Elf64_Rela` in `/usr/include/elf.h`) Instead, the addend is taken from the memory location described by offset member

  - Whether to use `.rel` or `.rela` is platform-dependent. For x86_32, it is .rel and for x86_64, .rela
- `.rel.XXX` : Compile-time/Static relocation table for other sections. For example, .rela.init_array is the relocation table for `.init_array` section.
- `.rodata` : Read-only data.
- `.shstrtab` : NULL-terminated strings of section names.
  - One can use commands such as `readelf -p .shstrtab a.out` to see these strings.
- `.strtab` : NULL-terminated strings of names of symbols in `.symtab` section.
  - One can use commands such as `readelf -p .strtab a.out` to see these strings.
- `.symtab` : Compile-time/Static symbol table.
  - This is the main symbol table used in compile-time linking or runtime debugging.
  - The symbol names (as NULL-terminated strings) are stored in `.strtab` section.
  - Both `.symtab` and `.symtab` can be stripped away by the `strip` command.
- `.tbss` : Similar to `.bss` section, but for `Thread-Local` data.
- `.tdata` : Similar to `.data` section, but for `Thread-Local` data
- `.text` : User's executable code

举例

源码和编译后对应section

- 源码和编译后对应section
  -

.bss

Uninitialized global data ("Block Started by Symbol").

Depending on the compilers, uninitialized global variables could be stored in a nameness section called COMMON (named after Fortran 77's "common blocks".) To wit, consider the following code:

```
int globalVar;
static int globalStaticVar;
void dummy() {
   static int localStaticVar;
}
```

Compile with gcc -c, then on x86_64, the resulting object file has the following structure:

```
$ objdump -t foo.o

SYMBOL TABLE:
 ....
0000000000000000 l     O .bss   0000000000000004 globalStaticVar
0000000000000004 l     O .bss   0000000000000004 localStaticVar.1619
 ....
0000000000000004       O *COM*  0000000000000004 globalVar
```

so only the file-scope and local-scope global variables are in the .bss section. If one wants globalVar to reside in the .bss section, use the -fno-common compiler command-line option. Using -fno-common is encouraged, as the following example shows:

```
$ cat foo.c
int globalVar;
$ cat bar.c
double globalVar;
int main(){}
$ gcc foo.c bar.c
```

Not only there is no error message about redefinition of the same symbol in both source files (notice we did not use the extern keyword here), there is no complaint about their different data types and sizes either. However, if one uses -fno-common, the compiler will complain:

```
/tmp/ccM71JR7.o:(.bss+0x0): multiple definition of `globalVar'
/tmp/ccIbS5MO.o:(.bss+0x0): first defined here
ld: Warning: size of symbol `globalVar' changed from 8 in /tmp/ccIbS5MO.o to 4 in /
tmp/ccM71JR7.o
```

Segment段

- `segment` = `段` 的类型和含义：
  - `DYNAMIC` ：对于动态二进制，此段保存了动态链接信息
    - =ELF的链接视图时的： `.dynamic` 节
  - `GNU_EH_FRAME` ：Frame unwind information (EH = Exception Handling). This segment is usually the same as .eh_frame_hdr section in ELF's linking view.
  - `GNU_RELRO` ：This segment indicates the memory region which should be made Read-Only after relocation is done. This segment usually appears in a dynamic link library and it contains .ctors, .dtors, .dynamic, .got sections. See paragraph below.
  - `GNU_STACK` ：The permission flag of this segment indicates whether the stack is executable or not. This segment does not have any content; it is just an indicator.
  - `INTERP` ：For dynamic binaries, this holds the full pathname of runtime linker `ld.so`
    - =ELF的链接视图时的： `.interp` 节
  - `LOAD` ：Loadable program segment. Only segments of this type are loaded into memory during execution.
  - `NOTE` ：Auxiliary information.
    - For core dumps, this segment contains the status of the process (when the core dump is created), such as the signal (the process received and caused it to dump core), pending & held signals, process ID, parent process ID, user ID, nice value, cumulative user & system time, values of registers (including the program counter!)
    - For more info, see struct elf_prstatus and struct elf_prpsinfo in Linux kernel source file include/linux/elfcore.h and struct user_regs_struct in arch/x86/include/asm/user_64.h
  - `TLS` ：Thread-Local Storage

ELF解析工具

读取信息

Linux通用

- readelf
- objdump
- rabin2

readelf对比objdump

- `readelf` 并不提供反汇编功能
- `readelf` 可以显示调试信息
- `objdump` 使用了bfd库进行文件读取

readelf

- `readelf`
    - 是什么：用来**read**读取**elf**格式的工具
    - 作用：可以查看ELF格式文件的各种详细内容

安装readelf

- `readelf` 是 `binutils` 中的其中一个工具

Mac中安装binutils

- 安装binutils
    - Intel Mac

      ```
      brew install binutils
      ```

    - Apple Silicon Mac

      ```
      arch -arm64 /opt/homebrew/bin/brew install binutils
      ```

readelf用法

概述：

- 单个参数
    - 显示头信息
        - file-header
            - `-h` = `--file-header`：显示ELF文件头信息

              ```
              readelf -h elfFile
              ```

        - program-headers
            - `-l` = `--program-headers` = `--segments`：显示程序的头信息和段信息

              ```
              readelf -l elfFile
              ```

        - section-headers
            - `-S` = `--section-headers` = `--sections`：显示节的头信息

              ```
              readelf -S elfFile
              ```

    - `-s` = `--syms` = `--symbols`：显示符号表

      ```
      readelf -s elfFile
      ```

      -sV：显示符号表且带版本信息
    - `-r` = `--relocs`：显示重定位信息

      ```
      readelf -r elfFile
      ```

    - 打印信息
        - 以 `hex` = `十六进制` = `二进制` 方式
            - `-x` = `--hex-dump=<number|name>`

              ```
              readelf -x .dynsym elfFile
              readelf -x 8 elfFile
              ```

        - 以 `string` = `字符串` 形式
            - `-p` = `--string-dump=<number|name>`

              ```
              readelf -p .dynsym elfFile
              readelf -p 8 elfFile
              ```

- 组合参数
    - `-e` = `--headers` = `-h -l -S`

      ```
      readelf -e elfFile
      ```

- `-a` = `--all` == `-h -l -S -s -r -d -V -A -I`

  ```
  readelf -a elfFile
  ```

- `-sV` = `--syms --version-info`：显示符号表且带版本信息

  ```
  readelf -sV elfFile
  ```

readelf用法举例

举例说明：

- 输入文件： `libtacker.so`

  ```
  ➜  arm64-v8a pwd
  /Users/crifan/dev/dev_root/androidReverse/popupSDK_libtacker/input/libtacker/arm64-
  v8a
  ➜  arm64-v8a ll
  total 1664
  -rw-------@ 1 crifan  staff   830K  6 29 22:27 libtacker.so
  ```

readelf解析ELF格式的 `libtacker.so` 的具体效果如下：

-h：显示ELF文件头信息

```
➜  arm64-v8a readelf -h libtacker.so
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           AArch64
  Version:                           0x1
  Entry point address:               0x1a5c0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          848344 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         27
  Section header string table index: 26
```

- 说明
  - Class： `ELF64`
  - Type: `DYN` (`Shared object file`)
    - 动态链接库
  - Machine: `AArch64` == `arm64`
  - Entry point address: `0x1a5c0`
    - 入口地址，应该就是之前的： `_start` 入口函数
  - Number of program headers: `9`
    - 程序有9个header

- Number of section headers: `27`
    - 有27个section header

-l：显示程序头信息和段信息

```
➜ arm64-v8a readelf -l libtacker.so


Elf file type is DYN (Shared object file)
Entry point 0x1a5c0
There are 9 program headers, starting at offset 64


Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000000040 0x0000000000000040
                 0x00000000000001f8 0x00000000000001f8  R      0x8
  LOAD           0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x00000000000c9520 0x00000000000c9520  R E    0x1000
  LOAD           0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003510  RW     0x1000
  LOAD           0x00000000000cca30 0x00000000000cea30 0x00000000000cea30
                 0x00000000000025d8 0x00000000000030b0  RW     0x1000
  DYNAMIC        0x00000000000cc618 0x00000000000cd618 0x00000000000cd618
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  GNU_RELRO      0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003ae0  R      0x1
  GNU_EH_FRAME   0x000000000000fb2c 0x000000000000fb2c 0x000000000000fb2c
                 0x0000000000001dbc 0x0000000000001dbc  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x0
  NOTE           0x0000000000000238 0x0000000000000238 0x0000000000000238
                 0x00000000000000bc 0x00000000000000bc  R      0x4


 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.android.ident .note.gnu.build-id .dynsym .gnu.version .gnu.version_r .g
nu.hash .hash .dynstr .rela.dyn .rela.plt .gcc_except_table .rodata .eh_frame_hdr .eh_f
rame .text .plt
   02     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   03     .data .bss
   04     .dynamic
   05     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   06     .eh_frame_hdr
   07
   08     .note.android.ident .note.gnu.build-id
```

- 说明
    - `PHDR` ：保存程序头表 （Program header => PHDR）
    - `LOAD` ：表示一个需要从二进制文件映射到虚拟地址空间的段，其中保存了常量数据（如字符

串），程序目标代码等。
  - `DYNAMIC`：保存了由动态连接器（即INTERP段中指定的解释器）使用的信息。
  - `INTERP`：指定程序从可行性文件映射到内存之后，必须调用的解释器。它是通过链接其他库来满足未解析的引用，用于在虚拟地址空间中插入程序运行所需的动态库

-S：显示节的头信息

```
➜  arm64-v8a readelf -S libtacker.so
There are 27 section headers, starting at offset 0xcf1d8:


Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .note.androi[...] NOTE             0000000000000238  00000238
       0000000000000098  0000000000000000   A       0     0     4
  [ 2] .note.gnu.bu[...] NOTE             00000000000002d0  000002d0
       0000000000000024  0000000000000000   A       0     0     4
  [ 3] .dynsym           DYNSYM           00000000000002f8  000002f8
       0000000000000b10  0000000000000018   A       8     1     8
  [ 4] .gnu.version      VERSYM           0000000000000e08  00000e08
       00000000000000ec  0000000000000002   A       3     0     2
  [ 5] .gnu.version_r    VERNEED          0000000000000ef4  00000ef4
       0000000000000040  0000000000000000   A       8     2     4
  [ 6] .gnu.hash         GNU_HASH         0000000000000f38  00000f38
       00000000000001ec  0000000000000000   A       3     0     8
  [ 7] .hash             HASH             0000000000001124  00001124
       00000000000003b8  0000000000000004   A       3     0     4
  [ 8] .dynstr           STRTAB           00000000000014dc  000014dc
       0000000000000c19  0000000000000000   A       0     0     1
  [ 9] .rela.dyn         RELA             00000000000020f8  000020f8
       0000000000008850  0000000000000018   A       3     0     8
  [10] .rela.plt         RELA             000000000000a948  0000a948
       0000000000000450  0000000000000018  AI       3    22     8
  [11] .gcc_except_table PROGBITS         000000000000ad98  0000ad98
       0000000000001960  0000000000000000   A       0     0     4
  [12] .rodata           PROGBITS         000000000000c6f8  0000c6f8
       0000000000003434  0000000000000000 AMS       0     0     8
  [13] .eh_frame_hdr     PROGBITS         000000000000fb2c  0000fb2c
       0000000000001dbc  0000000000000000   A       0     0     4
  [14] .eh_frame         PROGBITS         00000000000118e8  000118e8
       0000000000008cd4  0000000000000000   A       0     0     8
  [15] .text             PROGBITS         000000000001a5c0  0001a5c0
       00000000000aec60  0000000000000000  AX       0     0     16
  [16] .plt              PROGBITS         00000000000c9220  000c9220
       0000000000000300  0000000000000000  AX       0     0     16
  [17] .data.rel.ro      PROGBITS         00000000000ca520  000c9520
       0000000000002eb8  0000000000000000  WA       0     0     8
  [18] .fini_array       FINI_ARRAY       00000000000cd3d8  000cc3d8
       0000000000000010  0000000000000000  WA       0     0     8
  [19] .init_array       INIT_ARRAY       00000000000cd3e8  000cc3e8
```

```
       0000000000000230  0000000000000000  WA       0     0     8
  [20] .dynamic          DYNAMIC          00000000000cd618  000cc618
       00000000000001d0  0000000000000010  WA       8     0     8
  [21] .got              PROGBITS         00000000000cd7e8  000cc7e8
       00000000000000c0  0000000000000000  WA       0     0     8
  [22] .got.plt          PROGBITS         00000000000cd8a8  000cc8a8
       0000000000000188  0000000000000000  WA       0     0     8
  [23] .data             PROGBITS         00000000000cea30  000cca30
       00000000000025d8  0000000000000000  WA       0     0     16
  [24] .bss              NOBITS           00000000000d1010  000cf008
       0000000000000ad0  0000000000000000  WA       0     0     16
  [25] .comment          PROGBITS         0000000000000000  000cf008
       00000000000000c6  0000000000000001  MS       0     0     1
  [26] .shstrtab         STRTAB           0000000000000000  000cf0ce
       0000000000000104  0000000000000000           0     0     1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings), I (info),
  L (link order), O (extra OS processing required), G (group), T (TLS),
  C (compressed), x (unknown), o (OS specific), E (exclude),
  D (mbind), p (processor specific)
```

-r：显示重定位信息

```
➜  arm64-v8a readelf -r libtacker.so
。。。
0000000cd3f0  007100000101 R_AARCH64_ABS64   000000000002b930 .datadiv_decode17[...] + 0

0000000cd430  007200000101 R_AARCH64_ABS64   0000000000039464 .datadiv_decode17[...] + 0

0000000cd498  007300000101 R_AARCH64_ABS64   000000000005c790 .datadiv_decode15[...] + 0

0000000cd4b0  007400000101 R_AARCH64_ABS64   000000000005ed50 .datadiv_decode15[...] + 0

0000000cd5d8  007500000101 R_AARCH64_ABS64   00000000000a5e74 .datadiv_decode54[...] + 0


Relocation section '.rela.plt' at offset 0xa948 contains 46 entries:
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
0000000cd8c0  000100000402 R_AARCH64_JUMP_SL 0000000000000000 __cxa_finalize@LIBC + 0
0000000cd8c8  000200000402 R_AARCH64_JUMP_SL 0000000000000000 __cxa_atexit@LIBC + 0
0000000cd8d0  000300000402 R_AARCH64_JUMP_SL 0000000000000000 __android_log_print + 0
0000000cd8d8  000400000402 R_AARCH64_JUMP_SL 0000000000000000 __stack_chk_fail@LIBC + 0
0000000cd8e0  000500000402 R_AARCH64_JUMP_SL 0000000000000000 memset@LIBC + 0
0000000cd8e8  000600000402 R_AARCH64_JUMP_SL 0000000000000000 strncpy@LIBC + 0
0000000cd8f0  000700000402 R_AARCH64_JUMP_SL 0000000000000000 strncat@LIBC + 0
0000000cd8f8  000800000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_self@LIBC + 0
0000000cd900  000900000402 R_AARCH64_JUMP_SL 0000000000000000 malloc@LIBC + 0
0000000cd908  000a00000402 R_AARCH64_JUMP_SL 0000000000000000 free@LIBC + 0
0000000cd910  000b00000402 R_AARCH64_JUMP_SL 0000000000000000 posix_memalign@LIBC + 0
0000000cd918  000d00000402 R_AARCH64_JUMP_SL 0000000000000000 vfprintf@LIBC + 0
0000000cd920  000e00000402 R_AARCH64_JUMP_SL 0000000000000000 fputc@LIBC + 0
0000000cd928  000f00000402 R_AARCH64_JUMP_SL 0000000000000000 vasprintf@LIBC + 0
0000000cd930  001000000402 R_AARCH64_JUMP_SL 0000000000000000 android_set_abort[...]@LI
BC + 0
```

```
0000000cd938  001100000402 R_AARCH64_JUMP_SL 0000000000000000 openlog@LIBC + 0
0000000cd940  001200000402 R_AARCH64_JUMP_SL 0000000000000000 syslog@LIBC + 0
0000000cd948  001300000402 R_AARCH64_JUMP_SL 0000000000000000 closelog@LIBC + 0
0000000cd950  001400000402 R_AARCH64_JUMP_SL 0000000000000000 abort@LIBC + 0
0000000cd958  001500000402 R_AARCH64_JUMP_SL 0000000000000000 strlen@LIBC + 0
0000000cd960  001600000402 R_AARCH64_JUMP_SL 0000000000000000 realloc@LIBC + 0
0000000cd968  001700000402 R_AARCH64_JUMP_SL 0000000000000000 memmove@LIBC + 0
0000000cd970  001800000402 R_AARCH64_JUMP_SL 0000000000000000 __memmove_chk@LIBC + 0
0000000cd978  001900000402 R_AARCH64_JUMP_SL 0000000000000000 __strlen_chk@LIBC + 0
0000000cd980  001a00000402 R_AARCH64_JUMP_SL 0000000000000000 memchr@LIBC + 0
0000000cd988  001b00000402 R_AARCH64_JUMP_SL 0000000000000000 __vsnprintf_chk@LIBC + 0
0000000cd990  001c00000402 R_AARCH64_JUMP_SL 0000000000000000 memcpy@LIBC + 0
0000000cd998  001d00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_mutex_lock@LIBC +
0
0000000cd9a0  001e00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_mutex_unlock@LIBC
 + 0
0000000cd9a8  001f00000402 R_AARCH64_JUMP_SL 0000000000000000 calloc@LIBC + 0
0000000cd9b0  002000000402 R_AARCH64_JUMP_SL 0000000000000000 strcmp@LIBC + 0
0000000cd9b8  002100000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_getspecific@LIBC
+ 0
0000000cd9c0  002200000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_once@LIBC + 0
0000000cd9c8  002300000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_setspecific@LIBC
+ 0
0000000cd9d0  002400000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_key_delete@LIBC +
0
0000000cd9d8  002500000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_key_create@LIBC +
0
0000000cd9e0  002600000402 R_AARCH64_JUMP_SL 0000000000000000 getauxval@LIBC + 0
0000000cd9e8  002700000402 R_AARCH64_JUMP_SL 0000000000000000 __system_property_get@LIB
C + 0
0000000cd9f0  002800000402 R_AARCH64_JUMP_SL 0000000000000000 strncmp@LIBC + 0
0000000cd9f8  002900000402 R_AARCH64_JUMP_SL 0000000000000000 fprintf@LIBC + 0
0000000cda00  002a00000402 R_AARCH64_JUMP_SL 0000000000000000 fflush@LIBC + 0
0000000cda08  002b00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_wrlock@LIB
C + 0
0000000cda10  002c00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_unlock@LIB
C + 0
0000000cda18  002d00000402 R_AARCH64_JUMP_SL 0000000000000000 dl_iterate_phdr@LIBC + 0
0000000cda20  002e00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_rdlock@LIB
C + 0
0000000cda28  002f00000402 R_AARCH64_JUMP_SL 0000000000000000 fwrite@LIBC + 0
```

看出来了，前面的：

- -r --relocs Display the relocations (if present)
  - 输出：
    - Relocation section '.rela.dyn' at offset 0x20f8 contains 1454 entries
    - Relocation section '.rela.plt' at offset 0xa948 contains 46 entries == Imports

-s：显示符号表

```
Symbol table '.dynsym' contains 118 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_f[...]@LIBC (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@LIBC (2)
     3: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __android_log_print
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __stack[...]@LIBC (2)
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memset@LIBC (2)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncpy@LIBC (2)
     7: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncat@LIBC (2)
     8: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_self@LIBC (2)
     9: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND malloc@LIBC (2)
    10: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND free@LIBC (2)
    11: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND posix_m[...]@LIBC (2)
    12: 0000000000000000     0 OBJECT  GLOBAL DEFAULT  UND __sF@LIBC (2)
    13: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vfprintf@LIBC (2)
    14: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fputc@LIBC (2)
    15: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vasprintf@LIBC (2)
    16: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND android[...]@LIBC (2)
    17: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND openlog@LIBC (2)
    18: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND syslog@LIBC (2)
    19: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND closelog@LIBC (2)
    20: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND abort@LIBC (2)
    21: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strlen@LIBC (2)
    22: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND realloc@LIBC (2)
    23: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memmove@LIBC (2)
    24: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __memmo[...]@LIBC (2)
    25: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __strlen_chk@LIBC (2)
    26: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memchr@LIBC (2)
    27: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __vsnpr[...]@LIBC (2)
    28: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memcpy@LIBC (2)
    29: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    30: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    31: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND calloc@LIBC (2)
    32: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strcmp@LIBC (2)
    33: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    34: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_once@LIBC (2)
    35: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    36: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    37: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    38: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getauxval@LIBC (2)
    39: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __syste[...]@LIBC (2)
    40: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncmp@LIBC (2)
    41: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fprintf@LIBC (2)
    42: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fflush@LIBC (2)
    43: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    44: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    45: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND dl_iter[...]@LIBC (3)
    46: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    47: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fwrite@LIBC (2)
    48: 0000000000044ce8  6608 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    49: 0000000000078a04  2696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
    50: 00000000000a8a58  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    51: 0000000000076128  2160 FUNC    GLOBAL DEFAULT   15 .datadiv_decode4[...]
    52: 000000000008f8e8  8740 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
...
    70: 000000000008e650  3772 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
```

```
     71: 00000000000381fc  3696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode3[...]
     72: 000000000006f220  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
     73: 00000000000a8884     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
     74: 00000000000aa438  1436 FUNC    GLOBAL DEFAULT   15 JNI_OnLoad
     75: 0000000000026d98 18656 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
     76: 0000000000033a2c 11972 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
...
    116: 000000000005ed50  6304 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    117: 00000000000a5e74     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
```

-sV：显示符号表且带版本信息

```
    -s --syms              Display the symbol table
    -V --version-info      Display the version sections (if present)
```

->

```
➜  arm64-v8a readelf -sV libtacker.so


Symbol table '.dynsym' contains 118 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_f[...]@LIBC (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@LIBC (2)
     3: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __android_log_print
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __stack[...]@LIBC (2)
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memset@LIBC (2)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncpy@LIBC (2)
     7: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncat@LIBC (2)
     8: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_self@LIBC (2)
     9: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND malloc@LIBC (2)
    10: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND free@LIBC (2)
    11: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND posix_m[...]@LIBC (2)
    12: 0000000000000000     0 OBJECT  GLOBAL DEFAULT  UND __sF@LIBC (2)
    13: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vfprintf@LIBC (2)
    14: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fputc@LIBC (2)
    15: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vasprintf@LIBC (2)
    16: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND android[...]@LIBC (2)
    17: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND openlog@LIBC (2)
    18: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND syslog@LIBC (2)
    19: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND closelog@LIBC (2)
    20: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND abort@LIBC (2)
    21: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strlen@LIBC (2)
    22: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND realloc@LIBC (2)
    23: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memmove@LIBC (2)
    24: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __memmo[...]@LIBC (2)
    25: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __strlen_chk@LIBC (2)
    26: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memchr@LIBC (2)
    27: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __vsnpr[...]@LIBC (2)
    28: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memcpy@LIBC (2)
    29: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    30: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    31: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND calloc@LIBC (2)
    32: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strcmp@LIBC (2)
    33: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    34: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_once@LIBC (2)
    35: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    36: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    37: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    38: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getauxval@LIBC (2)
    39: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __syste[...]@LIBC (2)
    40: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncmp@LIBC (2)
    41: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fprintf@LIBC (2)
    42: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fflush@LIBC (2)
    43: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    44: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    45: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND dl_iter[...]@LIBC (3)
    46: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    47: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fwrite@LIBC (2)
    48: 0000000000044ce8  6608 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    49: 0000000000078a04  2696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
    50: 00000000000a8a58  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    51: 0000000000076128  2160 FUNC    GLOBAL DEFAULT   15 .datadiv_decode4[...]
    52: 000000000008f8e8  8740 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
    53: 00000000000523ec 10992 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    54: 0000000000055f24 26664 FUNC    GLOBAL DEFAULT   15 .datadiv_decode3[...]
    55: 000000000005ca48  6228 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    56: 0000000000072b58 13248 FUNC    GLOBAL DEFAULT   15 .datadiv_decode6[...]
    57: 000000000009f204  6296 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
    58: 0000000000032490  4596 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    59: 00000000000642dc 21616 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    60: 000000000007eb38     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    61: 0000000000091c0c  3612 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    62: 0000000000099d00  3748 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    63: 000000000004a620 24816 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    64: 0000000000079c64 15120 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
    65: 0000000000089d58  8212 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    66: 000000000009bf84     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    67: 000000000007ee60  7596 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
    68: 0000000000083388 16340 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    69: 00000000000890a8  2684 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    70: 000000000008e650  3772 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    71: 00000000000381fc  3696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode3[...]
    72: 000000000006f220  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    73: 00000000000a8884     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    74: 00000000000aa438  1436 FUNC    GLOBAL DEFAULT   15 JNI_OnLoad
    75: 0000000000026d98 18656 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    76: 0000000000033a2c 11972 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    77: 000000000003c8dc  8072 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
    78: 00000000000783f8  1112 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    79: 00000000000402e0 18000 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    80: 0000000000050d58  3556 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    81: 00000000000a0f34 12764 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    82: 000000000002fa98  1716 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    83: 0000000000050850  1052 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
    84: 000000000005e3ac  1512 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    85: 000000000007058c  8700 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    86: 0000000000076aa0  5588 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    87: 000000000007d9ac  4264 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    88: 0000000000092dc0  6148 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    89: 000000000002e2bc  5636 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    90: 000000000007eac4     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    91: 0000000000087514  6800 FUNC    GLOBAL DEFAULT   15 .datadiv_decode4[...]
    92: 00000000000a4228  1888 FUNC    GLOBAL DEFAULT   15 .datadiv_decode6[...]
    93: 00000000000a6a74  7124 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    94: 0000000000046ba4 12236 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    95: 0000000000080db8  7204 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    96: 000000000008bfc4  4784 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    97: 0000000000094aac 10132 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    98: 000000000003b9c0  2832 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    99: 000000000006a31c 19572 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   100: 000000000008d4fc  3852 FUNC    GLOBAL DEFAULT   15 .datadiv_decode6[...]
   101: 000000000009af44  4076 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   102: 00000000000aa164     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
   103: 0000000000030db4  5388 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
   104: 0000000000060904 10444 FUNC    GLOBAL DEFAULT   15 .datadiv_decode2[...]
   105: 00000000000979f8  8736 FUNC    GLOBAL DEFAULT   15 .datadiv_decode2[...]
   106: 000000000009c3ec 11308 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   107: 00000000000a4be8  4636 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   108: 00000000000a5f84  2352 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
   109: 00000000000aa9d4  5616 FUNC    GLOBAL DEFAULT   15 .datadiv_decode2[...]
   110: 0000000000036ab8  3832 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   111: 000000000003b2ac     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   112: 000000000003b8fc     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode7[...]
   113: 000000000002b930 10168 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   114: 0000000000039464  7640 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   115: 000000000005c790     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   116: 000000000005ed50  6304 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   117: 00000000000a5e74     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]


Version symbols section '.gnu.version' contains 118 entries:
 Addr: 0x0000000000000e08  Offset: 0x00000e08  Link: 3 (.dynsym)
  000:   0 (*local*)       2 (LIBC)          2 (LIBC)          1 (*global*)
  004:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  008:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  00c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  010:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  014:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  018:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  01c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  020:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  024:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  028:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  02c:   2 (LIBC)          3 (LIBC)          2 (LIBC)          2 (LIBC)
  030:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  034:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  038:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  03c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  040:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  044:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  048:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  04c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  050:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  054:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  058:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  05c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  060:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  064:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  068:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  06c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  070:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  074:   1 (*global*)      1 (*global*)


Version needs section '.gnu.version_r' contains 2 entries:
 Addr: 0x0000000000000ef4  Offset: 0x00000ef4  Link: 8 (.dynstr)
  000000: Version: 1  File: libdl.so  Cnt: 1
  0x0020:   Name: LIBC  Flags: none  Version: 3
  0x0010: Version: 1  File: libc.so  Cnt: 1
  0x0030:   Name: LIBC  Flags: none  Version: 2
```

比普通的 `-s` ，多出了版本信息：

- .gnu.version
- .gnu.version_r

-x：以hex方式打印信息

举例1

对于：

```
Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
...
  [ 3] .dynsym           DYNSYM           00000000000002f8  000002f8
       0000000000000b10  0000000000000018   A       8     1     8
```

中的：

- `[ 3] .dynsym`

去打印信息：

```
➜  arm64-v8a readelf -x .dynsym libtacker.so

Hex dump of section '.dynsym':
  0x000002f8 00000000 00000000 00000000 00000000 ................
  0x00000308 00000000 00000000 01000000 12000000 ................
  0x00000318 00000000 00000000 00000000 00000000 ................
  0x00000328 10000000 12000000 00000000 00000000 ................
  0x00000338 00000000 00000000 1d000000 12000000 ................
  0x00000348 00000000 00000000 00000000 00000000 ................
  0x00000358 31000000 12000000 00000000 00000000 1...............
```

```
    0x00000368 00000000 00000000 34020000 12000000 ........4.......
    0x00000378 00000000 00000000 00000000 00000000 ................
    0x00000388 3b020000 12000000 00000000 00000000 ;...............
    0x00000398 00000000 00000000 43020000 12000000 ........C.......
    0x000003a8 00000000 00000000 00000000 00000000 ................
    0x000003b8 81090000 12000000 00000000 00000000 ................
    0x000003c8 00000000 00000000 030a0000 12000000 ................
    0x000003d8 00000000 00000000 00000000 00000000 ................
    0x000003e8 0a0a0000 12000000 00000000 00000000 ................
    0x000003f8 00000000 00000000 0f0a0000 12000000 ................
    0x00000408 00000000 00000000 00000000 00000000 ................
    0x00000418 1e0a0000 11000000 00000000 00000000 ................
    0x00000428 00000000 00000000 230a0000 12000000 ........#.......
    0x00000438 00000000 00000000 00000000 00000000 ................
    0x00000448 2c0a0000 12000000 00000000 00000000 ,...............
    0x00000458 00000000 00000000 320a0000 12000000 ........2.......
    0x00000468 00000000 00000000 00000000 00000000 ................
    0x00000478 3c0a0000 12000000 00000000 00000000 <...............
    0x00000488 00000000 00000000 560a0000 12000000 ........V.......
    0x00000498 00000000 00000000 00000000 00000000 ................
    0x000004a8 5e0a0000 12000000 00000000 00000000 ^...............
    0x000004b8 00000000 00000000 650a0000 12000000 ........e.......
    0x000004c8 00000000 00000000 00000000 00000000 ................
    0x000004d8 6e0a0000 12000000 00000000 00000000 n...............
    0x000004e8 00000000 00000000 740a0000 12000000 ........t.......
    0x000004f8 00000000 00000000 00000000 00000000 ................
    0x00000508 7b0a0000 12000000 00000000 00000000 {...............
    0x00000518 00000000 00000000 830a0000 12000000 ................
    0x00000528 00000000 00000000 00000000 00000000 ................
    0x00000538 8b0a0000 12000000 00000000 00000000 ................
    0x00000548 00000000 00000000 990a0000 12000000 ................
    0x00000558 00000000 00000000 00000000 00000000 ................
    0x00000568 a60a0000 12000000 00000000 00000000 ................
    0x00000578 00000000 00000000 ad0a0000 12000000 ................
    0x00000588 00000000 00000000 00000000 00000000 ................
    0x00000598 bd0a0000 12000000 00000000 00000000 ................
    0x000005a8 00000000 00000000 c40a0000 12000000 ................
    0x000005b8 00000000 00000000 00000000 00000000 ................
    0x000005c8 d70a0000 12000000 00000000 00000000 ................
    0x000005d8 00000000 00000000 ec0a0000 12000000 ................
    0x000005e8 00000000 00000000 00000000 00000000 ................
    0x000005f8 f30a0000 12000000 00000000 00000000 ................
    0x00000608 00000000 00000000 fa0a0000 12000000 ................
    0x00000618 00000000 00000000 00000000 00000000 ................
    0x00000628 0e0b0000 12000000 00000000 00000000 ................
    0x00000638 00000000 00000000 1b0b0000 12000000 ................
    0x00000648 00000000 00000000 00000000 00000000 ................
    0x00000658 2f0b0000 12000000 00000000 00000000 /...............
    0x00000668 00000000 00000000 420b0000 12000000 ........B.......
    0x00000678 00000000 00000000 00000000 00000000 ................
    0x00000688 550b0000 12000000 00000000 00000000 U...............
    0x00000698 00000000 00000000 5f0b0000 12000000 ........_.......
    0x000006a8 00000000 00000000 00000000 00000000 ................
    0x000006b8 750b0000 12000000 00000000 00000000 u...............
    0x000006c8 00000000 00000000 7d0b0000 12000000 ........}.......
    0x000006d8 00000000 00000000 00000000 00000000 ................
    0x000006e8 850b0000 12000000 00000000 00000000 ................
    0x000006f8 00000000 00000000 8c0b0000 12000000 ................
    0x00000708 00000000 00000000 00000000 00000000 ................
    0x00000718 a20b0000 12000000 00000000 00000000 ................
    0x00000728 00000000 00000000 b80b0000 12000000 ................
    0x00000738 00000000 00000000 00000000 00000000 ................
    0x00000748 c80b0000 12000000 00000000 00000000 ................
    0x00000758 00000000 00000000 de0b0000 12000000 ................
    0x00000768 00000000 00000000 00000000 00000000 ................
    0x00000778 6f020000 12000f00 e84c0400 00000000 o........L......
    0x00000788 d0190000 00000000 37050000 12000f00 ........7.......
    0x00000798 048a0700 00000000 880a0000 00000000 ................
    0x000007a8 8e090000 12000f00 588a0a00 00000000 ........X.......
    0x000007b8 340f0000 00000000 cc040000 12000f00 4...............
    0x000007c8 28610700 00000000 70080000 00000000 (a......p.......
    0x000007d8 29070000 12000f00 e8f80800 00000000 )...............
    0x000007e8 24220000 00000000 22030000 12000f00 $"......".......
    0x000007f8 ec230500 00000000 f02a0000 00000000 .#.......*......
    0x00000808 46030000 12000f00 245f0500 00000000 F.......$_......
    0x00000818 28680000 00000000 8c030000 12000f00 (h..............
    0x00000828 48ca0500 00000000 54180000 00000000 H.......T.......
    0x00000838 a9040000 12000f00 582b0700 00000000 ........X+......
    0x00000848 c0330000 00000000 67080000 12000f00 .3......g.......
    0x00000858 04f20900 00000000 98180000 00000000 ................
    0x00000868 f4000000 12000f00 90240300 00000000 .........$......
    0x00000878 f4110000 00000000 1a040000 12000f00 ................
    0x00000888 dc420600 00000000 70540000 00000000 .B......pT......
    0x00000898 c5050000 12000f00 38eb0700 00000000 ........8.......
    0x000008a8 04000000 00000000 4c070000 12000f00 ........L.......
    0x000008b8 0c1c0900 00000000 1c0e0000 00000000 ................
    0x000008c8 d8070000 12000f00 009d0900 00000000 ................
    0x000008d8 a40e0000 00000000 b7020000 12000f00 ................
    0x000008e8 20a60400 00000000 f0600000 00000000  ........`......
    0x000008f8 5a050000 12000f00 649c0700 00000000 Z.......d.......
    0x00000908 103b0000 00000000 9a060000 12000f00 .;..............
    0x00000918 589d0800 00000000 14200000 00000000 X........ ......
    0x00000928 20080000 12000f00 84bf0900 00000000  ...............
    0x00000938 04000000 00000000 e9050000 12000f00 ................
    0x00000948 60ee0700 00000000 ac1d0000 00000000 `...............
    0x00000958 2f060000 12000f00 88330800 00000000 /........3......
    0x00000968 d43f0000 00000000 76060000 12000f00 .?......v.......
    0x00000978 a8900800 00000000 7c0a0000 00000000 ........|.......
    0x00000988 05070000 12000f00 50e60800 00000000 ........P.......
    0x00000998 bc0e0000 00000000 60010000 12000f00 ........`.......
    0x000009a8 fc810300 00000000 700e0000 00000000 ........p.......
    0x000009b8 62040000 12000f00 20f20600 00000000 b....... .......
    0x000009c8 340f0000 00000000 5d090000 12000f00 4.......].......
    0x000009d8 84880a00 00000000 04000000 00000000 ................
    0x000009e8 d5090000 12000f00 38a40a00 00000000 ........8.......
    0x000009f8 9c050000 00000000 42000000 12000f00 ........B.......
    0x00000a08 986d0200 00000000 e0480000 00000000 .m.......H......
    0x00000a18 18010000 12000f00 2c3a0300 00000000 ........,:......
    0x00000a28 c42e0000 00000000 11020000 12000f00 ................
    0x00000a38 dcc80300 00000000 881f0000 00000000 ................
    0x00000a48 13050000 12000f00 f8830700 00000000 ................
    0x00000a58 58040000 00000000 4b020000 12000f00 X.......K.......
```

```
0x00000a68 e0020400 00000000 50460000 00000000 ........PF......
0x00000a78 fe020000 12000f00 580d0500 00000000 ........X.......
0x00000a88 e40d0000 00000000 8a080000 12000f00 ................
0x00000a98 340f0a00 00000000 dc310000 00000000 4........1......
0x00000aa8 ae000000 12000f00 98fa0200 00000000 ................
0x00000ab8 b4060000 00000000 db020000 12000f00 ................
0x00000ac8 50080500 00000000 1c040000 00000000 P...............
0x00000ad8 af030000 12000f00 ace30500 00000000 ................
0x00000ae8 e8050000 00000000 85040000 12000f00 ................
0x00000af8 8c050700 00000000 fc210000 00000000 .........!......
0x00000b08 ef040000 12000f00 a06a0700 00000000 .........j......
0x00000b18 d4150000 00000000 7d050000 12000f00 ........}.......
0x00000b28 acd90700 00000000 a8100000 00000000 ................
0x00000b38 6e070000 12000f00 c02d0900 00000000 n........-......
0x00000b48 04180000 00000000 8a000000 12000f00 ................
0x00000b58 bce20200 00000000 04160000 00000000 ................
0x00000b68 a1050000 12000f00 c4ea0700 00000000 ................
0x00000b78 04000000 00000000 53060000 12000f00 ........S.......
0x00000b88 14750800 00000000 901a0000 00000000 .u..............
0x00000b98 ae080000 12000f00 28420a00 00000000 ........(B......
0x00000ba8 60070000 00000000 3a090000 12000f00 `.......:.......
0x00000bb8 746a0a00 00000000 d41b0000 00000000 tj..............
0x00000bc8 93020000 12000f00 a46b0400 00000000 .........k......
0x00000bd8 cc2f0000 00000000 0c060000 12000f00 ./..............
0x00000be8 b80d0800 00000000 241c0000 00000000 ........$.......
0x00000bf8 be060000 12000f00 c4bf0800 00000000 ................
0x00000c08 b0120000 00000000 92070000 12000f00 ................
0x00000c18 ac4a0900 00000000 94270000 00000000 .J.......'......
0x00000c28 ee010000 12000f00 c0b90300 00000000 ................
0x00000c38 100b0000 00000000 3e040000 12000f00 ........>.......
0x00000c48 1ca30600 00000000 744c0000 00000000 ........tL......
0x00000c58 e2060000 12000f00 fcd40800 00000000 ................
0x00000c68 0c0f0000 00000000 fc070000 12000f00 ................
0x00000c78 44af0900 00000000 ec0f0000 00000000 D...............
0x00000c88 b2090000 12000f00 64a10a00 00000000 ........d.......
0x00000c98 04000000 00000000 d1000000 12000f00 ................
0x00000ca8 b40d0300 00000000 0c150000 00000000 ................
0x00000cb8 f7030000 12000f00 04090600 00000000 ................
0x00000cc8 cc280000 00000000 b6070000 12000f00 .(..............
0x00000cd8 f8790900 00000000 20220000 00000000 .y...... "......
0x00000ce8 43080000 12000f00 ecc30900 00000000 C...............
0x00000cf8 2c2c0000 00000000 d1080000 12000f00 ,,..............
0x00000d08 e84b0a00 00000000 1c120000 00000000 .K..............
0x00000d18 17090000 12000f00 845f0a00 00000000 ........._......
0x00000d28 30090000 00000000 e0090000 12000f00 0...............
0x00000d38 d4a90a00 00000000 f0150000 00000000 ................
0x00000d48 3c010000 12000f00 b86a0300 00000000 <........j......
0x00000d58 f80e0000 00000000 a7010000 12000f00 ................
0x00000d68 acb20300 00000000 04000000 00000000 ................
0x00000d78 cb010000 12000f00 fcb80300 00000000 ................
0x00000d88 04000000 00000000 66000000 12000f00 ........f.......
0x00000d98 30b90200 00000000 b8270000 00000000 0........'......
0x00000da8 83010000 12000f00 64940300 00000000 ........d.......
0x00000db8 d81d0000 00000000 69030000 12000f00 ........i.......
0x00000dc8 90c70500 00000000 04000000 00000000 ................
0x00000dd8 d3030000 12000f00 50ed0500 00000000 ........P.......
```

```
  0x00000de8 a0180000 00000000 f4080000 12000f00 ................
  0x00000df8 745e0a00 00000000 04000000 00000000 t^..............
```

举例2

对于：

```
  [ 8] .dynstr           STRTAB           00000000000014dc  000014dc
       0000000000000c19  0000000000000000   A       0     0     1
```

去打印信息：

```
➜  arm64-v8a readelf -x 8 libtacker.so

Hex dump of section '.dynstr':
  0x000014dc 005f5f63 78615f66 696e616c 697a6500 .__cxa_finalize.
  0x000014ec 5f5f6378 615f6174 65786974 005f5f61 __cxa_atexit.__a
  0x000014fc 6e64726f 69645f6c 6f675f70 72696e74 ndroid_log_print
  0x0000150c 005f5f73 7461636b 5f63686b 5f666169 .__stack_chk_fai
  0x0000151c 6c002e64 61746164 69765f64 65636f64 l..datadiv_decod
  0x0000152c 65313233 33353032 37323838 39353431 e12335027288954l
  0x0000153c 32343732 33002e64 61746164 69765f64 24723..datadiv_d
  0x0000154c 65636f64 65313738 33383633 36333233 ecode17838636323
  0x0000155c 31393833 31303134 32002e64 61746164 198310142..datad
  0x0000156c 69765f64 65636f64 65313833 32383431 iv_decode1832841
  0x0000157c 37353239 34353435 34373030 34002e64 7529454547004..d
  0x0000158c 61746164 69765f64 65636f64 65383935 atadiv_decode895
  0x0000159c 32323436 38353132 36353037 30333639 2246851265070369
  0x000015ac 002e6461 74616469 765f6465 636f6465 ..datadiv_decode
  0x000015bc 38303130 32383830 33383333 39383933 8010288038339893
  0x000015cc 36303700 2e646174 61646976 5f646563 607..datadiv_dec
...
  0x00001d8c 61746164 69765f64 65636f64 65363430 atadiv_decode640
  0x00001d9c 35373231 36383033 35343634 39323630 5721680354649260
  0x00001dac 002e6461 74616469 765f6465 636f6465 ..datadiv_decode
  0x00001dbc 31363339 32363237 32383730 36373831 1639262728706781
  0x00001dcc 33303800 2e646174 61646976 5f646563 308..datadiv_dec
  0x00001ddc 6f646535 34353434 30363535 32303137 ode5454406552017
  0x00001dec 35353732 3936002e 64617461 6469765f 557296..datadiv_
  0x00001dfc 6465636f 64653535 33333233 36323439 decode5533236249
  0x00001e0c 31393233 32383335 35002e64 61746164 192328355..datad
  0x00001e1c 69765f64 65636f64 65383331 36333831 iv_decode8316381
  0x00001e2c 34383032 38383136 37353335 002e6461 480288167535..da
  0x00001e3c 74616469 765f6465 636f6465 31313730 tadiv_decode1170
  0x00001e4c 36313031 34313432 39353232 35393132 6101414295225912
  0x00001e5c 00707468 72656164 5f73656c 66002e64 .pthread_self..d
  0x00001e6c 61746164 69765f64 65636f64 65313437 atadiv_decode147
  0x00001e7c 31363230 32313831 34383632 32333832 1620218148622382
  0x00001e8c 32002e64 61746164 69765f64 65636f64 2..datadiv_decod
  0x00001e9c 65383735 38383430 37353530 32343830 e875884075502480
  0x00001eac 31313630 004a4e49 5f4f6e4c 6f616400 1160.JNI_OnLoad.
  0x00001ebc 2e646174 61646976 5f646563 6f646532 .datadiv_decode2
  0x00001ecc 34343434 39373231 32363930 38313033 4444972126908103
```

```
    0x00001edc 3630006d 616c6c6f 63006672 65650070 60.malloc.free.p
    0x00001eec 6f736978 5f6d656d 616c6967 6e005f5f osix_memalign.__
    0x00001efc 73460076 66707269 6e746600 66707574 sF.vfprintf.fput
    0x00001f0c 63007661 73707269 6e746600 616e6472 c.vasprintf.andr
    0x00001f1c 6f69645f 7365745f 61626f72 745f6d65 oid_set_abort_me
    0x00001f2c 73736167 65006f70 656e6c6f 67007379 ssage.openlog.sy
    0x00001f3c 736c6f67 00636c6f 73656c6f 67006162 slog.closelog.ab
    0x00001f4c 6f727400 7374726c 656e0072 65616c6c ort.strlen.reall
    0x00001f5c 6f63006d 656d6d6f 7665005f 5f6d656d oc.memmove.__mem
    0x00001f6c 6d6f7665 5f63686b 005f5f73 74726c65 move_chk.__strle
    0x00001f7c 6e5f6368 6b006d65 6d636872 005f5f76 n_chk.memchr.__v
    0x00001f8c 736e7072 696e7466 5f63686b 006d656d snprintf_chk.mem
    0x00001f9c 63707900 70746872 6561645f 6d757465 cpy.pthread_mute
    0x00001fac 785f6c6f 636b0070 74687265 61645f6d x_lock.pthread_m
    0x00001fbc 75746578 5f756e6c 6f636b00 63616c6c utex_unlock.call
    0x00001fcc 6f630073 7472636d 70007074 68726561 oc.strcmp.pthrea
    0x00001fdc 645f6765 74737065 63696669 63007074 d_getspecific.pt
    0x00001fec 68726561 645f6f6e 63650070 74687265 hread_once.pthre
    0x00001ffc 61645f73 65747370 65636966 69630070 ad_setspecific.p
    0x0000200c 74687265 61645f6b 65795f64 656c6574 thread_key_delet
    0x0000201c 65007074 68726561 645f6b65 795f6372 e.pthread_key_cr
    0x0000202c 65617465 00676574 61757876 616c005f eate.getauxval._
    0x0000203c 5f737973 74656d5f 70726f70 65727479 _system_property
    0x0000204c 5f676574 00737472 6e636d70 00667072 _get.strncmp.fpr
    0x0000205c 696e7466 0066666c 75736800 70746872 intf.fflush.pthr
    0x0000206c 6561645f 72776c6f 636b5f77 726c6f63 ead_rwlock_wrloc
    0x0000207c 6b007074 68726561 645f7277 6c6f636b k.pthread_rwlock
    0x0000208c 5f756e6c 6f636b00 646c5f69 74657261 _unlock.dl_itera
    0x0000209c 74655f70 68647200 70746872 6561645f te_phdr.pthread_
    0x000020ac 72776c6f 636b5f72 646c6f63 6b006677 rwlock_rdlock.fw
    0x000020bc 72697465 006c6962 646c2e73 6f004c49 rite.libdl.so.LI
    0x000020cc 4243006c 6962632e 736f006c 69626c6f BC.libc.so.liblo
    0x000020dc 672e736f 006c6962 6d2e736f 006c6962 g.so.libm.so.lib
    0x000020ec 666f7263 652e736f 00                force.so.
```

-p：以string字符串方式显示

```
➜  arm64-v8a readelf -p .dynsym libtacker.so


String dump of section '.dynsym':
  [    60]  1
  [    78]  4^B
  [    90]  ;^B
  [    a8]  C^B
  [   138]  #\n
  ...
  [   9e1]  y^I
  [   9e8]   "
  [   9f0]  C^H
  [   a00]  ,,
  [   a11]  K\n
  [   a29]  _\n
  [   a30]  0^I
  [   a50]  <^A
  [   a59]  j^C
  [   a98]  f
  [   aa0]  0�^B
  [   aa9]  '
  [   ab8]  d�^C
  [   ac8]  i^C
  [   ae8]  P�^E
  [   b00]  t^\n
```

和：

```
➜  arm64-v8a readelf -p 8 libtacker.so


String dump of section '.dynstr':
  [     1]  __cxa_finalize
  [    10]  __cxa_atexit
  [    1d]  __android_log_print
  [    31]  __stack_chk_fail
  [    42]  .datadiv_decode12335027288954124723
  [    66]  .datadiv_decode17838636323198310142
  [    8a]  .datadiv_decode18328417529454547004
...
  [   1ee]  .datadiv_decode17717905520691252O6
  [   211]  .datadiv_decode5616837089396308971
  [   234]  memset
  [   23b]  strncpy
  [   243]  strncat
  [   24b]  .datadiv_decode14151120317447827231
  [   26f]  .datadiv_decode16117807209816376729
...
  [   95d]  .datadiv_decode11706101414295225912
  [   981]  pthread_self
  [   98e]  .datadiv_decode14716202181486223822
  [   9b2]  .datadiv_decode8758840755024801160
  [   9d5]  JNI_OnLoad
  [   9e0]  .datadiv_decode2444497212690810360
  [   a03]  malloc
  [   a0a]  free
  [   a0f]  posix_memalign
  [   a1e]  __sF
  [   a23]  vfprintf
  [   a2c]  fputc
  [   a32]  vasprintf
  [   a3c]  android_set_abort_message
  [   a56]  openlog
  [   a5e]  syslog
  [   a65]  closelog
  [   a6e]  abort
  [   a74]  strlen
  [   a7b]  realloc
  [   a83]  memmove
  [   a8b]  __memmove_chk
  [   a99]  __strlen_chk
  [   aa6]  memchr
  [   aad]  __vsnprintf_chk
  [   abd]  memcpy
  [   ac4]  pthread_mutex_lock
  [   ad7]  pthread_mutex_unlock
  [   aec]  calloc
  [   af3]  strcmp
  [   afa]  pthread_getspecific
  [   b0e]  pthread_once
  [   b1b]  pthread_setspecific
  [   b2f]  pthread_key_delete
  [   b42]  pthread_key_create
  [   b55]  getauxval
  [   b5f]  __system_property_get
  [   b75]  strncmp
  [   b7d]  fprintf
  [   b85]  fflush
  [   b8c]  pthread_rwlock_wrlock
  [   ba2]  pthread_rwlock_unlock
  [   bb8]  dl_iterate_phdr
  [   bc8]  pthread_rwlock_rdlock
  [   bde]  fwrite
  [   be5]  libdl.so
  [   bee]  LIBC
  [   bf3]  libc.so
  [   bfb]  liblog.so
  [   c05]  libm.so
  [   c0d]  libforce.so
```

可以输出对应的函数和库名

-e：显示多个头信息

```
➜  arm64-v8a readelf -e libtacker.so
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           AArch64
  Version:                           0x1
  Entry point address:               0x1a5c0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          848344 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         27
  Section header string table index: 26
Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .note.androi[...] NOTE             0000000000000238  00000238
       0000000000000098  0000000000000000   A       0     0     4
  [ 2] .note.gnu.bu[...] NOTE             00000000000002d0  000002d0
       0000000000000024  0000000000000000   A       0     0     4
  [ 3] .dynsym           DYNSYM           00000000000002f8  000002f8
       0000000000000b10  0000000000000018   A       8     1     8
  [ 4] .gnu.version      VERSYM           0000000000000e08  00000e08
       00000000000000ec  0000000000000002   A       3     0     2
  [ 5] .gnu.version_r    VERNEED          0000000000000ef4  00000ef4
       0000000000000040  0000000000000000   A       8     2     4
  [ 6] .gnu.hash         GNU_HASH         0000000000000f38  00000f38
       00000000000001ec  0000000000000000   A       3     0     8
  [ 7] .hash             HASH             0000000000001124  00001124
       00000000000003b8  0000000000000004   A       3     0     4
  [ 8] .dynstr           STRTAB           00000000000014dc  000014dc
       0000000000000c19  0000000000000000   A       0     0     1
  [ 9] .rela.dyn         RELA             00000000000020f8  000020f8
       0000000000008850  0000000000000018   A       3     0     8
  [10] .rela.plt         RELA             000000000000a948  0000a948
       0000000000000450  0000000000000018  AI       3    22     8
  [11] .gcc_except_table PROGBITS         000000000000ad98  0000ad98
       0000000000001960  0000000000000000   A       0     0     4
  [12] .rodata           PROGBITS         000000000000c6f8  0000c6f8
       0000000000003434  0000000000000000 AMS       0     0     8
  [13] .eh_frame_hdr     PROGBITS         000000000000fb2c  0000fb2c
       0000000000001dbc  0000000000000000   A       0     0     4
  [14] .eh_frame         PROGBITS         00000000000118e8  000118e8
       0000000000008cd4  0000000000000000   A       0     0     8
  [15] .text             PROGBITS         000000000001a5c0  0001a5c0
       00000000000aec60  0000000000000000  AX       0     0     16
  [16] .plt              PROGBITS         00000000000c9220  000c9220
       0000000000000300  0000000000000000  AX       0     0     16
  [17] .data.rel.ro      PROGBITS         00000000000ca520  000c9520
       0000000000002eb8  0000000000000000  WA       0     0     8
  [18] .fini_array       FINI_ARRAY       00000000000cd3d8  000cc3d8
       0000000000000010  0000000000000000  WA       0     0     8
  [19] .init_array       INIT_ARRAY       00000000000cd3e8  000cc3e8
       0000000000000230  0000000000000000  WA       0     0     8
  [20] .dynamic          DYNAMIC          00000000000cd618  000cc618
       00000000000001d0  0000000000000010  WA       8     0     8
  [21] .got              PROGBITS         00000000000cd7e8  000cc7e8
       00000000000000c0  0000000000000000  WA       0     0     8
  [22] .got.plt          PROGBITS         00000000000cd8a8  000cc8a8
       0000000000000188  0000000000000000  WA       0     0     8
  [23] .data             PROGBITS         00000000000cea30  000cca30
       00000000000025d8  0000000000000000  WA       0     0     16
  [24] .bss              NOBITS           00000000000d1010  000cf008
       0000000000000ad0  0000000000000000  WA       0     0     16
  [25] .comment          PROGBITS         0000000000000000  000cf008
```

```
       00000000000000c6  0000000000000001  MS       0     0     1
  [26] .shstrtab         STRTAB           0000000000000000  000cf0ce
       0000000000000104  0000000000000000           0     0     1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings), I (info),
  L (link order), O (extra OS processing required), G (group), T (TLS),
  C (compressed), x (unknown), o (OS specific), E (exclude),
  D (mbind), p (processor specific)


Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000000040 0x0000000000000040
                 0x00000000000001f8 0x00000000000001f8  R      0x8
  LOAD           0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x00000000000c9520 0x00000000000c9520  R E    0x1000
  LOAD           0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003510  RW     0x1000
  LOAD           0x00000000000cca30 0x00000000000cea30 0x00000000000cea30
                 0x00000000000025d8 0x00000000000030b0  RW     0x1000
  DYNAMIC        0x00000000000cc618 0x00000000000cd618 0x00000000000cd618
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  GNU_RELRO      0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003ae0  R      0x1
  GNU_EH_FRAME   0x000000000000fb2c 0x000000000000fb2c 0x000000000000fb2c
                 0x0000000000001dbc 0x0000000000001dbc  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x0
  NOTE           0x0000000000000238 0x0000000000000238 0x0000000000000238
                 0x00000000000000bc 0x00000000000000bc  R      0x4


 Section to Segment mapping:
  Segment Sections...
   00
   01     .note.android.ident .note.gnu.build-id .dynsym .gnu.version .gnu.version_r .g
nu.hash .hash .dynstr .rela.dyn .rela.plt .gcc_except_table .rodata .eh_frame_hdr .eh_f
rame .text .plt
   02     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   03     .data .bss
   04     .dynamic
   05     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   06     .eh_frame_hdr
   07
   08     .note.android.ident .note.gnu.build-id
```

-a：显示所有信息

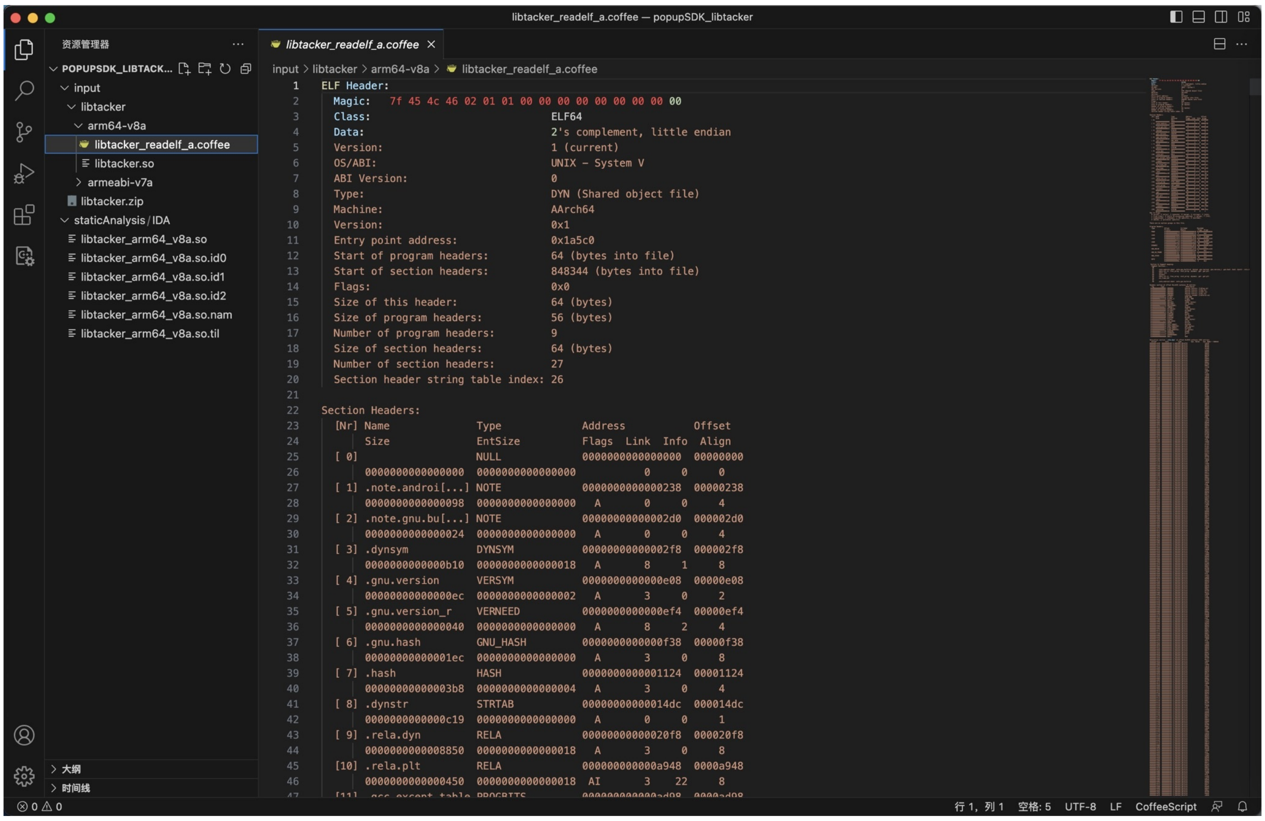
```
readelf -a libtacker.so
```

输出内容太多

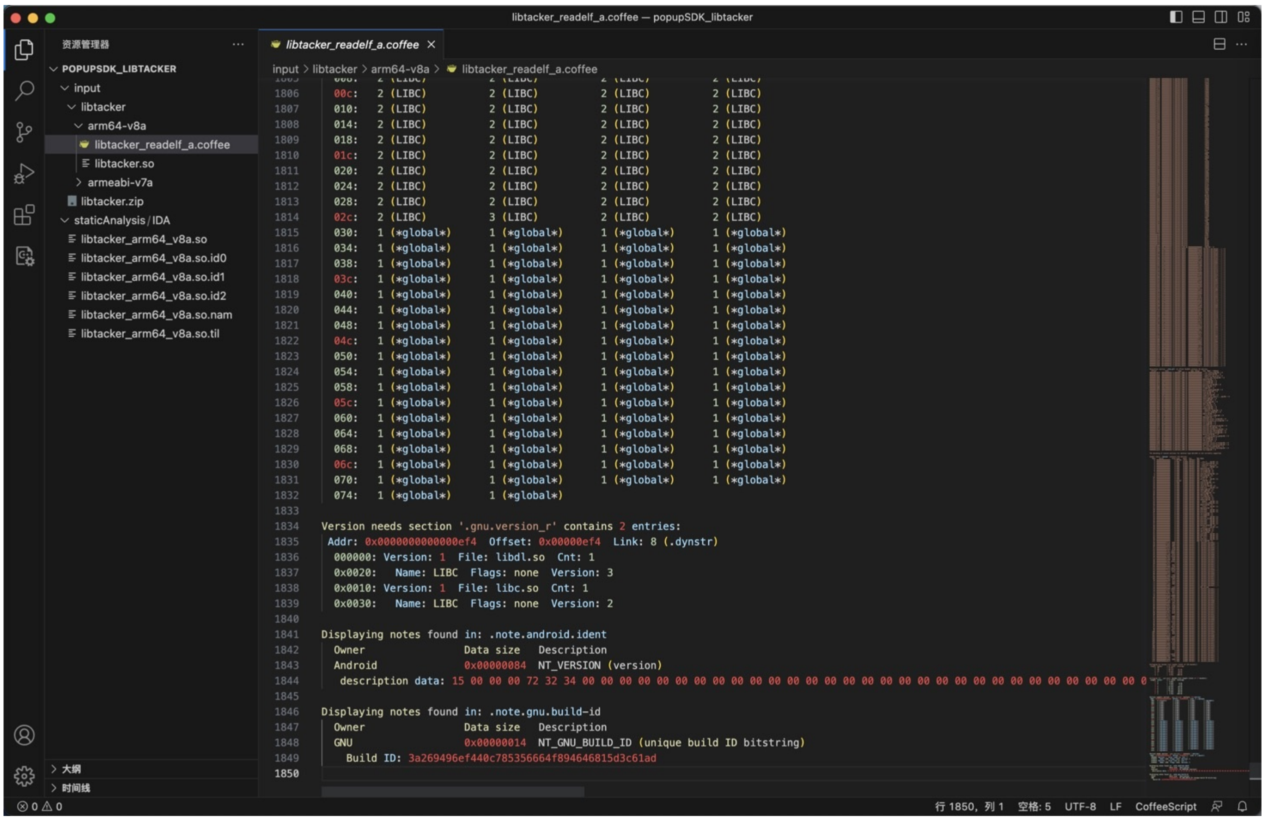
```
➜ arm64-v8a readelf -a libtacker.so > libtacker_readelf_a.coffee
```

保存到文件：`libtacker_readelf_a.coffee`

效果如下：


libtacker_readelf_a.coffee — popupSDK_libtacker
资源管理器
POPUPSDK_LIBTACK...
input
libtacker
arm64-v8a
libtacker_readelf_a.coffee
libtacker.so
armeabi-v7a
libtacker.zip
staticAnalysis / IDA
libtacker_arm64_v8a.so
libtacker_arm64_v8a.so.id0
libtacker_arm64_v8a.so.id1
libtacker_arm64_v8a.so.id2
libtacker_arm64_v8a.so.nam
libtacker_arm64_v8a.so.til
大纲
时间线
input > libtacker > arm64-v8a > libtacker_readelf_a.coffee
行 1, 列 1   空格: 5   UTF-8   LF   CoffeeScript

```
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              DYN (Shared object file)
  Machine:                           AArch64
  Version:                           0x1
  Entry point address:               0x1a5c0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          848344 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         27
  Section header string table index: 26

Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .note.androi[...] NOTE             0000000000000238  00000238
       0000000000000098  0000000000000000   A       0     0     4
  [ 2] .note.gnu.bu[...] NOTE             00000000000002d0  000002d0
       0000000000000024  0000000000000000   A       0     0     4
  [ 3] .dynsym           DYNSYM           00000000000002f8  000002f8
       0000000000000b10  0000000000000018   A       8     1     8
  [ 4] .gnu.version      VERSYM           0000000000000e08  00000e08
       00000000000000ec  0000000000000002   A       3     0     2
  [ 5] .gnu.version_r    VERNEED          0000000000000ef4  00000ef4
       0000000000000040  0000000000000000   A       8     2     4
  [ 6] .gnu.hash         GNU_HASH         0000000000000f38  00000f38
       00000000000001ec  0000000000000000   A       3     0     8
  [ 7] .hash             HASH             0000000000001124  00001124
       00000000000003b8  0000000000000004   A       3     0     4
  [ 8] .dynstr           STRTAB           00000000000014dc  000014dc
       0000000000000c19  0000000000000000   A       0     0     1
  [ 9] .rela.dyn         RELA             00000000000020f8  000020f8
       0000000000008850  0000000000000018   A       3     0     8
  [10] .rela.plt         RELA             000000000000a948  0000a948
       0000000000000450  0000000000000018  AI       3    22     8
```


libtacker_readelf_a.coffee — popupSDK_libtacker
资源管理器
POPUPSDK_LIBTACKER
input
libtacker
arm64-v8a
libtacker_readelf_a.coffee
libtacker.so
armeabi-v7a
libtacker.zip
staticAnalysis / IDA
libtacker_arm64_v8a.so
libtacker_arm64_v8a.so.id0
libtacker_arm64_v8a.so.id1
libtacker_arm64_v8a.so.id2
libtacker_arm64_v8a.so.nam
libtacker_arm64_v8a.so.til
大纲
时间线
input > libtacker > arm64-v8a > libtacker_readelf_a.coffee
行 1850, 列 1   空格: 5   UTF-8   LF   CoffeeScript

```
  00c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  010:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  014:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  018:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  01c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  020:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  024:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  028:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  02c:   2 (LIBC)          3 (LIBC)          2 (LIBC)          2 (LIBC)
  030:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  034:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  038:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  03c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  040:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  044:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  048:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  04c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  050:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  054:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  058:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  05c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  060:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  064:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  068:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  06c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  070:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  074:   1 (*global*)      1 (*global*)

Version needs section '.gnu.version_r' contains 2 entries:
 Addr: 0x0000000000000ef4  Offset: 0x00000ef4  Link: 8 (.dynstr)
  000000: Version: 1  File: libdl.so  Cnt: 1
  0x0020:   Name: LIBC  Flags: none  Version: 3
  0x0010: Version: 1  File: libc.so  Cnt: 1
  0x0030:   Name: LIBC  Flags: none  Version: 2

Displaying notes found in: .note.android.ident
  Owner                Data size  Description
  Android              0x00000084 NT_VERSION (version)
   description data: 15 00 00 00 72 32 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0

Displaying notes found in: .note.gnu.build-id
  Owner                Data size  Description
  GNU                  0x00000014 NT_GNU_BUILD_ID (unique build ID bitstring)
    Build ID: 3a269496ef440c785356664f894646815d3c61ad
```

拷贝出部分内容：

Section Headers

```
Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .note.androi[...] NOTE             0000000000000238  00000238
       0000000000000098  0000000000000000   A       0     0     4
  [ 2] .note.gnu.bu[...] NOTE             00000000000002d0  000002d0
       0000000000000024  0000000000000000   A       0     0     4
  [ 3] .dynsym           DYNSYM           00000000000002f8  000002f8
       0000000000000b10  0000000000000018   A       8     1     8
  [ 4] .gnu.version      VERSYM           0000000000000e08  00000e08
       00000000000000ec  0000000000000002   A       3     0     2
  [ 5] .gnu.version_r    VERNEED          0000000000000ef4  00000ef4
       0000000000000040  0000000000000000   A       8     2     4
  [ 6] .gnu.hash         GNU_HASH         0000000000000f38  00000f38
       00000000000001ec  0000000000000000   A       3     0     8
  [ 7] .hash             HASH             0000000000001124  00001124
       00000000000003b8  0000000000000004   A       3     0     4
  [ 8] .dynstr           STRTAB           00000000000014dc  000014dc
       0000000000000c19  0000000000000000   A       0     0     1
  [ 9] .rela.dyn         RELA             00000000000020f8  000020f8
       0000000000008850  0000000000000018   A       3     0     8
  [10] .rela.plt         RELA             000000000000a948  0000a948
       0000000000000450  0000000000000018  AI       3    22     8
  [11] .gcc_except_table PROGBITS         000000000000ad98  0000ad98
       0000000000001960  0000000000000000   A       0     0     4
  [12] .rodata           PROGBITS         000000000000c6f8  0000c6f8
       0000000000003434  0000000000000000 AMS       0     0     8
  [13] .eh_frame_hdr     PROGBITS         000000000000fb2c  0000fb2c
       0000000000001dbc  0000000000000000   A       0     0     4
  [14] .eh_frame         PROGBITS         00000000000118e8  000118e8
       0000000000008cd4  0000000000000000   A       0     0     8
  [15] .text             PROGBITS         000000000001a5c0  0001a5c0
       00000000000aec60  0000000000000000  AX       0     0     16
  [16] .plt              PROGBITS         00000000000c9220  000c9220
       0000000000000300  0000000000000000  AX       0     0     16
  [17] .data.rel.ro      PROGBITS         00000000000ca520  000c9520
       0000000000002eb8  0000000000000000  WA       0     0     8
  [18] .fini_array       FINI_ARRAY       00000000000cd3d8  000cc3d8
       0000000000000010  0000000000000000  WA       0     0     8
  [19] .init_array       INIT_ARRAY       00000000000cd3e8  000cc3e8
       0000000000000230  0000000000000000  WA       0     0     8
  [20] .dynamic          DYNAMIC          00000000000cd618  000cc618
       00000000000001d0  0000000000000010  WA       8     0     8
  [21] .got              PROGBITS         00000000000cd7e8  000cc7e8
       00000000000000c0  0000000000000000  WA       0     0     8
  [22] .got.plt          PROGBITS         00000000000cd8a8  000cc8a8
       0000000000000188  0000000000000000  WA       0     0     8
  [23] .data             PROGBITS         00000000000cea30  000cca30
       00000000000025d8  0000000000000000  WA       0     0     16
  [24] .bss              NOBITS           00000000000d1010  000cf008
       0000000000000ad0  0000000000000000  WA       0     0     16
  [25] .comment          PROGBITS         0000000000000000  000cf008
       00000000000000c6  0000000000000001  MS       0     0     1
  [26] .shstrtab         STRTAB           0000000000000000  000cf0ce
       0000000000000104  0000000000000000           0     0     1

Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings), I (info),
  L (link order), O (extra OS processing required), G (group), T (TLS),
  C (compressed), x (unknown), o (OS specific), E (exclude),
  D (mbind), p (processor specific)
```

有很多常见的：section

- bss
- data
- rodata
- got
- 等等

Program Headers

```
Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000000040 0x0000000000000040
                 0x00000000000001f8 0x00000000000001f8  R      0x8
  LOAD           0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x00000000000c9520 0x00000000000c9520  R E    0x1000
  LOAD           0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003510  RW     0x1000
  LOAD           0x00000000000cca30 0x00000000000cea30 0x00000000000cea30
                 0x00000000000025d8 0x00000000000030b0  RW     0x1000
  DYNAMIC        0x00000000000cc618 0x00000000000cd618 0x00000000000cd618
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  GNU_RELRO      0x00000000000c9520 0x00000000000ca520 0x00000000000ca520
                 0x0000000000003510 0x0000000000003ae0  R      0x1
  GNU_EH_FRAME   0x000000000000fb2c 0x000000000000fb2c 0x000000000000fb2c
                 0x0000000000001dbc 0x0000000000001dbc  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x0
  NOTE           0x0000000000000238 0x0000000000000238 0x0000000000000238
                 0x00000000000000bc 0x00000000000000bc  R      0x4
```

Section to Segment mapping

```
Section to Segment mapping:
  Segment Sections...
   00
   01     .note.android.ident .note.gnu.build-id .dynsym .gnu.version .gnu.version_r .gnu.hash .hash .dynstr .rela.dyn .rela.plt .gcc_except_table .rodata .eh_frame_hdr .eh_frame .text .plt
   02     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   03     .data .bss
   04     .dynamic
   05     .data.rel.ro .fini_array .init_array .dynamic .got .got.plt
   06     .eh_frame_hdr
   07
   08     .note.android.ident .note.gnu.build-id
```

Dynamic section

Dynamic section at offset 0xcc618 contains 29 entries

```
Dynamic section at offset 0xcc618 contains 29 entries:
  Tag        Type                         Name/Value
 0x0000000000000001 (NEEDED)             Shared library: [liblog.so]
 0x0000000000000001 (NEEDED)             Shared library: [libm.so]
 0x0000000000000001 (NEEDED)             Shared library: [libdl.so]
 0x0000000000000001 (NEEDED)             Shared library: [libc.so]
 0x000000000000000e (SONAME)             Library soname: [libforce.so]
 0x000000000000001e (FLAGS)              BIND_NOW
 0x000000006ffffffb (FLAGS_1)            Flags: NOW
 0x0000000000000007 (RELA)               0x20f8
 0x0000000000000008 (RELASZ)             34896 (bytes)
 0x0000000000000009 (RELAENT)            24 (bytes)
 0x000000006ffffff9 (RELACOUNT)          1384
 0x0000000000000017 (JMPREL)             0xa948
 0x0000000000000002 (PLTRELSZ)           1104 (bytes)
 0x0000000000000003 (PLTGOT)             0xcd8a8
 0x0000000000000014 (PLTREL)             RELA
 0x0000000000000006 (SYMTAB)             0x2f8
 0x000000000000000b (SYMENT)             24 (bytes)
 0x0000000000000005 (STRTAB)             0x14dc
 0x000000000000000a (STRSZ)              3097 (bytes)
 0x000000006ffffef5 (GNU_HASH)           0xf38
 0x0000000000000004 (HASH)               0x1124
 0x0000000000000019 (INIT_ARRAY)         0xcd3e8
 0x000000000000001b (INIT_ARRAYSZ)       560 (bytes)
 0x000000000000001a (FINI_ARRAY)         0xcd3d8
 0x000000000000001c (FINI_ARRAYSZ)       16 (bytes)
 0x000000006ffffff0 (VERSYM)             0xe08
 0x000000006ffffffe (VERNEED)            0xef4
 0x000000006fffffff (VERNEEDNUM)         2
 0x0000000000000000 (NULL)               0x0
```

Relocation section

Relocation section '.rela.dyn' at offset 0x20f8 contains 1454 entries

```
Relocation section '.rela.dyn' at offset 0x20f8 contains 1454 entries:
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
0000000ca520  000000000403 R_AARCH64_RELATIV                    ca520
0000000ca528  000000000403 R_AARCH64_RELATIV                    d0f58
```

```
0000000ca530  000000000403 R_AARCH64_RELATIV                    d0f90
0000000ca538  000000000403 R_AARCH64_RELATIV                    aa168
0000000ca548  000000000403 R_AARCH64_RELATIV                    ca5a8
0000000ca550  000000000403 R_AARCH64_RELATIV                    b05d8
...
0000000d0ff8  000000000403 R_AARCH64_RELATIV                    d12e0
0000000d1000  000000000403 R_AARCH64_RELATIV                    d1ae0
0000000cd828  000c00000401 R_AARCH64_GLOB_DA 0000000000000000 __sF@LIBC + 0
0000000cd460  003000000101 R_AARCH64_ABS64   0000000000044ce8 .datadiv_decode16[...] + 0

0000000cd500  003100000101 R_AARCH64_ABS64   0000000000078a04 .datadiv_decode99[...] + 0


...
0000000cd3f0  007100000101 R_AARCH64_ABS64   000000000002b930 .datadiv_decode17[...] + 0

0000000cd430  007200000101 R_AARCH64_ABS64   0000000000039464 .datadiv_decode17[...] + 0

0000000cd498  007300000101 R_AARCH64_ABS64   000000000005c790 .datadiv_decode15[...] + 0

0000000cd4b0  007400000101 R_AARCH64_ABS64   000000000005ed50 .datadiv_decode15[...] + 0

0000000cd5d8  007500000101 R_AARCH64_ABS64   00000000000a5e74 .datadiv_decode54[...] + 0
```

Relocation section '.rela.plt' at offset 0xa948 contains 46 entries

```
Relocation section '.rela.plt' at offset 0xa948 contains 46 entries:
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
0000000cd8c0  000100000402 R_AARCH64_JUMP_SL 0000000000000000 __cxa_finalize@LIBC + 0
0000000cd8c8  000200000402 R_AARCH64_JUMP_SL 0000000000000000 __cxa_atexit@LIBC + 0
0000000cd8d0  000300000402 R_AARCH64_JUMP_SL 0000000000000000 __android_log_print + 0
0000000cd8d8  000400000402 R_AARCH64_JUMP_SL 0000000000000000 __stack_chk_fail@LIBC + 0
0000000cd8e0  000500000402 R_AARCH64_JUMP_SL 0000000000000000 memset@LIBC + 0
0000000cd8e8  000600000402 R_AARCH64_JUMP_SL 0000000000000000 strncpy@LIBC + 0
0000000cd8f0  000700000402 R_AARCH64_JUMP_SL 0000000000000000 strncat@LIBC + 0
0000000cd8f8  000800000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_self@LIBC + 0
0000000cd900  000900000402 R_AARCH64_JUMP_SL 0000000000000000 malloc@LIBC + 0
0000000cd908  000a00000402 R_AARCH64_JUMP_SL 0000000000000000 free@LIBC + 0
0000000cd910  000b00000402 R_AARCH64_JUMP_SL 0000000000000000 posix_memalign@LIBC + 0
0000000cd918  000d00000402 R_AARCH64_JUMP_SL 0000000000000000 vfprintf@LIBC + 0
0000000cd920  000e00000402 R_AARCH64_JUMP_SL 0000000000000000 fputc@LIBC + 0
0000000cd928  000f00000402 R_AARCH64_JUMP_SL 0000000000000000 vasprintf@LIBC + 0
0000000cd930  001000000402 R_AARCH64_JUMP_SL 0000000000000000 android_set_abort[...]@LI
BC + 0
0000000cd938  001100000402 R_AARCH64_JUMP_SL 0000000000000000 openlog@LIBC + 0
0000000cd940  001200000402 R_AARCH64_JUMP_SL 0000000000000000 syslog@LIBC + 0
0000000cd948  001300000402 R_AARCH64_JUMP_SL 0000000000000000 closelog@LIBC + 0
0000000cd950  001400000402 R_AARCH64_JUMP_SL 0000000000000000 abort@LIBC + 0
0000000cd958  001500000402 R_AARCH64_JUMP_SL 0000000000000000 strlen@LIBC + 0
0000000cd960  001600000402 R_AARCH64_JUMP_SL 0000000000000000 realloc@LIBC + 0
0000000cd968  001700000402 R_AARCH64_JUMP_SL 0000000000000000 memmove@LIBC + 0
0000000cd970  001800000402 R_AARCH64_JUMP_SL 0000000000000000 __memmove_chk@LIBC + 0
0000000cd978  001900000402 R_AARCH64_JUMP_SL 0000000000000000 __strlen_chk@LIBC + 0
0000000cd980  001a00000402 R_AARCH64_JUMP_SL 0000000000000000 memchr@LIBC + 0
```

```
0000000cd988  001b00000402 R_AARCH64_JUMP_SL 0000000000000000 __vsnprintf_chk@LIBC + 0
0000000cd990  001c00000402 R_AARCH64_JUMP_SL 0000000000000000 memcpy@LIBC + 0
0000000cd998  001d00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_mutex_lock@LIBC +
0
0000000cd9a0  001e00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_mutex_unlock@LIBC
 + 0
0000000cd9a8  001f00000402 R_AARCH64_JUMP_SL 0000000000000000 calloc@LIBC + 0
0000000cd9b0  002000000402 R_AARCH64_JUMP_SL 0000000000000000 strcmp@LIBC + 0
0000000cd9b8  002100000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_getspecific@LIBC
+ 0
0000000cd9c0  002200000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_once@LIBC + 0
0000000cd9c8  002300000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_setspecific@LIBC
+ 0
0000000cd9d0  002400000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_key_delete@LIBC +
0
0000000cd9d8  002500000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_key_create@LIBC +
0
0000000cd9e0  002600000402 R_AARCH64_JUMP_SL 0000000000000000 getauxval@LIBC + 0
0000000cd9e8  002700000402 R_AARCH64_JUMP_SL 0000000000000000 __system_property_get@LIB
C + 0
0000000cd9f0  002800000402 R_AARCH64_JUMP_SL 0000000000000000 strncmp@LIBC + 0
0000000cd9f8  002900000402 R_AARCH64_JUMP_SL 0000000000000000 fprintf@LIBC + 0
0000000cda00  002a00000402 R_AARCH64_JUMP_SL 0000000000000000 fflush@LIBC + 0
0000000cda08  002b00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_wrlock@LIB
C + 0
0000000cda10  002c00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_unlock@LIB
C + 0
0000000cda18  002d00000402 R_AARCH64_JUMP_SL 0000000000000000 dl_iterate_phdr@LIBC + 0
0000000cda20  002e00000402 R_AARCH64_JUMP_SL 0000000000000000 pthread_rwlock_rdlock@LIB
C + 0
0000000cda28  002f00000402 R_AARCH64_JUMP_SL 0000000000000000 fwrite@LIBC + 0
```

至少这里是有函数名的。

看起来是调用了外部的函数 == `imports`

Symbol table

Symbol table '.dynsym' contains 118 entries

```
Symbol table '.dynsym' contains 118 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND 
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_f[...]@LIBC (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@LIBC (2)
     3: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __android_log_print
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __stack[...]@LIBC (2)
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memset@LIBC (2)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncpy@LIBC (2)
     7: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncat@LIBC (2)
     8: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_self@LIBC (2)
     9: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND malloc@LIBC (2)
    10: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND free@LIBC (2)
    11: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND posix_m[...]@LIBC (2)
    12: 0000000000000000     0 OBJECT  GLOBAL DEFAULT  UND __sF@LIBC (2)
    13: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vfprintf@LIBC (2)
    14: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fputc@LIBC (2)
    15: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND vasprintf@LIBC (2)
    16: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND android[...]@LIBC (2)
    17: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND openlog@LIBC (2)
    18: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND syslog@LIBC (2)
    19: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND closelog@LIBC (2)
    20: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND abort@LIBC (2)
    21: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strlen@LIBC (2)
    22: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND realloc@LIBC (2)
    23: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memmove@LIBC (2)
    24: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __memmo[...]@LIBC (2)
    25: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __strlen_chk@LIBC (2)
    26: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memchr@LIBC (2)
    27: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __vsnpr[...]@LIBC (2)
    28: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND memcpy@LIBC (2)
    29: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    30: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    31: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND calloc@LIBC (2)
    32: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strcmp@LIBC (2)
    33: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    34: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread_once@LIBC (2)
    35: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    36: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    37: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    38: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getauxval@LIBC (2)
    39: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __syste[...]@LIBC (2)
    40: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND strncmp@LIBC (2)
    41: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fprintf@LIBC (2)
    42: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fflush@LIBC (2)
    43: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    44: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    45: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND dl_iter[...]@LIBC (3)
    46: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND pthread[...]@LIBC (2)
    47: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fwrite@LIBC (2)
    48: 0000000000044ce8  6608 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    49: 0000000000078a04  2696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
    50: 00000000000a8a58  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    51: 0000000000076128  2160 FUNC    GLOBAL DEFAULT   15 .datadiv_decode4[...]
    52: 000000000008f8e8  8740 FUNC    GLOBAL DEFAULT   15 .datadiv_decode9[...]
...
    70: 000000000008e650  3772 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    71: 00000000000381fc  3696 FUNC    GLOBAL DEFAULT   15 .datadiv_decode3[...]
    72: 000000000006f220  3892 FUNC    GLOBAL DEFAULT   15 .datadiv_decode8[...]
    73: 00000000000a8884     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    74: 00000000000aa438  1436 FUNC    GLOBAL DEFAULT   15 JNI_OnLoad
    75: 0000000000026d98 18656 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
    76: 0000000000033a2c 11972 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
...
   116: 000000000005ed50  6304 FUNC    GLOBAL DEFAULT   15 .datadiv_decode1[...]
   117: 00000000000a5e74     4 FUNC    GLOBAL DEFAULT   15 .datadiv_decode5[...]
```

看起来 = IDA中的Exports = 导出函数

Histogram

Histogram for bucket list length (total of 118 buckets)

```
Histogram for bucket list length (total of 118 buckets):
 Length  Number     % of total  Coverage
      0  40         ( 33.9%)
      1  48         ( 40.7%)     41.0%
      2  22         ( 18.6%)     78.6%
      3  7          (  5.9%)     96.6%
      4  1          (  0.8%)    100.0%
```

Histogram for `.gnu.hash' bucket list length (total of 17 buckets)

```
Histogram for `.gnu.hash' bucket list length (total of 17 buckets):
 Length  Number     % of total  Coverage
      0  0          (  0.0%)
      1  0          (  0.0%)      0.0%
      2  1          (  5.9%)      2.9%
      3  4          ( 23.5%)     20.0%
      4  6          ( 35.3%)     54.3%
      5  5          ( 29.4%)     90.0%
      6  0          (  0.0%)     90.0%
      7  1          (  5.9%)    100.0%
```

Version section

Version symbols section '.gnu.version' contains 118 entries

```
Version symbols section '.gnu.version' contains 118 entries:
 Addr: 0x0000000000000e08  Offset: 0x00000e08  Link: 3 (.dynsym)
  000:   0 (*local*)       2 (LIBC)          2 (LIBC)          1 (*global*)
  004:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  008:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  00c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  010:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  014:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  018:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  01c:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  020:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  024:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  028:   2 (LIBC)          2 (LIBC)          2 (LIBC)          2 (LIBC)
  02c:   2 (LIBC)          3 (LIBC)          2 (LIBC)          2 (LIBC)
  030:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  034:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  038:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  03c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  040:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  044:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  048:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  04c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  050:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  054:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  058:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  05c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  060:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  064:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  068:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  06c:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  070:   1 (*global*)      1 (*global*)      1 (*global*)      1 (*global*)
  074:   1 (*global*)      1 (*global*)
```

Version needs section '.gnu.version_r' contains 2 entries

```
Version needs section '.gnu.version_r' contains 2 entries:
 Addr: 0x0000000000000ef4  Offset: 0x00000ef4  Link: 8 (.dynstr)
  000000: Version: 1  File: libdl.so  Cnt: 1
  0x0020:   Name: LIBC  Flags: none  Version: 3
  0x0010: Version: 1  File: libc.so  Cnt: 1
  0x0030:   Name: LIBC  Flags: none  Version: 2
```

其他

Displaying notes found in: .note.android.ident

```
Displaying notes found in: .note.android.ident
  Owner                 Data size     Description
  Android               0x00000084    NT_VERSION (version)
   description data: 15 00 00 00 72 32 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 32 31 35 38 38 38 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Displaying notes found in: .note.gnu.build-id

```
Displaying notes found in: .note.gnu.build-id
  Owner                 Data size     Description
  GNU                   0x00000014    NT_GNU_BUILD_ID (unique build ID bitstring)
    Build ID: 3a269496ef440c785356664f894646815d3c61ad
```

readelf的help语法帮助

```
➜  ~ readelf --help
Usage: readelf <option(s)> elf-file(s)
 Display information about the contents of ELF format files
 Options are:
  -a --all               Equivalent to: -h -l -S -s -r -d -V -A -I
  -h --file-header       Display the ELF file header
  -l --program-headers   Display the program headers
     --segments          An alias for --program-headers
  -S --section-headers   Display the sections' header
     --sections          An alias for --section-headers
  -g --section-groups    Display the section groups
  -t --section-details   Display the section details
  -e --headers           Equivalent to: -h -l -S
  -s --syms              Display the symbol table
     --symbols           An alias for --syms
     --dyn-syms          Display the dynamic symbol table
     --lto-syms          Display LTO symbol tables
     --sym-base=[0|8|10|16]
                         Force base for symbol sizes.  The options are
                         mixed (the default), octal, decimal, hexadecimal.
  -C --demangle[=STYLE]  Decode mangled/processed symbol names
                           STYLE can be "none", "auto", "gnu-v3", "java",
                           "gnat", "dlang", "rust"
     --no-demangle       Do not demangle low-level symbol names.  (default)
     --recurse-limit     Enable a demangling recursion limit.  (default)
     --no-recurse-limit  Disable a demangling recursion limit
     -U[dlexhi] --unicode=[default|locale|escape|hex|highlight|invalid]
                         Display unicode characters as determined by the current locale
                          (default), escape sequences, "<hex sequences>", highlighted
                          escape sequences, or treat them as invalid and display as
                          "{hex sequences}"
  -n --notes             Display the core notes (if present)
  -r --relocs            Display the relocations (if present)
  -u --unwind            Display the unwind info (if present)
  -d --dynamic           Display the dynamic section (if present)
  -V --version-info      Display the version sections (if present)
  -A --arch-specific     Display architecture specific information (if any)
  -c --archive-index     Display the symbol/file index in an archive
  -D --use-dynamic       Use the dynamic section info when displaying symbols
  -L --lint|--enable-checks
                         Display warning messages for possible problems
  -x --hex-dump=<number|name>
                         Dump the contents of section <number|name> as bytes
  -p --string-dump=<number|name>
                         Dump the contents of section <number|name> as strings
  -R --relocated-dump=<number|name>
                         Dump the relocated contents of section <number|name>
  -z --decompress        Decompress section before dumping it
  -w --debug-dump[a/=abbrev, A/=addr, r/=aranges, c/=cu_index, L/=decodedline,
                  f/=frames, F/=frames-interp, g/=gdb_index, i/=info, o/=loc,
                  m/=macro, p/=pubnames, t/=pubtypes, R/=Ranges, l/=rawline,
                    s/=str, O/=str-offsets, u/=trace_abbrev, T/=trace_aranges,
                    U/=trace_info]
                           Display the contents of DWARF debug sections
  -wk --debug-dump=links Display the contents of sections that link to separate
                            debuginfo files
  -P --process-links     Display the contents of non-debug sections in separate
                            debuginfo files.  (Implies -wK)
  -wK --debug-dump=follow-links
                           Follow links to separate debug info files (default)
  -wN --debug-dump=no-follow-links
                           Do not follow links to separate debug info files
  --dwarf-depth=N        Do not display DIEs at depth N or greater
  --dwarf-start=N        Display DIEs starting at offset N
  --ctf=<number|name>    Display CTF info from section <number|name>
  --ctf-parent=<name>    Use CTF archive member <name> as the CTF parent
  --ctf-symbols=<number|name>
                           Use section <number|name> as the CTF external symtab
  --ctf-strings=<number|name>
                           Use section <number|name> as the CTF external strtab
  --sframe[=NAME]        Display SFrame info from section NAME, (default '.sframe')
  -I --histogram         Display histogram of bucket list lengths
  -W --wide              Allow output width to exceed 80 characters
  -T --silent-truncation If a symbol name is truncated, do not add [...] suffix
  @<file>                Read options from <file>
  -H --help              Display this information
  -v --version           Display the version number of readelf
Report bugs to <https://sourceware.org/bugzilla/>
```

objdump

安装objdump

- objdump属于binutils中的一个
    - 所以直接去安装： `binutils` 即可
        - 注： `macOS` 自带，无需额外安装

objdump用法

概述：

- 单个参数
  - `-a` = `--archive-headers` ：显示存档文件头信息
    - 说明
      - 看一个 `.a` 静态库文件中包含了哪些目标文件
    - 语法

      ```
      objdump -a elfFile
      ```

  - `-f` = `--file-headers` ：显示全部文件头信息
    - 语法

      ```
      objdump -f elfFile
      ```

  - `-h` = `--headers` = `--section-headers` ：显示节的头信息
    - 对比
      - == `readelf -S`
    - 语法

      ```
      objdump -h elfFile
      ```

  - `-s` = `--full-contents` ：显示每个节的内容
    - 语法

      ```
      objdump -s elfFile
      ```

  - `-t` = `--syms` ：显示符号表
    - 语法

      ```
      objdump -t elfFile
      ```

  - `-T` = `--dynamic-syms` ：显示动态符号表
    - 说明
      - 输出目标文件的动态符号表（Dynamic Symbol Table），即目标ELF文件中名字叫做.dynsym节内的内容
      - 通过这张表内的信息，可以看出由本ELF文件中导出的符号，和需要从别的动态库中导入的符号。如果第三列显示“*UND*”表明这个符号在本ELF文件中未定义，也就是说这个符号要从别的动态库中导入，其它的情况表明这个符号由本ELF文件中定义。
    - 语法

      ```
      objdump -T elfFile
      ```

  - `-r` = `--reloc` ：显示静态重定位入口
    - 语法

```
objdump -r elfFile
```

  - `-R` = `--dynamic-reloc` ： 显示动态重定位入口
    - 说明
      - 这个参数仅仅对于动态目标文件有意义，比如动态库文件（ `.so` ）
    - 语法

    ```
    objdump -R elfFile
    ```

  - 反汇编
    - `-d` = `--disassemble` ： 反汇编可执行指令的内容
      - 语法

      ```
      objdump -d elfFile
      ```

    - `-D` = `--disassemble-all` ： 反汇编所有指令的内容
      - 语法

      ```
      objdump -D elfFile
      ```

- 组合参数
  - `-x` = `--all-headers` == `-h --syms --reloc` ： 显示全部头信息
    - 语法

    ```
    objdump -x elfFile
    ```

objdump用法举例

- 输入文件：ELF格式的 `libtacker.so`

用objdump读取解析ELF的 `libtacker.so` 的效果：

-a：显示存档文件头信息

```
➜  arm64-v8a objdump -a libtacker.so

libtacker.so:     file format elf64-littleaarch64
```

-f：显示全部文件头信息

```
➜  arm64-v8a objdump -f libtacker.so

libtacker.so:     file format elf64-littleaarch64
architecture: aarch64
start address: 0x000000000001a5c0
```

-h：显示节的头信息

```
➜  arm64-v8a objdump -h libtacker.so

libtacker.so:     file format elf64-littleaarch64

Sections:
Idx Name                 Size     VMA              Type
  0                      00000000 0000000000000000
  1 .note.android.ident 00000098 0000000000000238
  2 .note.gnu.build-id  00000024 00000000000002d0
  3 .dynsym             00000b10 00000000000002f8
  4 .gnu.version        000000ec 0000000000000e08
  5 .gnu.version_r      00000040 0000000000000ef4
  6 .gnu.hash           000001ec 0000000000000f38
  7 .hash               000003b8 0000000000001124
  8 .dynstr             00000c19 00000000000014dc
  9 .rela.dyn           00008850 00000000000020f8
 10 .rela.plt           00000450 000000000000a948
 11 .gcc_except_table   00001960 000000000000ad98 DATA
 12 .rodata             00003434 000000000000c6f8 DATA
 13 .eh_frame_hdr       00001dbc 000000000000fb2c DATA
 14 .eh_frame           00008cd4 00000000000118e8 DATA
 15 .text               000aec60 000000000001a5c0 TEXT
 16 .plt                00000300 00000000000c9220 TEXT
 17 .data.rel.ro        00002eb8 00000000000ca520 DATA
 18 .fini_array         00000010 00000000000cd3d8
 19 .init_array         00000230 00000000000cd3e8
 20 .dynamic            000001d0 00000000000cd618
 21 .got                000000c0 00000000000cd7e8 DATA
 22 .got.plt            00000188 00000000000cd8a8 DATA
 23 .data               000025d8 00000000000cea30 DATA
 24 .bss                00000ad0 00000000000d1010 BSS
 25 .comment            000000c6 0000000000000000
 26 .shstrtab           00000104 0000000000000000
```

-x：显示全部头信息

```
➜  arm64-v8a objdump -x libtacker.so

libtacker.so:     file format elf64-littleaarch64
architecture: aarch64
start address: 0x000000000001a5c0

Program Header:
    PHDR off    0x0000000000000040 vaddr 0x0000000000000040 paddr 0x0000000000000040 al
ign 2**3
         filesz 0x00000000000001f8 memsz 0x00000000000001f8 flags r--
    LOAD off    0x0000000000000000 vaddr 0x0000000000000000 paddr 0x0000000000000000 al
ign 2**12
         filesz 0x00000000000c9520 memsz 0x00000000000c9520 flags r-x
    LOAD off    0x00000000000c9520 vaddr 0x00000000000ca520 paddr 0x00000000000ca520 al
ign 2**12
         filesz 0x0000000000003510 memsz 0x0000000000003510 flags rw-
    LOAD off    0x00000000000cca30 vaddr 0x00000000000cea30 paddr 0x00000000000cea30 al
ign 2**12
         filesz 0x00000000000025d8 memsz 0x00000000000030b0 flags rw-
 DYNAMIC off    0x00000000000cc618 vaddr 0x00000000000cd618 paddr 0x00000000000cd618 al
ign 2**3
         filesz 0x00000000000001d0 memsz 0x00000000000001d0 flags rw-
   RELRO off    0x00000000000c9520 vaddr 0x00000000000ca520 paddr 0x00000000000ca520 al
ign 2**0
         filesz 0x0000000000003510 memsz 0x0000000000003ae0 flags r--
EH_FRAME off    0x000000000000fb2c vaddr 0x000000000000fb2c paddr 0x000000000000fb2c al
ign 2**2
         filesz 0x0000000000001dbc memsz 0x0000000000001dbc flags r--
   STACK off    0x0000000000000000 vaddr 0x0000000000000000 paddr 0x0000000000000000 al
ign 2**64
         filesz 0x0000000000000000 memsz 0x0000000000000000 flags rw-
    NOTE off    0x0000000000000238 vaddr 0x0000000000000238 paddr 0x0000000000000238 al
ign 2**2
         filesz 0x00000000000000bc memsz 0x00000000000000bc flags r--

Dynamic Section:
  NEEDED       liblog.so
  NEEDED       libm.so
  NEEDED       libdl.so
  NEEDED       libc.so
  SONAME       libforce.so
  FLAGS        0x0000000000000008
  FLAGS_1      0x0000000000000001
  RELA         0x00000000000020f8
  RELASZ       0x0000000000008850
  RELAENT      0x0000000000000018
  RELACOUNT    0x0000000000000568
  JMPREL       0x000000000000a948
  PLTRELSZ     0x0000000000000450
  PLTGOT       0x00000000000cd8a8
  PLTREL       0x0000000000000007
  SYMTAB       0x00000000000002f8
  SYMENT       0x0000000000000018
  STRTAB       0x00000000000014dc
  STRSZ        0x0000000000000c19
  GNU_HASH     0x0000000000000f38
  HASH         0x0000000000001124
  INIT_ARRAY   0x00000000000cd3e8
  INIT_ARRAYSZ 0x0000000000000230
  FINI_ARRAY   0x00000000000cd3d8
  FINI_ARRAYSZ 0x0000000000000010
  VERSYM       0x0000000000000e08
  VERNEED      0x0000000000000ef4
  VERNEEDNUM   0x0000000000000002

Version References:
  required from libdl.so:
    0x00050d63 0x00 03 LIBC
  required from libc.so:
    0x00050d63 0x00 02 LIBC

Sections:
Idx Name                 Size     VMA              Type
  0                      00000000 0000000000000000
  1 .note.android.ident  00000098 0000000000000238
  2 .note.gnu.build-id   00000024 00000000000002d0
  3 .dynsym              00000b10 00000000000002f8
  4 .gnu.version         000000ec 0000000000000e08
  5 .gnu.version_r       00000040 0000000000000ef4
  6 .gnu.hash            000001ec 0000000000000f38
  7 .hash                000003b8 0000000000001124
  8 .dynstr              00000c19 00000000000014dc
  9 .rela.dyn            00008850 00000000000020f8
 10 .rela.plt            00000450 000000000000a948
 11 .gcc_except_table    00001960 000000000000ad98 DATA
 12 .rodata              00003434 000000000000c6f8 DATA
 13 .eh_frame_hdr        00001dbc 000000000000fb2c DATA
 14 .eh_frame            00008cd4 00000000000118e8 DATA
 15 .text                000aec60 000000000001a5c0 TEXT
 16 .plt                 00000300 00000000000c9220 TEXT
 17 .data.rel.ro         00002eb8 00000000000ca520 DATA
 18 .fini_array          00000010 00000000000cd3d8
 19 .init_array          00000230 00000000000cd3e8
 20 .dynamic             000001d0 00000000000cd618
 21 .got                 000000c0 00000000000cd7e8 DATA
 22 .got.plt             00000188 00000000000cd8a8 DATA
 23 .data                000025d8 00000000000cea30 DATA
 24 .bss                 00000ad0 00000000000d1010 BSS
 25 .comment             000000c6 0000000000000000
 26 .shstrtab            00000104 0000000000000000
```

```
SYMBOL TABLE:
```

-d：反汇编可执行指令的内容

text段反汇编

相关参数：

```
       -d
       --disassemble
       --disassemble=symbol
           Display the assembler mnemonics for the machine instructions
           from the input file.  This option only disassembles those
           sections which are expected to contain instructions.  If the
           optional symbol argument is given, then display the assembler
           mnemonics starting at symbol.  If symbol is a function name
           then disassembly will stop at the end of the function,
           otherwise it will stop when the next symbol is encountered.
           If there are no matches for symbol then nothing will be
           displayed.

           Note if the --dwarf=follow-links option is enabled then any
           symbol tables in linked debug info files will be read in and
           used when disassembling.

       -j name
       --section=name
           Display information only for section name.
```

->

```
➜  arm64-v8a objdump -d -j .text libtacker.so
...
   c8e54: 60 ca 00 39      strb    w0, [x19, #50]
   c8e58: e8 02 40 39      ldrb    w8, [x23]
   c8e5c: 1f e9 01 71      cmp     w8, #122
   c8e60: 80 01 00 54      b.eq    0xc8e90 <.datadiv_decode2444497212690810360+0x1e4bc>
   c8e64: 44 00 00 14      b       0xc8f74 <.datadiv_decode2444497212690810360+0x1e5a0>
   c8e68: 20 fa ff 90      adrp    x0, 0xc000 <.datadiv_decode2444497212690810360+0x1e1a
4>
   c8e6c: 00 ec 3d 91      add     x0, x0, #3963
   c8e70: 46 00 00 14      b       0xc8f88 <.datadiv_decode2444497212690810360+0x1e5b4>
   c8e74: a0 63 00 91      add     x0, x29, #24
   c8e78: e1 03 16 aa      mov     x1, x22
   c8e7c: b5 fa ff 97      bl      0xc7950 <.datadiv_decode2444497212690810360+0x1cf7c>
   c8e80: 60 ca 00 39      strb    w0, [x19, #50]
   c8e84: e8 02 40 39      ldrb    w8, [x23]
   c8e88: 1f e9 01 71      cmp     w8, #122
   c8e8c: 41 07 00 54      b.ne    0xc8f74 <.datadiv_decode2444497212690810360+0x1e5a0>
   c8e90: a0 63 00 91      add     x0, x29, #24
   c8e94: e1 03 16 aa      mov     x1, x22
   c8e98: ae fa ff 97      bl    0xc7950 <.datadiv_decode2444497212690810360+0x1cf7c>
   c8e9c: 38 fa ff f0      adrp    x24, 0xf000 <.datadiv_decode2444497212690810360+0x1e
1e4>
   c8ea0: 18 73 25 91      add    x24, x24, #2396
   c8ea4: 39 00 80 52      mov    w25, #1
   c8ea8: 05 00 00 14      b    0xc8ebc <.datadiv_decode2444497212690810360+0x1e4e8>
   c8eac: 9f 24 03 d5      hint    #36
   c8eb0: 79 ce 00 39      strb    w25, [x19, #51]
   c8eb4: 9f 24 03 d5      hint    #36
   c8eb8: f7 06 00 91      add    x23, x23, #1
...
   c9174: 10 7c 40 f9      ldr    x16, [x0, #248]
   c9178: 00 04 40 a9      ldp    x0, x1, [x0]
   c917c: 1f 02 00 91      mov    sp, x16
   c9180: c0 03 5f d6      ret
   c9184: 5f 24 03 d5      hint    #34
   c9188: 00 04 00 a9      stp    x0, x1, [x0]
   c918c: 02 0c 01 a9      stp    x2, x3, [x0, #16]
   c9190: 04 14 02 a9      stp    x4, x5, [x0, #32]
   c9194: 06 1c 03 a9      stp    x6, x7, [x0, #48]
   c9198: 08 24 04 a9      stp    x8, x9, [x0, #64]
   c919c: 0a 2c 05 a9      stp    x10, x11, [x0, #80]
   c91a0: 0c 34 06 a9      stp    x12, x13, [x0, #96]
   c91a4: 0e 3c 07 a9      stp    x14, x15, [x0, #112]
   c91a8: 10 44 08 a9      stp    x16, x17, [x0, #128]
   c91ac: 12 4c 09 a9      stp    x18, x19, [x0, #144]
   c91b0: 14 54 0a a9      stp    x20, x21, [x0, #160]
   c91b4: 16 5c 0b a9      stp    x22, x23, [x0, #176]
   c91b8: 18 64 0c a9      stp    x24, x25, [x0, #192]
   c91bc: 1a 6c 0d a9      stp    x26, x27, [x0, #208]
   c91c0: 1c 74 0e a9      stp    x28, x29, [x0, #224]
   c91c4: 1e 78 00 f9      str    x30, [x0, #240]
   c91c8: e1 03 00 91      mov    x1, sp
   c91cc: 01 7c 00 f9      str    x1, [x0, #248]
   c91d0: 1e 80 00 f9      str    x30, [x0, #256]
   c91d4: 00 04 11 6d      stp    d0, d1, [x0, #272]
   c91d8: 02 0c 12 6d      stp    d2, d3, [x0, #288]
   c91dc: 04 14 13 6d      stp    d4, d5, [x0, #304]
   c91e0: 06 1c 14 6d      stp    d6, d7, [x0, #320]
   c91e4: 08 24 15 6d      stp    d8, d9, [x0, #336]
   c91e8: 0a 2c 16 6d      stp    d10, d11, [x0, #352]
   c91ec: 0c 34 17 6d      stp    d12, d13, [x0, #368]
   c91f0: 0e 3c 18 6d      stp    d14, d15, [x0, #384]
   c91f4: 10 44 19 6d      stp    d16, d17, [x0, #400]
   c91f8: 12 4c 1a 6d      stp    d18, d19, [x0, #416]
   c91fc: 14 54 1b 6d      stp    d20, d21, [x0, #432]
   c9200: 16 5c 1c 6d      stp    d22, d23, [x0, #448]
   c9204: 18 64 1d 6d      stp    d24, d25, [x0, #464]
   c9208: 1a 6c 1e 6d      stp    d26, d27, [x0, #480]
   c920c: 1c 74 1f 6d      stp    d28, d29, [x0, #496]
   c9210: 1e 00 01 fd      str    d30, [x0, #512]
   c9214: 1f 04 01 fd      str    d31, [x0, #520]
   c9218: 00 00 80 d2      mov    x0, #0
   c921c: c0 03 5f d6      ret
```

内容太多了。

看来是把全部的汇编都反编译了。

reloc段反汇编

相关参数：

```
       -d
       --disassemble
       --disassemble=symbol
           Display the assembler mnemonics for the machine instructions
           from the input file.  This option only disassembles those
           sections which are expected to contain instructions.  If the
           optional symbol argument is given, then display the assembler
           mnemonics starting at symbol.  If symbol is a function name
           then disassembly will stop at the end of the function,
           otherwise it will stop when the next symbol is encountered.
           If there are no matches for symbol then nothing will be
           displayed.


           Note if the --dwarf=follow-links option is enabled then any
           symbol tables in linked debug info files will be read in and
           used when disassembling.


       -r
       --reloc
           Print the relocation entries of the file.  If used with -d or
           -D, the relocations are printed interspersed with the
           disassembly.
```

->

```
➜  arm64-v8a objdump -d -r libtacker.so > libtacker_objdump_d_r.coffee

libtacker.so:     file format elf64-littleaarch64


Disassembly of section .text:


000000000001a5c0 <.text>:
   1a5c0: 5f 24 03 d5      hint    #34
   1a5c4: 80 05 00 90      adrp    x0, 0xca000 <.text+0x2c4>
   1a5c8: 00 80 14 91      add     x0, x0, #1312
   1a5cc: 1d bb 02 14      b    0xc9240 <__cxa_finalize@plt>
   1a5d0: 5f 24 03 d5      hint    #34
   1a5d4: c0 03 5f d6      ret
   1a5d8: 5f 24 03 d5      hint    #34
   1a5dc: ac a9 02 14      b    0xc4c8c <.datadiv_decode2444497212690810360+0x1a2b8>
   1a5e0: 5f 24 03 d5      hint    #34
   1a5e4: 60 00 00 b4      cbz     x0, 0x1a5f0 <.text+0x30>
   1a5e8: f0 03 00 aa      mov    x16, x0
   1a5ec: 00 02 1f d6      br    x16
   1a5f0: c0 03 5f d6      ret
   1a5f4: 5f 24 03 d5      hint    #34
   1a5f8: 08 00 00 90      adrp    x8, 0x1a000 <.text+0x38>
   1a5fc: 08 81 17 91      add    x8, x8, #1504
   1a600: 82 05 00 90      adrp    x2, 0xca000 <.text+0x300>
   1a604: 42 80 14 91      add    x2, x2, #1312
...
   2e11c: 08 b6 b3 72      movk    w8, #40368, lsl #16
   2e120: 1f 01 17 6b      cmp    w8, w23
   2e124: 60 01 00 54      b.eq    0x2e150 <.datadiv_decode17838636323198310142+0x2820>
   2e128: 1f 01 16 6b      cmp    w8, w22
   2e12c: a1 ff ff 54      b.ne    0x2e120 <.datadiv_decode17838636323198310142+0x27f0>
   2e130: e0 03 14 aa      mov    x0, x20
   2e134: e1 03 13 aa      mov    x1, x19
   2e138: e2 03 15 2a      mov    w2, w21
   2e13c: e3 03 1f 2a      mov    w3, wzr
   2e140: 03 08 00 94      bl    0x3014c <.datadiv_decode8952246851265070369+0x6b4>
   2e144: 28 95 94 52      mov    w8, #42153
   2e148: a8 a2 a4 72      movk    w8, #9493, lsl #16
   2e14c: f5 ff ff 17      b    0x2e120 <.datadiv_decode17838636323198310142+0x27f0>
   2e150: f4 4f 43 a9      ldp    x20, x19, [sp, #48]
   2e154: f6 57 42 a9      ldp    x22, x21, [sp, #32]
   2e158: f7 0b 40 f9      ldr    x23, [sp, #16]
   2e15c: fd 7b c4 a8      ldp    x29, x30, [sp], #64
   2e160: c0 03 5f d6      ret
   2e164: ff 43 01 d1      sub    sp, sp, #80
   2e168: fd 7b 02 a9      stp    x29, x30, [sp, #32]
   2e16c: fd 83 00 91      add    x29, sp, #32
...


000000000002e2bc <.datadiv_decode18328417529454547004>:
   2e2bc: fd 7b ba a9      stp    x29, x30, [sp, #-96]!
   2e2c0: fc 6f 01 a9      stp    x28, x27, [sp, #16]
   2e2c4: fa 67 02 a9      stp    x26, x25, [sp, #32]
   2e2c8: f8 5f 03 a9      stp    x24, x23, [sp, #48]
   2e2cc: f6 57 04 a9      stp    x22, x21, [sp, #64]
   2e2d0: f4 4f 05 a9      stp    x20, x19, [sp, #80]
   2e2d4: 01 05 00 90      adrp    x1, 0xce000 <.datadiv_decode18328417529454547004+0x29
8>
   2e2d8: d1 0e 80 52      mov    w17, #118
   2e2dc: 21 c0 33 91      add    x1, x1, #3312
   2e2e0: 69 0b 80 52      mov    w9, #91
   2e2e4: 6f 0b 80 12      mov    w15, #-92
   2e2e8: a0 05 80 12      mov    w0, #-46
   2e2ec: d0 0e 80 12      mov    w16, #-119
   2e2f0: cd 02 80 52      mov    w13, #22
   2e2f4: 28 00 40 39      ldrb    w8, [x1]
   2e2f8: 85 0d 80 12      mov    w5, #-109
   2e2fc: 95 0d 80 52      mov    w21, #108
   2e300: 27 0e 80 52      mov    w7, #113
   2e304: 43 0f 80 52      mov    w3, #122
   2e308: 19 0d 80 52      mov    w25, #104
   2e30c: ea 03 28 2a      mvn    w10, w8
```

```
    2e310: 0b 01 28 0a      bic    w11, w8, w8
    2e314: 0c 05 1a 12      and    w12, w8, #0xc0
...


    c913c: 06 1c 54 6d      ldp    d6, d7, [x0, #320]
    c9140: 08 24 55 6d      ldp    d8, d9, [x0, #336]
    c9144: 0a 2c 56 6d      ldp    d10, d11, [x0, #352]
    c9148: 0c 34 57 6d      ldp    d12, d13, [x0, #368]
    c914c: 0e 3c 58 6d      ldp    d14, d15, [x0, #384]
    c9150: 10 44 59 6d      ldp    d16, d17, [x0, #400]
    c9154: 12 4c 5a 6d      ldp    d18, d19, [x0, #416]
    c9158: 14 54 5b 6d      ldp    d20, d21, [x0, #432]
    c915c: 16 5c 5c 6d      ldp    d22, d23, [x0, #448]
    c9160: 18 64 5d 6d      ldp    d24, d25, [x0, #464]
    c9164: 1a 6c 5e 6d      ldp    d26, d27, [x0, #480]
    c9168: 1c 74 5f 6d      ldp    d28, d29, [x0, #496]
    c916c: 1e 00 41 fd      ldr    d30, [x0, #512]
    c9170: 1f 04 41 fd      ldr    d31, [x0, #520]
    c9174: 10 7c 40 f9      ldr    x16, [x0, #248]
    c9178: 00 04 40 a9      ldp    x0, x1, [x0]
    c917c: 1f 02 00 91      mov    sp, x16
    c9180: c0 03 5f d6      ret
    c9184: 5f 24 03 d5      hint    #34
    c9188: 00 04 00 a9      stp    x0, x1, [x0]
    c918c: 02 0c 01 a9      stp    x2, x3, [x0, #16]
    c9190: 04 14 02 a9      stp    x4, x5, [x0, #32]
    c9194: 06 1c 03 a9      stp    x6, x7, [x0, #48]
    c9198: 08 24 04 a9      stp    x8, x9, [x0, #64]
    c919c: 0a 2c 05 a9      stp    x10, x11, [x0, #80]
    c91a0: 0c 34 06 a9      stp    x12, x13, [x0, #96]
    c91a4: 0e 3c 07 a9      stp    x14, x15, [x0, #112]
    c91a8: 10 44 08 a9      stp    x16, x17, [x0, #128]
    c91ac: 12 4c 09 a9      stp    x18, x19, [x0, #144]
    c91b0: 14 54 0a a9      stp    x20, x21, [x0, #160]
    c91b4: 16 5c 0b a9      stp    x22, x23, [x0, #176]
    c91b8: 18 64 0c a9      stp    x24, x25, [x0, #192]
    c91bc: 1a 6c 0d a9      stp    x26, x27, [x0, #208]
    c91c0: 1c 74 0e a9      stp    x28, x29, [x0, #224]
    c91c4: 1e 78 00 f9      str    x30, [x0, #240]
    c91c8: e1 03 00 91      mov    x1, sp
    c91cc: 01 7c 00 f9      str    x1, [x0, #248]
    c91d0: 1e 80 00 f9      str    x30, [x0, #256]
    c91d4: 00 04 11 6d      stp    d0, d1, [x0, #272]
    c91d8: 02 0c 12 6d      stp    d2, d3, [x0, #288]
    c91dc: 04 14 13 6d      stp    d4, d5, [x0, #304]
    c91e0: 06 1c 14 6d      stp    d6, d7, [x0, #320]
    c91e4: 08 24 15 6d      stp    d8, d9, [x0, #336]
    c91e8: 0a 2c 16 6d      stp    d10, d11, [x0, #352]
    c91ec: 0c 34 17 6d      stp    d12, d13, [x0, #368]
    c91f0: 0e 3c 18 6d      stp    d14, d15, [x0, #384]
    c91f4: 10 44 19 6d      stp    d16, d17, [x0, #400]
    c91f8: 12 4c 1a 6d      stp    d18, d19, [x0, #416]
    c91fc: 14 54 1b 6d      stp    d20, d21, [x0, #432]
    c9200: 16 5c 1c 6d      stp    d22, d23, [x0, #448]
    c9204: 18 64 1d 6d      stp    d24, d25, [x0, #464]
```

```
   c9208: 1a 6c 1e 6d      stp    d26, d27, [x0, #480]
   c920c: 1c 74 1f 6d      stp    d28, d29, [x0, #496]
   c9210: 1e 00 01 fd      str    d30, [x0, #512]
   c9214: 1f 04 01 fd      str    d31, [x0, #520]
   c9218: 00 00 80 d2      mov    x0, #0
   c921c: c0 03 5f d6      ret


Disassembly of section .plt:


00000000000c9220 <.plt>:
   c9220: f0 7b bf a9      stp    x16, x30, [sp, #-16]!
   c9224: 30 00 00 90      adrp    x16, 0xcd000 <.plt+0x14>
   c9228: 11 5e 44 f9      ldr    x17, [x16, #2232]
   c922c: 10 e2 22 91      add    x16, x16, #2232
   c9230: 20 02 1f d6      br    x17
   c9234: 1f 20 03 d5      nop
   c9238: 1f 20 03 d5      nop
   c923c: 1f 20 03 d5      nop


00000000000c9240 <__cxa_finalize@plt>:
   c9240: 30 00 00 90      adrp    x16, 0xcd000 <__cxa_atexit@plt>
   c9244: 11 62 44 f9      ldr    x17, [x16, #2240]
   c9248: 10 02 23 91      add    x16, x16, #2240
   c924c: 20 02 1f d6      br    x17

...

00000000000c9370 <strlen@plt>:
   c9370: 30 00 00 90      adrp    x16, 0xcd000 <realloc@plt>
   c9374: 11 ae 44 f9      ldr    x17, [x16, #2392]
   c9378: 10 62 25 91      add    x16, x16, #2392
   c937c: 20 02 1f d6      br    x17

...

00000000000c9480 <getauxval@plt>:
   c9480: 30 00 00 90      adrp    x16, 0xcd000 <__system_property_get@plt>
   c9484: 11 f2 44 f9      ldr    x17, [x16, #2528]
   c9488: 10 82 27 91      add    x16, x16, #2528
   c948c: 20 02 1f d6      br    x17


00000000000c9490 <__system_property_get@plt>:
   c9490: 30 00 00 90      adrp    x16, 0xcd000 <strncmp@plt>
   c9494: 11 f6 44 f9      ldr    x17, [x16, #2536]
   c9498: 10 a2 27 91      add    x16, x16, #2536
   c949c: 20 02 1f d6      br    x17


00000000000c94a0 <strncmp@plt>:
   c94a0: 30 00 00 90      adrp    x16, 0xcd000 <fprintf@plt>
   c94a4: 11 fa 44 f9      ldr    x17, [x16, #2544]
   c94a8: 10 c2 27 91      add    x16, x16, #2544
   c94ac: 20 02 1f d6       br    x17

...

00000000000c94e0 <pthread_rwlock_unlock@plt>:
   c94e0: 30 00 00 90       adrp    x16, 0xcd000 <dl_iterate_phdr@plt>
   c94e4: 11 0a 45 f9       ldr    x17, [x16, #2576]
   c94e8: 10 42 28 91       add    x16, x16, #2576
   c94ec: 20 02 1f d6       br    x17


00000000000c94f0 <dl_iterate_phdr@plt>:
   c94f0: 30 00 00 90       adrp    x16, 0xcd000 <pthread_rwlock_rdlock@plt>
   c94f4: 11 0e 45 f9       ldr    x17, [x16, #2584]
   c94f8: 10 62 28 91       add    x16, x16, #2584
   c94fc: 20 02 1f d6       br    x17


00000000000c9500 <pthread_rwlock_rdlock@plt>:
   c9500: 30 00 00 90       adrp    x16, 0xcd000 <fwrite@plt>
   c9504: 11 12 45 f9       ldr    x17, [x16, #2592]
   c9508: 10 82 28 91       add    x16, x16, #2592
   c950c: 20 02 1f d6       br    x17


00000000000c9510 <fwrite@plt>:
   c9510: 30 00 00 90       adrp    x16, 0xcd000 <fwrite@plt+0x10>
   c9514: 11 16 45 f9       ldr    x17, [x16, #2600]
   c9518: 10 a2 28 91       add    x16, x16, #2600
   c951c: 20 02 1f d6       br    x17
```

-s：显示每个节的内容

```
➜  arm64-v8a objdump -s libtacker.so > libtacker_objdump_s.coffee
...
libtacker.so:    file format elf64-littleaarch64
Contents of section .note.android.ident:
 0238 08000000 84000000 01000000 416e6472  ............Andr
 0248 6f696400 15000000 72323400 00000000  oid.....r24.....
 0258 00000000 00000000 00000000 00000000  ................
 0268 00000000 00000000 00000000 00000000  ................
 0278 00000000 00000000 00000000 00000000  ................
 0288 00000000 00000000 38323135 38383800  ........8215888.
 0298 00000000 00000000 00000000 00000000  ................
 02a8 00000000 00000000 00000000 00000000  ................
 02b8 00000000 00000000 00000000 00000000  ................
 02c8 00000000 00000000                    ........
Contents of section .note.gnu.build-id:
 02d0 04000000 14000000 03000000 474e5500  ............GNU.
 02e0 3a269496 ef440c78 5356664f 89464681  :&...D.xSVfO.FF.
 02f0 5d3c61ad                             ]<a.
Contents of section .dynsym:
 02f8 00000000 00000000 00000000 00000000  ................
 0308 00000000 00000000 01000000 12000000  ................
 0318 00000000 00000000 00000000 00000000  ................
 0328 10000000 12000000 00000000 00000000  ................
 0338 00000000 00000000 1d000000 12000000  ................
 0348 00000000 00000000 00000000 00000000  ................
...
 d0f70 5a565416 5f564b5a 5c165849 49165550  ZVT._VKZ\.XII.UP
 d0f80 5b167758 4d504f5c 715c5549 5c4b3900  [.wXMPO\q\UI\K9.
 d0f90 a9cde0ef e5f3eee8 e5aee0f1 f1aec0f1  ................
 d0fa0 f1ede8e2 e0f5e8ee efbaa8d7 81000000  ................
 d0fb0 10000000 00000000 08000000 00000000  ................
 d0fc0 00000000 00000000 00000000 00000000  ................
 d0fd0 00000000 00000000 00000000 00000000  ................
 d0fe0 00000000 00000000 00000000 00000000  ................
 d0ff0 00000000 00000000 00000000 00000000  ................
 d1000 00000000 00000000                    ........
Contents of section .bss:
<skipping contents of bss section at [d1010, d1ae0)>
Contents of section .comment:
 0000 4c696e6b 65723a20 4c4c4420 31342e30  Linker: LLD 14.0
 0010 2e310063 6c616e67 20766572 73696f6e  .1.clang version
 0020 2031342e 302e3000 416e6472 6f696420   14.0.0.Android
 0030 28383037 35313738 2c206261 73656420  (8075178, based
 0040 6f6e2072 34333731 31326229 20636c61  on r437112b) cla
 0050 6e672076 65727369 6f6e2031 342e302e  ng version 14.0.
 0060 31202868 74747073 3a2f2f61 6e64726f  1 (https://andro
 0070 69642e67 6f6f676c 65736f75 7263652e  id.googlesource.
 0080 636f6d2f 746f6f6c 63686169 6e2f6c6c  com/toolchain/ll
 0090 766d2d70 726f6a65 63742038 36373133  vm-project 86713
 00a0 34386238 31623935 66633630 33353035  48b81b95fc603505
 00b0 64666338 38316234 35313033 62656531  dfc881b45103bee1
 00c0 37333129 0000                        731)..
Contents of section .shstrtab:
 0000 002e696e 69745f61 72726179 002e6669  ..init_array..fi
 0010 6e695f61 72726179 002e7465 7874002e  ni_array..text..
 0020 676f7400 2e636f6d 6d656e74 002e6e6f  got..comment..no
 0030 74652e61 6e64726f 69642e69 64656e74  te.android.ident
 0040 002e676f 742e706c 74002e72 656c612e  ..got.plt..rela.
 0050 706c7400 2e627373 002e6479 6e737472  plt..bss..dynstr
 0060 002e6568 5f667261 6d655f68 6472002e  ..eh_frame_hdr..
 0070 676e752e 76657273 696f6e5f 72002e64  gnu.version_r..d
 0080 6174612e 72656c2e 726f002e 72656c61  ata.rel.ro..rela
 0090 2e64796e 002e676e 752e7665 7273696f  .dyn..gnu.versio
 00a0 6e002e64 796e7379 6d002e67 6e752e68  n..dynsym..gnu.h
 00b0 61736800 2e65685f 6672616d 65002e67  ash..eh_frame..g
 00c0 63635f65 78636570 745f7461 626c6500  cc_except_table.
 00d0 2e6e6f74 652e676e 752e6275 696c642d  .note.gnu.build-
 00e0 6964002e 64796e61 6d696300 2e736873  id..dynamic..shs
 00f0 74727461 62002e72 6f646174 61002e64  trtab..rodata..d
 0100 61746100                             ata.
```

-t：显示符号表

```
➜ arm64-v8a objdump -t libtacker.so
```

```
libtacker.so:     file format elf64-littleaarch64

SYMBOL TABLE:
```

-T：显示动态符号表

```
➜  arm64-v8a objdump -T libtacker.so


libtacker.so:     file format elf64-littleaarch64


DYNAMIC SYMBOL TABLE:
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __cxa_finalize
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __cxa_atexit
0000000000000000      DF *UND*    0000000000000000              __android_log_print
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __stack_chk_fail
0000000000000000      DF *UND*    0000000000000000 (LIBC)       memset
0000000000000000      DF *UND*    0000000000000000 (LIBC)       strncpy
0000000000000000      DF *UND*    0000000000000000 (LIBC)       strncat
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_self
0000000000000000      DF *UND*    0000000000000000 (LIBC)       malloc
0000000000000000      DF *UND*    0000000000000000 (LIBC)       free
0000000000000000      DF *UND*    0000000000000000 (LIBC)       posix_memalign
0000000000000000      DO *UND*    0000000000000000 (LIBC)       __sF
0000000000000000      DF *UND*    0000000000000000 (LIBC)       vfprintf
0000000000000000      DF *UND*    0000000000000000 (LIBC)       fputc
0000000000000000      DF *UND*    0000000000000000 (LIBC)       vasprintf
0000000000000000      DF *UND*    0000000000000000 (LIBC)       android_set_abort_messa
ge
0000000000000000      DF *UND*    0000000000000000 (LIBC)       openlog
0000000000000000      DF *UND*    0000000000000000 (LIBC)       syslog
0000000000000000      DF *UND*    0000000000000000 (LIBC)       closelog
0000000000000000      DF *UND*    0000000000000000 (LIBC)       abort
0000000000000000      DF *UND*    0000000000000000 (LIBC)       strlen
0000000000000000      DF *UND*    0000000000000000 (LIBC)       realloc
0000000000000000      DF *UND*    0000000000000000 (LIBC)       memmove
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __memmove_chk
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __strlen_chk
0000000000000000      DF *UND*    0000000000000000 (LIBC)       memchr
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __vsnprintf_chk
0000000000000000      DF *UND*    0000000000000000 (LIBC)       memcpy
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_mutex_lock
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_mutex_unlock
0000000000000000      DF *UND*    0000000000000000 (LIBC)       calloc
0000000000000000      DF *UND*    0000000000000000 (LIBC)       strcmp
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_getspecific
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_once
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_setspecific
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_key_delete
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_key_create
0000000000000000      DF *UND*    0000000000000000 (LIBC)       getauxval
0000000000000000      DF *UND*    0000000000000000 (LIBC)       __system_property_get
```

```
0000000000000000      DF *UND*    0000000000000000 (LIBC)       strncmp
0000000000000000      DF *UND*    0000000000000000 (LIBC)       fprintf
0000000000000000      DF *UND*    0000000000000000 (LIBC)       fflush
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_rwlock_wrlock
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_rwlock_unlock
0000000000000000      DF *UND*    0000000000000000 (LIBC)       dl_iterate_phdr
0000000000000000      DF *UND*    0000000000000000 (LIBC)       pthread_rwlock_rdlock
0000000000000000      DF *UND*    0000000000000000 (LIBC)       fwrite
0000000000044ce8 g    DF .text    00000000000019d0              .datadiv_decode16117807
209816376729
0000000000078a04 g    DF .text    0000000000000a88              .datadiv_decode99019400
71257331957
00000000000a8a58 g    DF .text    0000000000000f34              .datadiv_decode14716202
181486223822
0000000000076128 g    DF .text    0000000000000870              .datadiv_decode48915969
74783633952
...
00000000000a8884 g    DF .text    0000000000000004              .datadiv_decode11706101
414295225912
00000000000aa438 g    DF .text    000000000000059c              JNI_OnLoad
0000000000026d98 g    DF .text    00000000000048e0              .datadiv_decode12335027
288954124723
...
000000000005c790 g    DF .text    0000000000000004              .datadiv_decode15522057
00074701063
000000000005ed50 g    DF .text    00000000000018a0              .datadiv_decode15147620
753704794795
00000000000a5e74 g    DF .text    0000000000000004              .datadiv_decode54544065
52017557296
```

-r：显示静态重定位入口

```
➜  arm64-v8a objdump -r libtacker.so

libtacker.so:     file format elf64-littleaarch64
```

-R：显示动态重定位入口

```
➜  arm64-v8a objdump -R libtacker.so
...


00000000000cd858 R_AARCH64_RELATIVE       *ABS*+0xcc650
00000000000cd860 R_AARCH64_RELATIVE       *ABS*+0xf3b9
00000000000cd868 R_AARCH64_RELATIVE       *ABS*+0xcc668
00000000000cd870 R_AARCH64_RELATIVE       *ABS*+0xf3b0
00000000000cd878 R_AARCH64_RELATIVE       *ABS*+0xcc680
00000000000cd880 R_AARCH64_RELATIVE       *ABS*+0xcc698
00000000000cd888 R_AARCH64_RELATIVE       *ABS*+0xcd1a8
00000000000cd890 R_AARCH64_RELATIVE       *ABS*+0xcd1d0
00000000000cd898 R_AARCH64_RELATIVE       *ABS*+0xcd2a0
00000000000cd8a0 R_AARCH64_RELATIVE       *ABS*+0xcd2c8
```

```
00000000000cece8 R_AARCH64_RELATIVE       *ABS*+0xc206c
00000000000d0fd0 R_AARCH64_RELATIVE       *ABS*+0xacb9c
00000000000d0fd8 R_AARCH64_RELATIVE       *ABS*+0xaccbc
00000000000d0fe0 R_AARCH64_RELATIVE       *ABS*+0xd4a7
00000000000d0fe8 R_AARCH64_RELATIVE       *ABS*+0xd12a1
00000000000d0ff0 R_AARCH64_RELATIVE       *ABS*+0xd12e0
00000000000d0ff8 R_AARCH64_RELATIVE       *ABS*+0xd12e0
00000000000d1000 R_AARCH64_RELATIVE       *ABS*+0xd1ae0
00000000000cd828 R_AARCH64_GLOB_DAT       __sF
00000000000cd460 R_AARCH64_ABS64          .datadiv_decode16117807209816376729
00000000000cd500 R_AARCH64_ABS64          .datadiv_decode9901940071257331957
00000000000cd5f8 R_AARCH64_ABS64          .datadiv_decode14716202181486223822
00000000000cd4e8 R_AARCH64_ABS64          .datadiv_decode4891596974783633952
00000000000cd570 R_AARCH64_ABS64          .datadiv_decode9339716730803528690
00000000000cd488 R_AARCH64_ABS64          .datadiv_decode13183548145838600894
00000000000cd490 R_AARCH64_ABS64          .datadiv_decode3525000441545555282
00000000000cd4a0 R_AARCH64_ABS64          .datadiv_decode8298916886736451040
00000000000cd4e0 R_AARCH64_ABS64          .datadiv_decode6610309240369344955
00000000000cd5b8 R_AARCH64_ABS64          .datadiv_decode5742785139195766122
00000000000cd410 R_AARCH64_ABS64          .datadiv_decode12946686905750037712
00000000000cd4c0 R_AARCH64_ABS64          .datadiv_decode12620377358555187834
00000000000cd520 R_AARCH64_ABS64          .datadiv_decode15294134973280561020
00000000000cd578 R_AARCH64_ABS64          .datadiv_decode123525175395121774
00000000000cd598 R_AARCH64_ABS64          .datadiv_decode17486258817593906496
00000000000cd470 R_AARCH64_ABS64          .datadiv_decode15789135661111847785
00000000000cd508 R_AARCH64_ABS64          .datadiv_decode9708024290202508361
00000000000cd550 R_AARCH64_ABS64          .datadiv_decode12389017544255092540
00000000000cd5a8 R_AARCH64_ABS64          .datadiv_decode8646520184225136992
00000000000cd528 R_AARCH64_ABS64          .datadiv_decode5555087159661513939
00000000000cd538 R_AARCH64_ABS64          .datadiv_decode18018071361702630102
00000000000cd548 R_AARCH64_ABS64          .datadiv_decode11327300974587163078
00000000000cd568 R_AARCH64_ABS64          .datadiv_decode13214095259256631718
00000000000cd428 R_AARCH64_ABS64          .datadiv_decode3631146530348700705
00000000000cd4d0 R_AARCH64_ABS64          .datadiv_decode8050698040297613930
00000000000cd5f0 R_AARCH64_ABS64          .datadiv_decode11706101414295225912
00000000000cd3e8 R_AARCH64_ABS64          .datadiv_decode12335027288954124723
00000000000cd418 R_AARCH64_ABS64          .datadiv_decode18261546535841772752
00000000000cd450 R_AARCH64_ABS64          .datadiv_decode5616837089396308971
00000000000cd4f8 R_AARCH64_ABS64          .datadiv_decode13827763977763901235
00000000000cd458 R_AARCH64_ABS64          .datadiv_decode14151120317447827231
00000000000cd480 R_AARCH64_ABS64          .datadiv_decode11770512853690929982
00000000000cd5c0 R_AARCH64_ABS64          .datadiv_decode14401673864441325462
00000000000cd400 R_AARCH64_ABS64          .datadiv_decode8952246851265070369
00000000000cd478 R_AARCH64_ABS64          .datadiv_decode5074647643595886391
00000000000cd4a8 R_AARCH64_ABS64          .datadiv_decode11553230420239584337
00000000000cd4d8 R_AARCH64_ABS64          .datadiv_decode11784788714666544642
00000000000cd4f0 R_AARCH64_ABS64          .datadiv_decode15357274442415173716
00000000000cd510 R_AARCH64_ABS64          .datadiv_decode16313801769778548889
00000000000cd580 R_AARCH64_ABS64          .datadiv_decode18380577887196024106
00000000000cd3f8 R_AARCH64_ABS64          .datadiv_decode18328417529454547004
00000000000cd518 R_AARCH64_ABS64          .datadiv_decode12121706469311219939
00000000000cd540 R_AARCH64_ABS64          .datadiv_decode4193671268968804409
00000000000cd5c8 R_AARCH64_ABS64          .datadiv_decode6405721680354649260
00000000000cd5e8 R_AARCH64_ABS64          .datadiv_decode8316381480288167535
00000000000cd468 R_AARCH64_ABS64          .datadiv_decode12672127785521407892
00000000000cd530 R_AARCH64_ABS64          .datadiv_decode1095597082262436137
```

```
00000000000cd558 R_AARCH64_ABS64          .datadiv_decode11689284992262612148
00000000000cd588 R_AARCH64_ABS64          .datadiv_decode17940233878698364930
00000000000cd448 R_AARCH64_ABS64          .datadiv_decode1771790552069125206
00000000000cd4c8 R_AARCH64_ABS64          .datadiv_decode12950792008805936966
00000000000cd560 R_AARCH64_ABS64          .datadiv_decode6931257938975476985
00000000000cd5a0 R_AARCH64_ABS64          .datadiv_decode15620942718649555403
00000000000cd600 R_AARCH64_ABS64          .datadiv_decode8758840755024801160
00000000000cd408 R_AARCH64_ABS64          .datadiv_decode8010288038339893607
00000000000cd4b8 R_AARCH64_ABS64          .datadiv_decode2175705720332001599
00000000000cd590 R_AARCH64_ABS64          .datadiv_decode256915654516018196
00000000000cd5b0 R_AARCH64_ABS64          .datadiv_decode16710377636940783008
00000000000cd5d0 R_AARCH64_ABS64          .datadiv_decode1639262728706781308
00000000000cd5e0 R_AARCH64_ABS64          .datadiv_decode5533236249192328355
00000000000cd608 R_AARCH64_ABS64          .datadiv_decode2444497212690810360
00000000000cd420 R_AARCH64_ABS64          .datadiv_decode15481956303235929690
00000000000cd438 R_AARCH64_ABS64          .datadiv_decode17677705362567080649
00000000000cd440 R_AARCH64_ABS64          .datadiv_decode7614718738418908679
00000000000cd3f0 R_AARCH64_ABS64          .datadiv_decode17838636323198310142
00000000000cd430 R_AARCH64_ABS64          .datadiv_decode17330405497468958994
00000000000cd498 R_AARCH64_ABS64          .datadiv_decode1552205700074701063
00000000000cd4b0 R_AARCH64_ABS64          .datadiv_decode15147620753704794795
00000000000cd5d8 R_AARCH64_ABS64          .datadiv_decode5454406552017557296
00000000000cd8c0 R_AARCH64_JUMP_SLOT      __cxa_finalize
00000000000cd8c8 R_AARCH64_JUMP_SLOT      __cxa_atexit
00000000000cd8d0 R_AARCH64_JUMP_SLOT      __android_log_print
00000000000cd8d8 R_AARCH64_JUMP_SLOT      __stack_chk_fail
00000000000cd8e0 R_AARCH64_JUMP_SLOT      memset
00000000000cd8e8 R_AARCH64_JUMP_SLOT      strncpy
00000000000cd8f0 R_AARCH64_JUMP_SLOT      strncat
00000000000cd8f8 R_AARCH64_JUMP_SLOT      pthread_self
00000000000cd900 R_AARCH64_JUMP_SLOT      malloc
00000000000cd908 R_AARCH64_JUMP_SLOT      free
00000000000cd910 R_AARCH64_JUMP_SLOT      posix_memalign
00000000000cd918 R_AARCH64_JUMP_SLOT      vfprintf
00000000000cd920 R_AARCH64_JUMP_SLOT      fputc
00000000000cd928 R_AARCH64_JUMP_SLOT      vasprintf
00000000000cd930 R_AARCH64_JUMP_SLOT      android_set_abort_message
00000000000cd938 R_AARCH64_JUMP_SLOT      openlog
00000000000cd940 R_AARCH64_JUMP_SLOT      syslog
00000000000cd948 R_AARCH64_JUMP_SLOT      closelog
00000000000cd950 R_AARCH64_JUMP_SLOT      abort
00000000000cd958 R_AARCH64_JUMP_SLOT      strlen
00000000000cd960 R_AARCH64_JUMP_SLOT      realloc
00000000000cd968 R_AARCH64_JUMP_SLOT      memmove
00000000000cd970 R_AARCH64_JUMP_SLOT      __memmove_chk
00000000000cd978 R_AARCH64_JUMP_SLOT      __strlen_chk
00000000000cd980 R_AARCH64_JUMP_SLOT      memchr
00000000000cd988 R_AARCH64_JUMP_SLOT      __vsnprintf_chk
00000000000cd990 R_AARCH64_JUMP_SLOT      memcpy
00000000000cd998 R_AARCH64_JUMP_SLOT      pthread_mutex_lock
00000000000cd9a0 R_AARCH64_JUMP_SLOT      pthread_mutex_unlock
00000000000cd9a8 R_AARCH64_JUMP_SLOT      calloc
00000000000cd9b0 R_AARCH64_JUMP_SLOT      strcmp
00000000000cd9b8 R_AARCH64_JUMP_SLOT      pthread_getspecific
00000000000cd9c0 R_AARCH64_JUMP_SLOT      pthread_once
00000000000cd9c8 R_AARCH64_JUMP_SLOT      pthread_setspecific
```

```
00000000000cd9d0 R_AARCH64_JUMP_SLOT      pthread_key_delete
00000000000cd9d8 R_AARCH64_JUMP_SLOT      pthread_key_create
00000000000cd9e0 R_AARCH64_JUMP_SLOT      getauxval
00000000000cd9e8 R_AARCH64_JUMP_SLOT      __system_property_get
00000000000cd9f0 R_AARCH64_JUMP_SLOT      strncmp
00000000000cd9f8 R_AARCH64_JUMP_SLOT      fprintf
00000000000cda00 R_AARCH64_JUMP_SLOT      fflush
00000000000cda08 R_AARCH64_JUMP_SLOT      pthread_rwlock_wrlock
00000000000cda10 R_AARCH64_JUMP_SLOT      pthread_rwlock_unlock
00000000000cda18 R_AARCH64_JUMP_SLOT      dl_iterate_phdr
00000000000cda20 R_AARCH64_JUMP_SLOT      pthread_rwlock_rdlock
00000000000cda28 R_AARCH64_JUMP_SLOT      fwrite
```

objdump的语法help

```
➜  ~ objdump --help
OVERVIEW: llvm object file dumper

USAGE: objdump [options] <input object files>

OPTIONS:
  --adjust-vma=offset     Increase the displayed address by the specified offset
  --all-headers           Display all available header information, relocation entries
and the symbol table
  --arch-name=<value>     Target arch to disassemble for, see --version for available t
argets
  --archive-headers       Display archive header information
  -a                      Alias for --archive-headers
  -C                      Alias for --demangle
  --debug-vars-indent=<value>
                          Distance to indent the source-level variable display, relativ
e to the start of the disassembly
  --debug-vars=<value>    Print the locations (in registers or memory) of source-level
variables alongside disassembly. Supported formats: ascii, unicode (default)
  --demangle              Demangle symbol names
  --disassemble-all       Disassemble all sections found in the input files
  --disassemble-symbols=<value>
                          List of symbols to disassemble. Accept demangled names when -
-demangle is specified, otherwise accept mangled names
  --disassemble-zeroes    Do not skip blocks of zeroes when disassembling
  --disassembler-options=options
                          Pass target specific disassembler options
  --disassemble           Disassemble all executable sections found in the input files
  --dwarf=<value>         Dump the specified DWARF debug sections. The only supported v
alue is 'frames'
  --dynamic-reloc         Display the dynamic relocation entries in the file
  --dynamic-syms          Display the contents of the dynamic symbol table
  -D                      Alias for --disassemble-all
  -d                      Alias for --disassemble
  --fault-map-section     Display the content of the fault map section
  --file-headers          Display the contents of the overall file header
  --full-contents         Display the content of each section
  -f                      Alias for --file-headers
  --headers               Alias for --section-headers
  --help                  Display available options (--help-hidden for more)
  -h                      Alias for --section-headers
  -j <value>              Alias for --section
  --line-numbers          When disassembling, display source line numbers. Implies --di
sassemble
  -l                      Alias for --line-numbers
  --macho                 Use MachO specific object file parser
  --mattr=a1,+a2,-a3,...  Target specific attributes (--mattr=help for details)
  --mcpu=cpu-name         Target a specific cpu type (--mcpu=help for details)
  -M <value>              Alias for --disassembler-options=
  -m                      Alias for --macho
  --no-leading-addr       When disassembling, do not print leading addresses
  --no-print-imm-hex      Do not use hex format for immediate values (default)
  --no-show-raw-insn      When disassembling instructions, do not print the instruction
 bytes.
  --prefix-strip=prefix   Strip out initial directories from absolute paths. No effect
without --prefix
  --prefix=prefix         Add prefix to absolute paths
  --print-imm-hex         Use hex format for immediate values
  --private-headers       Display format specific file headers
  -p                      Alias for --private-headers
  --raw-clang-ast         Dump the raw binary contents of the clang AST section
  --reloc                 Display the relocation entries in the file
  -R                      Alias for --dynamic-reloc
  -r                      Alias for --reloc
  --section-headers       Display summaries of the headers for each section.
  --section=<value>       Operate on the specified sections only. With --macho dump seg
ment,section
  --show-lma              Display LMA column when dumping ELF section headers
  --source                When disassembling, display source interleaved with the disas
sembly. Implies --disassemble
  --start-address=address Set the start address for disassembling, printing relocations
 and printing symbols
  --stop-address=address  Set the stop address for disassembling, printing relocations
and printing symbols
  --symbol-description    Add symbol description for disassembly. This option is for XC
OFF files only.
  --symbolize-operands    Symbolize instruction operands when disassembling
  --syms                  Display the symbol table
  -S                      Alias for --source
  -s                      Alias for --full-contents
  --triple=<value>        Target triple to disassemble for, see --version for available
 targets
  -T                      Alias for --dynamic-syms
  -t                      Alias for --syms
  --unwind-info           Display unwind information
  -u                      Alias for --unwind-info
  --version               Display the version of this program
  -v                      Alias for --version
  --wide                  Ignored for compatibility with GNU objdump
  --x86-asm-syntax=att    Emit AT&T-style disassembly
  --x86-asm-syntax=intel  Emit Intel-style disassembly
  -x                      Alias for --all-headers
  -z                      Alias for --disassemble-zeroes

llvm-objdump MachO Specific Options:
  --arch=<value>         architecture(s) from a Mach-O file to dump
  --archive-member-offsets
                         Print the offset to each archive member for Mach-O archives (r
equires --macho and --archive-headers)
  --bind                 Display mach-o binding info
  --data-in-code         Print the data in code table for Mach-O objects (requires --ma
cho)
  --dis-symname=<value>  disassemble just this symbol's instructions (requires --macho)
  --dsym=<value>         Use .dSYM file for debug info
  --dyld_info            Print bind and rebase information used by dyld to resolve exte
rnal references in a final linked binary (requires --macho)
  --dylib-id             Print the shared library's id for the dylib Mach-O file (requi
res --macho)
  --dylibs-used          Print the shared libraries used for linked Mach-O files (requi
res --macho)
  --exports-trie         Display mach-o exported symbols
  --full-leading-addr    Print full leading address
  --function-starts      Print the function starts table for Mach-O objects (requires -
-macho)
  -g                     Print line information from debug info if available
  --indirect-symbols     Print indirect symbol table for Mach-O objects (requires --mac
ho)
  --info-plist           Print the info plist section as strings for Mach-O objects (re
quires --macho)
  --lazy-bind            Display mach-o lazy binding info
  --link-opt-hints       Print the linker optimization hints for Mach-O objects (requir
es --macho)
  --no-leading-headers   Print no leading headers
  --no-symbolic-operands do not symbolic operands when disassembling (requires --macho)
  --non-verbose          Print the info for Mach-O objects in non-verbose or numeric fo
rm (requires --macho)
  --objc-meta-data       Print the Objective-C runtime meta data for Mach-O files (requ
ires --macho)
  --private-header       Display only the first format specific file header
  --rebase               Display mach-o rebasing info
  --rpaths               Print the runtime search paths for the Mach-O file (requires -
-macho)
  --universal-headers    Print Mach-O universal headers (requires --macho)
  --weak-bind            Display mach-o weak binding info

Pass @FILE as argument to read options from FILE.
```

objdump的man手册

```
OBJDUMP(1)                 GNU Development Tools                OBJDUMP(1)

NAME         top
       objdump - display information from object files

SYNOPSIS         top
       objdump [-a|--archive-headers]
               [-b bfdname|--target=bfdname]
               [-C|--demangle[=style] ]
               [-d|--disassemble[=symbol]]
               [-D|--disassemble-all]
               [-z|--disassemble-zeroes]
               [-EB|-EL|--endian={big | little }]
               [-f|--file-headers]
               [-F|--file-offsets]
               [--file-start-context]
               [-g|--debugging]
               [-e|--debugging-tags]
               [-h|--section-headers|--headers]
               [-i|--info]
               [-j section|--section=section]
               [-l|--line-numbers]
               [-S|--source]
               [--source-comment[=text]]
               [-m machine|--architecture=machine]
               [-M options|--disassembler-options=options]
               [-p|--private-headers]
               [-P options|--private=options]
               [-r|--reloc]
               [-R|--dynamic-reloc]
               [-s|--full-contents]
               [-W[lLiaprmfFsoORtUuTgAck]|
                --dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=
frames,=frames-interp,=str,=str-offsets,=loc,=Ranges,=pubtypes,=trace_info,=trace_abbre
v,=trace_aranges,=gdb_index,=addr,=cu_index,=links]]
               [-WK|--dwarf=follow-links]
               [-WN|--dwarf=no-follow-links]
               [-wD|--dwarf=use-debuginfod]
               [-wE|--dwarf=do-not-use-debuginfod]
               [-L|--process-links]
               [--ctf=section]
               [--sframe=section]
               [-G|--stabs]
               [-t|--syms]
               [-T|--dynamic-syms]
               [-x|--all-headers]
               [-w|--wide]
               [--start-address=address]
               [--stop-address=address]
               [--no-addresses]
               [--prefix-addresses]
```

```
                 [--[no-]show-raw-insn]
                 [--adjust-vma=offset]
                 [--show-all-symbols]
                 [--dwarf-depth=n]
                 [--dwarf-start=n]
                 [--ctf-parent=section]
                 [--no-recurse-limit|--recurse-limit]
                 [--special-syms]
                 [--prefix=prefix]
                 [--prefix-strip=level]
                 [--insn-width=width]
                 [--visualize-jumps[=color|=extended-color|=off]
                 [--disassembler-color=[off|terminal|on|extended]
                 [-U method] [--unicode=method]
                 [-V|--version]
                 [-H|--help]
                 objfile...

DESCRIPTION         top
       objdump displays information about one or more object files.  The
       options control what particular information to display.  This
       information is mostly useful to programmers who are working on
       the compilation tools, as opposed to programmers who just want
       their program to compile and work.

       objfile... are the object files to be examined.  When you specify
       archives, objdump shows information on each of the member object
       files.

OPTIONS         top
       The long and short forms of options, shown here as alternatives,
       are equivalent.  At least one option from the list
       -a,-d,-D,-e,-f,-g,-G,-h,-H,-p,-P,-r,-R,-s,-S,-t,-T,-V,-x must be
       given.

       -a
       --archive-header
           If any of the objfile files are archives, display the archive
           header information (in a format similar to ls -l).  Besides
           the information you could list with ar tv, objdump -a shows
           the object file format of each archive member.

       --adjust-vma=offset
           When dumping information, first add offset to all the section
           addresses.  This is useful if the section addresses do not
           correspond to the symbol table, which can happen when putting
           sections at particular addresses when using a format which
           can not represent section addresses, such as a.out.

       -b bfdname
       --target=bfdname
           Specify that the object-code format for the object files is
           bfdname.  This option may not be necessary; objdump can
           automatically recognize many formats.

           For example,
                objdump -b oasys -m vax -h fu.o

        displays summary information from the section headers (-h) of
        fu.o, which is explicitly identified (-m) as a VAX object
        file in the format produced by Oasys compilers.  You can list
        the formats available with the -i option.

    -C
    --demangle[=style]
        Decode (demangle) low-level symbol names into user-level
        names.  Besides removing any initial underscore prepended by
        the system, this makes C++ function names readable.
        Different compilers have different mangling styles. The
        optional demangling style argument can be used to choose an
        appropriate demangling style for your compiler.

    --recurse-limit
    --no-recurse-limit
    --recursion-limit
    --no-recursion-limit
        Enables or disables a limit on the amount of recursion
        performed whilst demangling strings.  Since the name mangling
        formats allow for an infinite level of recursion it is
        possible to create strings whose decoding will exhaust the
        amount of stack space available on the host machine,
        triggering a memory fault.  The limit tries to prevent this
        from happening by restricting recursion to 2048 levels of
        nesting.

        The default is for this limit to be enabled, but disabling it
        may be necessary in order to demangle truly complicated
        names.  Note however that if the recursion limit is disabled
        then stack exhaustion is possible and any bug reports about
        such an event will be rejected.

    -g
    --debugging
        Display debugging information.  This attempts to parse STABS
        debugging format information stored in the file and print it
        out using a C like syntax.  If no STABS debugging was found
        this option falls back on the -W option to print any DWARF
        information in the file.

    -e
    --debugging-tags
        Like -g, but the information is generated in a format
        compatible with ctags tool.

    -d
    --disassemble
    --disassemble=symbol
        Display the assembler mnemonics for the machine instructions
        from the input file.  This option only disassembles those
        sections which are expected to contain instructions.  If the
        optional symbol argument is given, then display the assembler
```

```
           mnemonics starting at symbol.  If symbol is a function name
           then disassembly will stop at the end of the function,
           otherwise it will stop when the next symbol is encountered.
           If there are no matches for symbol then nothing will be
           displayed.

           Note if the --dwarf=follow-links option is enabled then any
           symbol tables in linked debug info files will be read in and
           used when disassembling.

       -D
       --disassemble-all
           Like -d, but disassemble the contents of all sections, not
           just those expected to contain instructions.

           This option also has a subtle effect on the disassembly of
           instructions in code sections.  When option -d is in effect
           objdump will assume that any symbols present in a code
           section occur on the boundary between instructions and it
           will refuse to disassemble across such a boundary.  When
           option -D is in effect however this assumption is supressed.
           This means that it is possible for the output of -d and -D to
           differ if, for example, data is stored in code sections.

           If the target is an ARM architecture this switch also has the
           effect of forcing the disassembler to decode pieces of data
           found in code sections as if they were instructions.

           Note if the --dwarf=follow-links option is enabled then any
           symbol tables in linked debug info files will be read in and
           used when disassembling.

       --no-addresses
           When disassembling, don't print addresses on each line or for
           symbols and relocation offsets.  In combination with
           --no-show-raw-insn this may be useful for comparing compiler
           output.

       --prefix-addresses
           When disassembling, print the complete address on each line.
           This is the older disassembly format.

       -EB
       -EL
       --endian={big|little}
           Specify the endianness of the object files.  This only
           affects disassembly.  This can be useful when disassembling a
           file format which does not describe endianness information,
           such as S-records.

       -f
       --file-headers
           Display summary information from the overall header of each
           of the objfile files.

       -F
       --file-offsets
           When disassembling sections, whenever a symbol is displayed,
           also display the file offset of the region of data that is
           about to be dumped.  If zeroes are being skipped, then when
           disassembly resumes, tell the user how many zeroes were
           skipped and the file offset of the location from where the
           disassembly resumes.  When dumping sections, display the file
           offset of the location from where the dump starts.

       --file-start-context
           Specify that when displaying interlisted source
           code/disassembly (assumes -S) from a file that has not yet
           been displayed, extend the context to the start of the file.

       -h
       --section-headers
       --headers
           Display summary information from the section headers of the
           object file.

           File segments may be relocated to nonstandard addresses, for
           example by using the -Ttext, -Tdata, or -Tbss options to ld.
           However, some object file formats, such as a.out, do not
           store the starting address of the file segments.  In those
           situations, although ld relocates the sections correctly,
           using objdump -h to list the file section headers cannot show
           the correct addresses.  Instead, it shows the usual
           addresses, which are implicit for the target.

           Note, in some cases it is possible for a section to have both
           the READONLY and the NOREAD attributes set.  In such cases
           the NOREAD attribute takes precedence, but objdump will
           report both since the exact setting of the flag bits might be
           important.

       -H
       --help
           Print a summary of the options to objdump and exit.

       -i
       --info
           Display a list showing all architectures and object formats
           available for specification with -b or -m.

       -j name
       --section=name
           Display information only for section name.

       -L
       --process-links
           Display the contents of non-debug sections found in separate
           debuginfo files that are linked to the main file.  This
           option automatically implies the -WK option, and only
           sections requested by other command line options will be
           displayed.
       -l
       --line-numbers
           Label the display (using debugging information) with the
           filename and source line numbers corresponding to the object
           code or relocs shown.  Only useful with -d, -D, or -r.

       -m machine
       --architecture=machine
           Specify the architecture to use when disassembling object
           files.  This can be useful when disassembling object files
           which do not describe architecture information, such as
           S-records.  You can list the available architectures with the
           -i option.

           For most architectures it is possible to supply an
           architecture name and a machine name, separated by a colon.
           For example foo:bar would refer to the bar machine type in
           the foo architecture.  This can be helpful if objdump has
           been configured to support multiple architectures.

           If the target is an ARM architecture then this switch has an
           additional effect.  It restricts the disassembly to only
           those instructions supported by the architecture specified by
           machine.  If it is necessary to use this switch because the
           input file does not contain any architecture information, but
           it is also desired to disassemble all the instructions use
           -marm.

       -M options
       --disassembler-options=options
           Pass target specific information to the disassembler.  Only
           supported on some targets.  If it is necessary to specify
           more than one disassembler option then multiple -M options
           can be used or can be placed together into a comma separated
           list.

           For ARC, dsp controls the printing of DSP instructions, spfp
           selects the printing of FPX single precision FP instructions,
           dpfp selects the printing of FPX double precision FP
           instructions, quarkse_em selects the printing of special
           QuarkSE-EM instructions, fpuda selects the printing of double
           precision assist instructions, fpus selects the printing of
           FPU single precision FP instructions, while fpud selects the
           printing of FPU double precision FP instructions.
           Additionally, one can choose to have all the immediates
           printed in hexadecimal using hex.  By default, the short
           immediates are printed using the decimal representation,
           while the long immediate values are printed as hexadecimal.

           cpu=... allows one to enforce a particular ISA when
           disassembling instructions, overriding the -m value or
           whatever is in the ELF file.  This might be useful to select
           ARC EM or HS ISA, because architecture is same for those and
           disassembler relies on private ELF header data to decide if
           code is for EM or HS.  This option might be specified
           multiple times - only the latest value will be used.  Valid
```

```
             values are same as for the assembler -mcpu=... option.

             If the target is an ARM architecture then this switch can be
             used to select which register name set is used during
             disassembler.  Specifying -M reg-names-std (the default) will
             select the register names as used in ARM's instruction set
             documentation, but with register 13 called 'sp', register 14
             called 'lr' and register 15 called 'pc'.  Specifying -M reg-
             names-apcs will select the name set used by the ARM Procedure
             Call Standard, whilst specifying -M reg-names-raw will just
             use r followed by the register number.

             There are also two variants on the APCS register naming
             scheme enabled by -M reg-names-atpcs and -M reg-names-
             special-atpcs which use the ARM/Thumb Procedure Call Standard
             naming conventions.  (Either with the normal register names
             or the special register names).

             This option can also be used for ARM architectures to force
             the disassembler to interpret all instructions as Thumb
             instructions by using the switch
             --disassembler-options=force-thumb.  This can be useful when
             attempting to disassemble thumb code produced by other
             compilers.

             For AArch64 targets this switch can be used to set whether
             instructions are disassembled as the most general instruction
             using the -M no-aliases option or whether instruction notes
             should be generated as comments in the disasssembly using -M
             notes.

             For the x86, some of the options duplicate functions of the
             -m switch, but allow finer grained control.

             "x86-64"
             "i386"
             "i8086"
                 Select disassembly for the given architecture.

             "intel"
             "att"
                 Select between intel syntax mode and AT&T syntax mode.

             "amd64"
             "intel64"
                 Select between AMD64 ISA and Intel64 ISA.

             "intel-mnemonic"
             "att-mnemonic"
                 Select between intel mnemonic mode and AT&T mnemonic
                 mode.  Note: "intel-mnemonic" implies "intel" and
                 "att-mnemonic" implies "att".

             "addr64"
             "addr32"
             "addr16"
             "data32"
             "data16"
                 Specify the default address size and operand size.  These
                 five options will be overridden if "x86-64", "i386" or
                 "i8086" appear later in the option string.

             "suffix"
                 When in AT&T mode and also for a limited set of
                 instructions when in Intel mode, instructs the
                 disassembler to print a mnemonic suffix even when the
                 suffix could be inferred by the operands or, for certain
                 instructions, the execution mode's defaults.

             For PowerPC, the -M argument raw selects disasssembly of
             hardware insns rather than aliases.  For example, you will
             see "rlwinm" rather than "clrlwi", and "addi" rather than
             "li".  All of the -m arguments for gas that select a CPU are
             supported.  These are: 403, 405, 440, 464, 476, 601, 603,
             604, 620, 7400, 7410, 7450, 7455, 750cl, 821, 850, 860, a2,
             booke, booke32, cell, com, e200z2, e200z4, e300, e500,
             e500mc, e500mc64, e500x2, e5500, e6500, efs, power4, power5,
             power6, power7, power8, power9, power10, ppc, ppc32, ppc64,
             ppc64bridge, ppcps, pwr, pwr2, pwr4, pwr5, pwr5x, pwr6, pwr7,
             pwr8, pwr9, pwr10, pwrx, titan, vle, and future.  32 and 64
             modify the default or a prior CPU selection, disabling and
             enabling 64-bit insns respectively.  In addition, altivec,
             any, lsp, htm, vsx, spe and  spe2 add capabilities to a
             previous or later CPU selection.  any will disassemble any
             opcode known to binutils, but in cases where an opcode has
             two different meanings or different arguments, you may not
             see the disassembly you expect.  If you disassemble without
             giving a CPU selection, a default will be chosen from
             information gleaned by BFD from the object files headers, but
             the result again may not be as you expect.

             For MIPS, this option controls the printing of instruction
             mnemonic names and register names in disassembled
             instructions.  Multiple selections from the following may be
             specified as a comma separated string, and invalid options
             are ignored:

             "no-aliases"
                 Print the 'raw' instruction mnemonic instead of some
                 pseudo instruction mnemonic.  I.e., print 'daddu' or 'or'
                 instead of 'move', 'sll' instead of 'nop', etc.

             "msa"
                 Disassemble MSA instructions.

             "virt"
                 Disassemble the virtualization ASE instructions.

             "xpa"
                 Disassemble the eXtended Physical Address (XPA) ASE
                 instructions.
```

```
           "gpr-names=ABI"
               Print GPR (general-purpose register) names as appropriate
               for the specified ABI.  By default, GPR names are
               selected according to the ABI of the binary being
               disassembled.

           "fpr-names=ABI"
               Print FPR (floating-point register) names as appropriate
               for the specified ABI.  By default, FPR numbers are
               printed rather than names.

           "cp0-names=ARCH"
               Print CP0 (system control coprocessor; coprocessor 0)
               register names as appropriate for the CPU or architecture
               specified by ARCH.  By default, CP0 register names are
               selected according to the architecture and CPU of the
               binary being disassembled.

           "hwr-names=ARCH"
               Print HWR (hardware register, used by the "rdhwr"
               instruction) names as appropriate for the CPU or
               architecture specified by ARCH.  By default, HWR names
               are selected according to the architecture and CPU of the
               binary being disassembled.

           "reg-names=ABI"
               Print GPR and FPR names as appropriate for the selected
               ABI.

           "reg-names=ARCH"
               Print CPU-specific register names (CP0 register and HWR
               names) as appropriate for the selected CPU or
               architecture.

           For any of the options listed above, ABI or ARCH may be
           specified as numeric to have numbers printed rather than
           names, for the selected types of registers.  You can list the
           available values of ABI and ARCH using the --help option.

           For VAX, you can specify function entry addresses with -M
           entry:0xf00ba.  You can use this multiple times to properly
           disassemble VAX binary files that don't contain symbol tables
           (like ROM dumps).  In these cases, the function entry mask
           would otherwise be decoded as VAX instructions, which would
           probably lead the rest of the function being wrongly
           disassembled.

       -p
       --private-headers
           Print information that is specific to the object file format.
           The exact information printed depends upon the object file
           format.  For some object file formats, no additional
           information is printed.

       -P options
       --private=options
```

```
           Print information that is specific to the object file format.
           The argument options is a comma separated list that depends
           on the format (the lists of options is displayed with the
           help).

           For XCOFF, the available options are:

           "header"
           "aout"
           "sections"
           "syms"
           "relocs"
           "lineno,"
           "loader"
           "except"
           "typchk"
           "traceback"
           "toc"
           "ldinfo"

           Not all object formats support this option.  In particular
           the ELF format does not use it.

       -r
       --reloc
           Print the relocation entries of the file.  If used with -d or
           -D, the relocations are printed interspersed with the
           disassembly.

       -R
       --dynamic-reloc
           Print the dynamic relocation entries of the file.  This is
           only meaningful for dynamic objects, such as certain types of
           shared libraries.  As for -r, if used with -d or -D, the
           relocations are printed interspersed with the disassembly.

       -s
       --full-contents
           Display the full contents of any sections requested.  By
           default all non-empty sections are displayed.

       -S
       --source
           Display source code intermixed with disassembly, if possible.
           Implies -d.

       --show-all-symbols
           When disassembling, show all the symbols that match a given
           address, not just the first one.

       --source-comment[=txt]
           Like the -S option, but all source code lines are displayed
           with a prefix of txt.  Typically txt will be a comment string
           which can be used to distinguish the assembler code from the
           source code.  If txt is not provided then a default string of
           "# " (hash followed by a space), will be used.
       --prefix=prefix
           Specify prefix to add to the absolute paths when used with
           -S.

       --prefix-strip=level
           Indicate how many initial directory names to strip off the
           hardwired absolute paths. It has no effect without
           --prefix=prefix.

       --show-raw-insn
           When disassembling instructions, print the instruction in hex
           as well as in symbolic form.  This is the default except when
           --prefix-addresses is used.

       --no-show-raw-insn
           When disassembling instructions, do not print the instruction
           bytes.  This is the default when --prefix-addresses is used.

       --insn-width=width
           Display width bytes on a single line when disassembling
           instructions.

       --visualize-jumps[=color|=extended-color|=off]
           Visualize jumps that stay inside a function by drawing ASCII
           art between the start and target addresses.  The optional
           =color argument adds color to the output using simple
           terminal colors.  Alternatively the =extended-color argument
           will add color using 8bit colors, but these might not work on
           all terminals.

           If it is necessary to disable the visualize-jumps option
           after it has previously been enabled then use
           visualize-jumps=off.

       --disassembler-color=off
       --disassembler-color=terminal
       --disassembler-color=on|color|colour
       --disassembler-color=extened|extended-color|extened-colour
           Enables or disables the use of colored syntax highlighting in
           disassembly output.  The default behaviour is determined via
           a configure time option.  Note, not all architectures support
           colored syntax highlighting, and depending upon the terminal
           used, colored output may not actually be legible.

           The on argument adds colors using simple terminal colors.

           The terminal argument does the same, but only if the output
           device is a terminal.

           The extended-color argument is similar to the on argument,
           but it uses 8-bit colors.  These may not work on all
           terminals.

           The off argument disables colored disassembly.
       -W[lLiaprmfFsoORtUuTgAckK]
       --dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=f
rames-interp,=str,=str-offsets,=loc,=Ranges,=pubtypes,=trace_info,=trace_abbrev,=trace_
aranges,=gdb_index,=addr,=cu_index,=links,=follow-links]
           Displays the contents of the DWARF debug sections in the
           file, if any are present.  Compressed debug sections are
           automatically decompressed (temporarily) before they are
           displayed.  If one or more of the optional letters or words
           follows the switch then only those type(s) of data will be
           dumped.  The letters and words refer to the following
           information:

           "a"
           "=abbrev"
               Displays the contents of the .debug_abbrev section.

           "A"
           "=addr"
               Displays the contents of the .debug_addr section.

           "c"
           "=cu_index"
               Displays the contents of the .debug_cu_index and/or
               .debug_tu_index sections.

           "f"
           "=frames"
               Display the raw contents of a .debug_frame section.

           "F"
           "=frames-interp"
               Display the interpreted contents of a .debug_frame
               section.

           "g"
           "=gdb_index"
               Displays the contents of the .gdb_index and/or
               .debug_names sections.

           "i"
           "=info"
               Displays the contents of the .debug_info section.  Note:
               the output from this option can also be restricted by the
               use of the --dwarf-depth and --dwarf-start options.

           "k"
           "=links"
               Displays the contents of the .gnu_debuglink,
               .gnu_debugaltlink and .debug_sup sections, if any of them
               are present.  Also displays any links to separate dwarf
               object files (dwo), if they are specified by the
               DW_AT_GNU_dwo_name or DW_AT_dwo_name attributes in the
               .debug_info section.

           "K"
           "=follow-links"
```

```
            Display the contents of any selected debug sections that
            are found in linked, separate debug info file(s).  This
            can result in multiple versions of the same debug section
            being displayed if it exists in more than one file.

            In addition, when displaying DWARF attributes, if a form
            is found that references the separate debug info file,
            then the referenced contents will also be displayed.

            Note - in some distributions this option is enabled by
            default.  It can be disabled via the N debug option.  The
            default can be chosen when configuring the binutils via
            the --enable-follow-debug-links yes or
            --enable-follow-debug-links no options.  If these are not
            used then the default is to enable the following of debug
            links.

            Note - if support for the debuginfod protocol was enabled
            when the binutils were built then this option will also
            include an attempt to contact any debuginfod servers
            mentioned in the DEBUGINFOD_URLS environment variable.
            This could take some time to resolve.  This behaviour can
            be disabled via the =do-not-use-debuginfod debug option.

        "N"
        "=no-follow-links"
            Disables the following of links to separate debug info
            files.

        "D"
        "=use-debuginfod"
            Enables contacting debuginfod servers if there is a need
            to follow debug links.  This is the default behaviour.

        "E"
        "=do-not-use-debuginfod"
            Disables contacting debuginfod servers when there is a
            need to follow debug links.

        "l"
        "=rawline"
            Displays the contents of the .debug_line section in a raw
            format.

        "L"
        "=decodedline"
            Displays the interpreted contents of the .debug_line
            section.

        "m"
        "=macro"
            Displays the contents of the .debug_macro and/or
            .debug_macinfo sections.

        "o"
        "=loc"
```

```
              Displays the contents of the .debug_loc and/or
              .debug_loclists sections.

          "O"
          "=str-offsets"
              Displays the contents of the .debug_str_offsets section.

          "p"
          "=pubnames"
              Displays the contents of the .debug_pubnames and/or
              .debug_gnu_pubnames sections.

          "r"
          "=aranges"
              Displays the contents of the .debug_aranges section.

          "R"
          "=Ranges"
              Displays the contents of the .debug_ranges and/or
              .debug_rnglists sections.

          "s"
          "=str"
              Displays the contents of the .debug_str, .debug_line_str
              and/or .debug_str_offsets sections.

          "t"
          "=pubtype"
              Displays the contents of the .debug_pubtypes and/or
              .debug_gnu_pubtypes sections.

          "T"
          "=trace_aranges"
              Displays the contents of the .trace_aranges section.

          "u"
          "=trace_abbrev"
              Displays the contents of the .trace_abbrev section.

          "U"
          "=trace_info"
              Displays the contents of the .trace_info section.

          Note: displaying the contents of .debug_static_funcs,
          .debug_static_vars and debug_weaknames sections is not
          currently supported.

      --dwarf-depth=n
          Limit the dump of the ".debug_info" section to n children.
          This is only useful with --debug-dump=info.  The default is
          to print all DIEs; the special value 0 for n will also have
          this effect.

          With a non-zero value for n, DIEs at or deeper than n levels
          will not be printed.  The range for n is zero-based.
```

```
         --dwarf-start=n
             Print only DIEs beginning with the DIE numbered n.  This is
             only useful with --debug-dump=info.

             If specified, this option will suppress printing of any
             header information and all DIEs before the DIE numbered n.
             Only siblings and children of the specified DIE will be
             printed.

             This can be used in conjunction with --dwarf-depth.

         --dwarf-check
             Enable additional checks for consistency of Dwarf
             information.

         --ctf[=section]
             Display the contents of the specified CTF section.  CTF
             sections themselves contain many subsections, all of which
             are displayed in order.

             By default, display the name of the section named .ctf, which
             is the name emitted by ld.

         --ctf-parent=member
             If the CTF section contains ambiguously-defined types, it
             will consist of an archive of many CTF dictionaries, all
             inheriting from one dictionary containing unambiguous types.
             This member is by default named .ctf, like the section
             containing it, but it is possible to change this name using
             the "ctf_link_set_memb_name_changer" function at link time.
             When looking at CTF archives that have been created by a
             linker that uses the name changer to rename the parent
             archive member, --ctf-parent can be used to specify the name
             used for the parent.

         --sframe[=section]
             Display the contents of the specified SFrame section.

             By default, display the name of the section named .sframe,
             which is the name emitted by ld.

         -G
         --stabs
             Display the full contents of any sections requested.  Display
             the contents of the .stab and .stab.index and .stab.excl
             sections from an ELF file.  This is only useful on systems
             (such as Solaris 2.0) in which ".stab" debugging symbol-table
             entries are carried in an ELF section.  In most other file
             formats, debugging symbol-table entries are interleaved with
             linkage symbols, and are visible in the --syms output.

         --start-address=address
             Start displaying data at the specified address.  This affects
             the output of the -d, -r and -s options.

         --stop-address=address
```

```
            Stop displaying data at the specified address.  This affects
            the output of the -d, -r and -s options.

        -t
        --syms
            Print the symbol table entries of the file.  This is similar
            to the information provided by the nm program, although the
            display format is different.  The format of the output
            depends upon the format of the file being dumped, but there
            are two main types.  One looks like this:

                    [  4](sec  3)(fl 0x00)(ty   0)(scl   3) (nx 1) 0x00000000 .bss
                    [  6](sec  1)(fl 0x00)(ty   0)(scl   2) (nx 0) 0x00000000 fred

            where the number inside the square brackets is the number of
            the entry in the symbol table, the sec number is the section
            number, the fl value are the symbol's flag bits, the ty
            number is the symbol's type, the scl number is the symbol's
            storage class and the nx value is the number of auxiliary
            entries associated with the symbol.  The last two fields are
            the symbol's value and its name.

            The other common output format, usually seen with ELF based
            files, looks like this:

                    00000000 l    d  .bss   00000000 .bss
                    00000000 g       .text  00000000 fred

            Here the first number is the symbol's value (sometimes
            referred to as its address).  The next field is actually a
            set of characters and spaces indicating the flag bits that
            are set on the symbol.  These characters are described below.
            Next is the section with which the symbol is associated or
            *ABS* if the section is absolute (ie not connected with any
            section), or *UND* if the section is referenced in the file
            being dumped, but not defined there.

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

            The flag characters are divided into 7 groups as follows:

            "l"
            "g"
            "u"
            "!" The symbol is a local (l), global (g), unique global (u),
                neither global nor local (a space) or both global and
                local (!).  A symbol can be neither local or global for a
                variety of reasons, e.g., because it is used for
                debugging, but it is probably an indication of a bug if
                it is ever both local and global.  Unique global symbols
                are a GNU extension to the standard set of ELF symbol
                bindings.  For such a symbol the dynamic linker will make
                sure that in the entire process there is just one symbol
                with this name and type in use.
```

```
             "w" The symbol is weak (w) or strong (a space).

             "C" The symbol denotes a constructor (C) or an ordinary
                 symbol (a space).

             "W" The symbol is a warning (W) or a normal symbol (a space).
                 A warning symbol's name is a message to be displayed if
                 the symbol following the warning symbol is ever
                 referenced.

             "I"
             "i" The symbol is an indirect reference to another symbol
                 (I), a function to be evaluated during reloc processing
                 (i) or a normal symbol (a space).

             "d"
             "D" The symbol is a debugging symbol (d) or a dynamic symbol
                 (D) or a normal symbol (a space).

             "F"
             "f"
             "O" The symbol is the name of a function (F) or a file (f) or
                 an object (O) or just a normal symbol (a space).

         -T
         --dynamic-syms
             Print the dynamic symbol table entries of the file.  This is
             only meaningful for dynamic objects, such as certain types of
             shared libraries.  This is similar to the information
             provided by the nm program when given the -D (--dynamic)
             option.

             The output format is similar to that produced by the --syms
             option, except that an extra field is inserted before the
             symbol's name, giving the version information associated with
             the symbol.  If the version is the default version to be used
             when resolving unversioned references to the symbol then it's
             displayed as is, otherwise it's put into parentheses.

         --special-syms
             When displaying symbols include those which the target
             considers to be special in some way and which would not
             normally be of interest to the user.

         -U [d|i|l|e|x|h]
         --unicode=[default|invalid|locale|escape|hex|highlight]
             Controls the display of UTF-8 encoded multibyte characters in
             strings.  The default (--unicode=default) is to give them no
             special treatment.  The --unicode=locale option displays the
             sequence in the current locale, which may or may not support
             them.  The options --unicode=hex and --unicode=invalid
             display them as hex byte sequences enclosed by either angle
             brackets or curly braces.

             The --unicode=escape option displays them as escape sequences
```

```
           (\uxxxx) and the --unicode=highlight option displays them as
           escape sequences highlighted in red (if supported by the
           output device).  The colouring is intended to draw attention
           to the presence of unicode sequences where they might not be
           expected.

       -V
       --version
           Print the version number of objdump and exit.

       -x
       --all-headers
           Display all available header information, including the
           symbol table and relocation entries.  Using -x is equivalent
           to specifying all of -a -f -h -p -r -t.

       -w
       --wide
           Format some lines for output devices that have more than 80
           columns.  Also do not truncate symbol names when they are
           displayed.

       -z
       --disassemble-zeroes
           Normally the disassembly output will skip blocks of zeroes.
           This option directs the disassembler to disassemble those
           blocks, just like any other data.

       @file
           Read command-line options from file.  The options read are
           inserted in place of the original @file option.  If file does
           not exist, or cannot be read, then the option will be treated
           literally, and not removed.

           Options in file are separated by whitespace.  A whitespace
           character may be included in an option by surrounding the
           entire option in either single or double quotes.  Any
           character (including a backslash) may be included by
           prefixing the character to be included with a backslash.  The
           file may itself contain additional @file options; any such
           options will be processed recursively.

SEE ALSO         top
       nm(1), readelf(1), and the Info entries for binutils.

COPYRIGHT         top
       Copyright (c) 1991-2023 Free Software Foundation, Inc.

       Permission is granted to copy, distribute and/or modify this
       document under the terms of the GNU Free Documentation License,
       Version 1.3 or any later version published by the Free Software
       Foundation; with no Invariant Sections, with no Front-Cover
       Texts, and with no Back-Cover Texts.  A copy of the license is
       included in the section entitled "GNU Free Documentation
       License".
```

```
COLOPHON         top
       This page is part of the binutils (a collection of tools for
       working with executable binaries) project.  Information about the
       project can be found at ⟨http://www.gnu.org/software/binutils/⟩.
       If you have a bug report for this manual page, see
       ⟨http://sourceware.org/bugzilla/enter_bug.cgi?product binutils⟩.
       This page was obtained from the tarball binutils-2.40.tar.gz
       fetched from ⟨https://ftp.gnu.org/gnu/binutils/⟩ on 2023-06-23.
       If you discover any rendering problems in this HTML version of
       the page, or you believe there is a better or more up-to-date
       source for the page, or you have corrections or improvements to
       the information in this COLOPHON (which is not part of the
       original manual page), send a mail to man-pages@man7.org

binutils-2.40.00               2023-06-23                     OBJDUMP(1)
```

rabin2

- rabin2
  - 是什么：radare2的其中一个命令行工具
  - 作用：查看二进制信息
  - 文档
    - Rabin2 - The Official Radare2 Book
    - rabin2 - r2wiki
  - 下载
    - radare2
      - Releases · radareorg/radare2

rabin2用法

概述：

- `-I` ：binary info

  ```
  rabin2 -I binaryFile
  ```

- `-i` ：imports

  ```
  rabin2 -i binaryFile
  ```

- `-E` ：exports

  ```
  rabin2 -E binaryFile
  ```

- `-l` ：linked libraries

  ```
  rabin2 -l binaryFile
  ```

- `-z` ：strings (from data section)

  ```
  rabin2 -z binaryFile
  ```

- `-s` ：symbols

  ```
  rabin2 -s binaryFile
  ```

- `-S` ：sections

  ```
  rabin2 -S binaryFile
  ```

- `-SS` ：segments

  ```
  rabin2 -SS binaryFile
  ```

rabin2用法举例

`-I`：binary info

```
➜  arm64-v8a rabin2 -I libtacker.so
arch     arm
baddr    0x0
binsz    848338
bintype  elf
bits     64
canary   true
class    ELF64
compiler Linker: LLD 14.0.1 clang version 14.0.0
crypto   false
endian   little
havecode true
laddr    0x0
lang     c
linenum  false
lsyms    false
machine  ARM aarch64
nx       true
os       android
pic      true
relocs   false
relro    full
rpath    NONE
sanitize false
static   false
stripped true
subsys   android
va       true
```

`-i`：imports

```
➜  arm64-v8a rabin2 -i libtacker.so
[Imports]
nth vaddr      bind   type lib name
―――――――――――――――――――――――――――――――――――
1   0x000c9240 GLOBAL FUNC     __cxa_finalize
2   0x000c9250 GLOBAL FUNC     __cxa_atexit
3   0x000c9260 GLOBAL FUNC     __android_log_print
4   0x000c9270 GLOBAL FUNC     __stack_chk_fail
5   0x000c9280 GLOBAL FUNC     memset
6   0x000c9290 GLOBAL FUNC     strncpy
7   0x000c92a0 GLOBAL FUNC     strncat
8   0x000c92b0 GLOBAL FUNC     pthread_self
9   0x000c92c0 GLOBAL FUNC     malloc
10  0x000c92d0 GLOBAL FUNC     free
11  0x000c92e0 GLOBAL FUNC     posix_memalign
```

```
12  ---------- GLOBAL OBJ      __sF
13  0x000c92f0 GLOBAL FUNC     vfprintf
14  0x000c9300 GLOBAL FUNC     fputc
15  0x000c9310 GLOBAL FUNC     vasprintf
16  0x000c9320 GLOBAL FUNC     android_set_abort_message
17  0x000c9330 GLOBAL FUNC     openlog
18  0x000c9340 GLOBAL FUNC     syslog
19  0x000c9350 GLOBAL FUNC     closelog
20  0x000c9360 GLOBAL FUNC     abort
21  0x000c9370 GLOBAL FUNC     strlen
22  0x000c9380 GLOBAL FUNC     realloc
23  0x000c9390 GLOBAL FUNC     memmove
24  0x000c93a0 GLOBAL FUNC     __memmove_chk
25  0x000c93b0 GLOBAL FUNC     __strlen_chk
26  0x000c93c0 GLOBAL FUNC     memchr
27  0x000c93d0 GLOBAL FUNC     __vsnprintf_chk
28  0x000c93e0 GLOBAL FUNC     memcpy
29  0x000c93f0 GLOBAL FUNC     pthread_mutex_lock
30  0x000c9400 GLOBAL FUNC     pthread_mutex_unlock
31  0x000c9410 GLOBAL FUNC     calloc
32  0x000c9420 GLOBAL FUNC     strcmp
33  0x000c9430 GLOBAL FUNC     pthread_getspecific
34  0x000c9440 GLOBAL FUNC     pthread_once
35  0x000c9450 GLOBAL FUNC     pthread_setspecific
36  0x000c9460 GLOBAL FUNC     pthread_key_delete
37  0x000c9470 GLOBAL FUNC     pthread_key_create
38  0x000c9480 GLOBAL FUNC     getauxval
39  0x000c9490 GLOBAL FUNC     __system_property_get
40  0x000c94a0 GLOBAL FUNC     strncmp
41  0x000c94b0 GLOBAL FUNC     fprintf
42  0x000c94c0 GLOBAL FUNC     fflush
43  0x000c94d0 GLOBAL FUNC     pthread_rwlock_wrlock
44  0x000c94e0 GLOBAL FUNC     pthread_rwlock_unlock
45  0x000c94f0 GLOBAL FUNC     dl_iterate_phdr
46  0x000c9500 GLOBAL FUNC     pthread_rwlock_rdlock
47  0x000c9510 GLOBAL FUNC     fwrite
```

-E : exports

```
➜  arm64-v8a rabin2 -E libtacker.so
[Exports]
nth paddr      vaddr      bind   type size  lib name                              dem
angled
―――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――
――――――
48  0x00044ce8 0x00044ce8 GLOBAL FUNC 6608      .datadiv_decode16117807209816376729
49  0x00078a04 0x00078a04 GLOBAL FUNC 2696      .datadiv_decode9901940071257331957
50  0x000a8a58 0x000a8a58 GLOBAL FUNC 3892      .datadiv_decode14716202181486223822
...
70  0x0008e650 0x0008e650 GLOBAL FUNC 3772      .datadiv_decode13214095259256631718
71  0x000381fc 0x000381fc GLOBAL FUNC 3696      .datadiv_decode3631146530348700705
72  0x0006f220 0x0006f220 GLOBAL FUNC 3892      .datadiv_decode8050698040297613930
73  0x000a8884 0x000a8884 GLOBAL FUNC 4         .datadiv_decode11706101414295225912
74  0x000aa438 0x000aa438 GLOBAL FUNC 1436      JNI_OnLoad
```

```
75  0x00026d98 0x00026d98 GLOBAL FUNC 18656     .datadiv_decode12335027288954124723
76  0x00033a2c 0x00033a2c GLOBAL FUNC 11972     .datadiv_decode18261546535841772752
77  0x0003c8dc 0x0003c8dc GLOBAL FUNC 8072      .datadiv_decode5616837089396308971
...
115 0x0005c790 0x0005c790 GLOBAL FUNC 4         .datadiv_decode15522057000747010 63
116 0x0005ed50 0x0005ed50 GLOBAL FUNC 6304      .datadiv_decode15147620753704794795
117 0x000a5e74 0x000a5e74 GLOBAL FUNC 4         .datadiv_decode5454406552017557296
```

`-l`：linked libraries

```
➜  arm64-v8a rabin2 -l libtacker.so
[Linked libraries]
liblog.so
libm.so
libdl.so
libc.so

4 libraries
```

`-z`：strings (from data section)

```
➜  arm64-v8a rabin2 -z libtacker.so
[Strings]
nth paddr      vaddr      len size section type    string
―――――――――――――――――――――――――――――――――――――――――――――――――――――――――
0   0x0000c6f9 0x0000c6f9 26  27   .rodata ascii   covariant return thunk to
1   0x0000c719 0x0000c719 10  11   .rodata ascii   operator^=
2   0x0000c724 0x0000c724 10  11   .rodata ascii   operator<=
3   0x0000c72f 0x0000c72f 24  25   .rodata ascii   unknown pointer encoding
4   0x0000c748 0x0000c748 47  48   .rodata ascii   unsupported restore location for flo
at register
5   0x0000c782 0x0000c782 9   10   .rodata ascii   decltype(
6   0x0000c78f 0x0000c78f 8   9    .rodata ascii   typeid (
7   0x0000c798 0x0000c798 5   6    .rodata ascii   {...}
8   0x0000c79e 0x0000c79e 11  12   .rodata ascii   operator>>=
9   0x0000c7aa 0x0000c7aa 11  12   .rodata ascii   operator<=>
10  0x0000c7b6 0x0000c7b6 4   5    .rodata ascii   long
11  0x0000c7bb 0x0000c7bb 8   9    .rodata ascii   char32_t
12  0x0000c7c4 0x0000c7c4 64  65   .rodata ascii   libunwind: malformed DW_CFA_register
 DWARF unwind, reg2 too big\n
13  0x0000c805 0x0000c805 68  69   .rodata ascii   libunwind: malformed DW_CFA_val_offs
et_sf DWARF unwind, reg too big\n
14  0x0000c84e 0x0000c84e 5   6    .rodata ascii   throw
15  0x0000c854 0x0000c854 7   8    .rodata ascii   wchar_t
16  0x0000c85c 0x0000c85c 7   8    .rodata ascii   'lambda
17  0x0000c864 0x0000c864 9   10   .rodata ascii   operator~
18  0x0000c86e 0x0000c86e 11  12   .rodata ascii   operator""
19  0x0000c87a 0x0000c87a 17  18   .rodata ascii   std::basic_string
20  0x0000c88c 0x0000c88c 14  15   .rodata ascii   decltype(auto)
21  0x0000c89b 0x0000c89b 32  33   .rodata ascii   Deleted virtual function called!
22  0x0000c8bc 0x0000c8bc 14  15   .rodata ascii   std::exception
```

```
23  0x0000c8cb 0x0000c8cb 40  41   .rodata ascii   terminating with %s exception of typ
e %s
24  0x0000c8f4 0x0000c8f4 10  11   .rodata ascii   const_cast
25  0x0000c902 0x0000c902 17  18   .rodata ascii   unsigned __int128
26  0x0000c914 0x0000c914 15  16   .rodata ascii   operator delete
27  0x0000c924 0x0000c924 10  11   .rodata ascii   operator>=
28  0x0000c92f 0x0000c92f 13  14   .rodata ascii   unwind_phase2
29  0x0000c93d 0x0000c93d 26  27   .rodata ascii   unsupported arm64 register
30  0x0000c958 0x0000c958 62  63   .rodata ascii   libunwind: malformed DW_CFA_def_cfa
DWARF unwind, reg too big\n
31  0x0000c997 0x0000c997 10  11   .rodata ascii   getSLEB128
32  0x0000c9a2 0x0000c9a2 16  17   .rodata ascii   getSavedRegister
33  0x0000c9bb 0x0000c9bb 18  19   .rodata ascii   typeinfo name for
34  0x0000c9ce 0x0000c9ce 12  13   .rodata ascii   operator new
35  0x0000c9db 0x0000c9db 5   6    .rodata ascii   ) ? (
36  0x0000c9e1 0x0000c9e1 12  13   .rodata ascii    [enable_if:
37  0x0000c9ee 0x0000c9ee 14  15   .rodata ascii   std::nullptr_t
38  0x0000c9fd 0x0000c9fd 11  12   .rodata ascii   objc_object
39  0x0000ca09 0x0000ca09 14  15   .rodata ascii   std::bad_alloc
40  0x0000ca18 0x0000ca18 15  16   .rodata ascii   std::bad_typeid
41  0x0000ca28 0x0000ca28 11  12   .rodata ascii   getEncodedP
42  0x0000ca3e 0x0000ca3e 13  14   .rodata ascii   typeinfo for
43  0x0000ca4c 0x0000ca4c 24  25   .rodata ascii   reference temporary for
44  0x0000ca65 0x0000ca65 13  14   .rodata ascii   unsigned char
45  0x0000ca75 0x0000ca75 10  11   .rodata ascii   operator&=
46  0x0000ca80 0x0000ca80 10  11   .rodata ascii   operator*=
47  0x0000ca8b 0x0000ca8b 70  71   .rodata ascii   std::basic_string<char, std::char_tr
aits<char>, std::allocator<char> >
48  0x0000cad2 0x0000cad2 21  22   .rodata ascii   getSavedFloatRegister
49  0x0000caf6 0x0000caf6 17  18   .rodata ascii   operator delete[]
50  0x0000cb10 0x0000cb10 11  12   .rodata ascii   std::string
51  0x0000cb23 0x0000cb23 4   5    .rodata ascii   auto
52  0x0000cb2a 0x0000cb2a 14  15   .rodata ascii   unsigned short
53  0x0000cb39 0x0000cb39 5   6    .rodata ascii   false
54  0x0000cb3f 0x0000cb3f 4   5    .rodata ascii   %LaL
55  0x0000cb44 0x0000cb44 9   10   .rodata ascii   operator/
56  0x0000cb4e 0x0000cb4e 9   10   .rodata ascii   operator|
57  0x0000cb5c 0x0000cb5c 10  11   .rodata ascii   exynos9810
58  0x0000cb67 0x0000cb67 77  78   .rodata ascii   libunwind: malformed DW_CFA_val_offs
et DWARF unwind, reg (%lu) out of range\n\n
59  0x0000cbb9 0x0000cbb9 19  20   .rodata ascii   FDE has zero length
60  0x0000cbcd 0x0000cbcd 19  20   .rodata ascii   FDE is really a CIE
61  0x0000cbe5 0x0000cbe5 6   7    .rodata ascii   delete
62  0x0000cbec 0x0000cbec 9   10   .rodata ascii   operator&
63  0x0000cbf6 0x0000cbf6 9   10   .rodata ascii   operator%
64  0x0000cc00 0x0000cc00 10  11   .rodata ascii   operator>>
65  0x0000cc0b 0x0000cc0b 5   6    .rodata ascii   ) : (
66  0x0000cc11 0x0000cc11 5   6    .rodata ascii   [abi:
67  0x0000cc1b 0x0000cc1b 65  66   .rodata ascii   libunwind: malformed DW_CFA_same_val
ue DWARF unwind, reg too big\n
68  0x0000cc5d 0x0000cc5d 47  48   .rodata ascii   DW_EH_PE_aligned pointer encoding no
t supported
69  0x0000cc8d 0x0000cc8d 28  29   .rodata ascii   truncated sleb128 expression
70  0x0000ccad 0x0000ccad 39  40   .rodata ascii   terminate_handler unexpectedly retur
ned
71  0x0000ccd7 0x0000ccd7 11  12   .rodata ascii   signed char
```

```
72  0x0000cce6 0x0000cce6 10  11   .rodata ascii   sizeof...(
73  0x0000ccf1 0x0000ccf1 13  14   .rodata ascii   basic_ostream
74  0x0000ccff 0x0000ccff 12  13   .rodata ascii   std::ostream
75  0x0000cd0c 0x0000cd0c 13  14   .rodata ascii   std::iostream
76  0x0000cd1a 0x0000cd1a 9   10   .rodata ascii   long long
77  0x0000cd24 0x0000cd24 9   10   .rodata ascii   noexcept(
78  0x0000cd2e 0x0000cd2e 41  42   .rodata ascii   unsupported restore location for reg
ister
79  0x0000cd6a 0x0000cd6a 14  15   .rodata ascii   operator new[]
80  0x0000cd79 0x0000cd79 9   10   .rodata ascii   operator!
81  0x0000cd83 0x0000cd83 49  50   .rodata ascii   std::basic_ostream<char, std::char_t
raits<char> >
82  0x0000cdb5 0x0000cdb5 10  11   .rodata ascii   __float128
83  0x0000cdc0 0x0000cdc0 8   9    .rodata ascii   char16_t
84  0x0000cdc9 0x0000cdc9 98  99   .rodata ascii   during phase1 personality function s
aid it would stop here, but now in phase2 it did not stop here
85  0x0000ce2c 0x0000ce2c 83  84   .rodata ascii   libunwind: malformed DW_CFA_GNU_nega
tive_offset_extended DWARF unwind, reg too big\n
86  0x0000ce96 0x0000ce96 9   10   .rodata ascii   typename
87  0x0000cea4 0x0000cea4 10  11   .rodata ascii   operator()
88  0x0000ceaf 0x0000ceaf 9   10   .rodata ascii   operator>
89  0x0000ceb9 0x0000ceb9 10  11   .rodata ascii   operator[]
90  0x0000cec4 0x0000cec4 10  11   .rodata ascii   operator->
91  0x0000cecf 0x0000cecf 13  14   .rodata ascii   unsigned long
92  0x0000cee1 0x0000cee1 13  14   .rodata ascii   std::bad_cast
93  0x0000ceef 0x0000ceef 11  12   .rodata ascii   setRegister
94  0x0000cefb 0x0000cefb 70  71   .rodata ascii   libunwind: malformed DW_CFA_offset_e
xtended DWARF unwind, reg too big\n
95  0x0000cf45 0x0000cf45 11  12   .rodata ascii   > typename
96  0x0000cf51 0x0000cf51 21  22   .rodata ascii   (anonymous namespace)
97  0x0000cf67 0x0000cf67 10  11   .rodata ascii   operator==
98  0x0000cf72 0x0000cf72 8   9    .rodata ascii    complex
99  0x0000cf7b 0x0000cf7b 25  26   .rodata ascii   CIE version is not 1 or 3
100 0x0000cf9d 0x0000cf9d 11  12   .rodata ascii   vtable for
101 0x0000cfa9 0x0000cfa9 8   9    .rodata ascii   VTT for
102 0x0000cfb2 0x0000cfb2 9   10   .rodata ascii   alignof (
103 0x0000cfbe 0x0000cfbe 10  11   .rodata ascii   noexcept (
104 0x0000cfc9 0x0000cfc9 4   5    .rodata ascii   char
105 0x0000cfd0 0x0000cfd0 9   10   .rodata ascii   operator<
106 0x0000cfda 0x0000cfda 11  12   .rodata ascii   operator->*
107 0x0000cfe6 0x0000cfe6 12  13   .rodata ascii   unsigned int
108 0x0000cff3 0x0000cff3 47  48   .rodata ascii   DW_EH_PE_funcrel pointer encoding no
t supported
109 0x0000d023 0x0000d023 45  46   .rodata ascii   libunwind: Unsupported .eh_frame_hdr
 version\n
110 0x0000d055 0x0000d055 9   10   .rodata ascii   libc++abi
111 0x0000d05f 0x0000d05f 12  13   .rodata ascii   dynamic_cast
112 0x0000d074 0x0000d074 5   6    .rodata ascii   short
113 0x0000d07a 0x0000d07a 5   6    .rodata ascii    ...
114 0x0000d080 0x0000d080 6   7    .rodata ascii   string
115 0x0000d087 0x0000d087 7   8    .rodata ascii   ostream
116 0x0000d08f 0x0000d08f 11  12   .rodata ascii   long double
117 0x0000d0a2 0x0000d0a2 10  11   .rodata ascii   unexpected
118 0x0000d0ad 0x0000d0ad 19  20   .rodata ascii   guard variable for
119 0x0000d0c4 0x0000d0c4 4   5    .rodata ascii   true
120 0x0000d0c9 0x0000d0c9 9   10   .rodata ascii   operator?
```

```
121 0x0000d0d3 0x0000d0d3 20  21   .rodata ascii   bad_array_new_length
122 0x0000d0e8 0x0000d0e8 19  20   .rodata ascii   libunwind: %s - %s\n
123 0x0000d103 0x0000d103 17  18   .rodata ascii   virtual thunk to
124 0x0000d123 0x0000d123 9   10   .rodata ascii   operator*
125 0x0000d12d 0x0000d12d 10  11   .rodata ascii   operator||
126 0x0000d138 0x0000d138 7   8    .rodata ascii   istream
127 0x0000d144 0x0000d144 7   8    .rodata ascii   char8_t
128 0x0000d14c 0x0000d14c 30  31   .rodata ascii   DW_OP_deref_size with bad size
129 0x0000d16b 0x0000d16b 40  41   .rodata ascii   Unknown DWARF encoding for search ta
ble.
130 0x0000d19c 0x0000d19c 40  41   .rodata ascii   unexpected_handler unexpectedly retu
rned
131 0x0000d1c5 0x0000d1c5 24  25   .rodata ascii   construction vtable for
132 0x0000d1e3 0x0000d1e3 8   9    .rodata ascii   __int128
133 0x0000d1ec 0x0000d1ec 9   10   .rodata ascii   template<
134 0x0000d1f6 0x0000d1f6 10  11   .rodata ascii   operator<<
135 0x0000d201 0x0000d201 9   10   .rodata ascii   operator+
136 0x0000d20b 0x0000d20b 10  11   .rodata ascii   operator+=
137 0x0000d216 0x0000d216 10  11   .rodata ascii   operator++
138 0x0000d221 0x0000d221 14  15   .rodata ascii   string literal
139 0x0000d230 0x0000d230 18  19   .rodata ascii   unsigned long long
140 0x0000d243 0x0000d243 10  11   .rodata ascii    imaginary
141 0x0000d24e 0x0000d24e 65  66   .rodata ascii   libunwind: malformed DW_CFA_expressi
on DWARF unwind, reg too big\n
142 0x0000d2a6 0x0000d2a6 9   10   .rodata ascii   operator=
143 0x0000d2b0 0x0000d2b0 10  11   .rodata ascii   operator/=
144 0x0000d2bb 0x0000d2bb 4   5    .rodata ascii   bool
145 0x0000d2c0 0x0000d2c0 18  19   .rodata ascii   evaluateExpression
146 0x0000d2de 0x0000d2de 9   10   .rodata ascii   operator^
147 0x0000d2e8 0x0000d2e8 9   10   .rodata ascii    restrict
148 0x0000d2f2 0x0000d2f2 9   10   .rodata ascii   decimal64
149 0x0000d2fc 0x0000d2fc 64  65   .rodata ascii   libunwind: malformed DW_CFA_undefine
d DWARF unwind, reg too big\n
150 0x0000d34c 0x0000d34c 44  45   .rodata ascii   terminating with %s exception of typ
e %s: %s
151 0x0000d379 0x0000d379 21  22   .rodata ascii   non-virtual thunk to
152 0x0000d396 0x0000d396 49  50   .rodata ascii   std::basic_istream<char, std::char_t
raits<char> >
153 0x0000d3c8 0x0000d3c8 8   9    .rodata ascii   iostream
154 0x0000d3d1 0x0000d3d1 13  14   .rodata ascii   pixel vector[
155 0x0000d3df 0x0000d3df 5   6    .rodata ascii   union
156 0x0000d3e5 0x0000d3e5 29  30   .rodata ascii   _Unwind_Resume() can't return
157 0x0000d403 0x0000d403 63  64   .rodata ascii   libunwind: malformed DW_CFA_register
 DWARF unwind, reg too big\n
158 0x0000d446 0x0000d446 4   5    .rodata ascii   yptn
159 0x0000d44b 0x0000d44b 10  11   .rodata ascii   operator%=
160 0x0000d456 0x0000d456 6   7    .rodata ascii    const
161 0x0000d45d 0x0000d45d 27  28   .rodata ascii   DW_OP_fbreg not implemented
162 0x0000d481 0x0000d481 37  38   .rodata ascii   terminating with %s foreign exceptio
n
163 0x0000d4a7 0x0000d4a7 8   9    .rodata ascii   uncaught
164 0x0000d4b3 0x0000d4b3 10  11   .rodata ascii   operator--
165 0x0000d4be 0x0000d4be 10  11   .rodata ascii   operator|=
166 0x0000d4c9 0x0000d4c9 50  51   .rodata ascii   std::basic_iostream<char, std::char_
traits<char> >
167 0x0000d4fc 0x0000d4fc 14  15   .rodata ascii   _Unwind_Resume
```

```
168 0x0000d50b 0x0000d50b 65  66   .rodata ascii   libunwind: malformed DW_CFA_def_cfa_
sf DWARF unwind, reg too big\n
169 0x0000d55d 0x0000d55d 15  16   .rodata ascii   'block-literal'
170 0x0000d56d 0x0000d56d 9   10   .rodata ascii   operator-
171 0x0000d577 0x0000d577 13  14   .rodata ascii   basic_istream
172 0x0000d585 0x0000d585 12  13   .rodata ascii   std::istream
173 0x0000d592 0x0000d592 6   7    .rodata ascii   double
174 0x0000d59c 0x0000d59c 33  34   .rodata ascii   invocation function for block in
175 0x0000d5be 0x0000d5be 11  12   .rodata ascii   static_cast
176 0x0000d5ca 0x0000d5ca 11  12   .rodata ascii   sizeof... (
177 0x0000d5dc 0x0000d5dc 10  11   .rodata ascii   operator-=
178 0x0000d5e7 0x0000d5e7 73  74   .rodata ascii   libunwind: malformed DW_CFA_offset_e
xtended_sf DWARF unwind, reg too big\n
179 0x0000d631 0x0000d631 10  11   .rodata ascii   getULEB128
180 0x0000d63c 0x0000d63c 28  29   .rodata ascii   malformed uleb128 expression
181 0x0000d659 0x0000d659 28  29   .rodata ascii   DWARF opcode not implemented
182 0x0000d683 0x0000d683 7   8    .rodata ascii   nullptr
183 0x0000d68b 0x0000d68b 11  12   .rodata ascii   operator<<=
184 0x0000d697 0x0000d697 11  12   .rodata ascii   ::operator
185 0x0000d6a3 0x0000d6a3 4   5    .rodata ascii   enum
186 0x0000d6a8 0x0000d6a8 69  70   .rodata ascii   libunwind: malformed DW_CFA_val_expr
ession DWARF unwind, reg too big\n
187 0x0000d6fb 0x0000d6fb 11  12   .rodata ascii   terminating
188 0x0000d70b 0x0000d70b 16  17   .rodata ascii   reinterpret_cast
189 0x0000d721 0x0000d721 47  48   .rodata ascii   DW_EH_PE_textrel pointer encoding no
t supported
190 0x0000d751 0x0000d751 28  29   .rodata ascii   truncated uleb128 expression
191 0x0000d773 0x0000d773 9   10   .rodata ascii   operator
192 0x0000d77d 0x0000d77d 6   7    .rodata ascii   throw
193 0x0000d784 0x0000d784 12  13   .rodata ascii   basic_string
194 0x0000d791 0x0000d791 4   5    .rodata ascii   void
195 0x0000d796 0x0000d796 5   6    .rodata ascii   float
196 0x0000d79c 0x0000d79c 10  11   .rodata ascii   decimal128
197 0x0000d7a7 0x0000d7a7 7   8    .rodata ascii   ro.arch
198 0x0000d7af 0x0000d7af 71  72   .rodata ascii   libunwind: malformed DW_CFA_restore_
extended DWARF unwind, reg too big\n
199 0x0000d7f7 0x0000d7f7 17  18   .rodata ascii   getTableEntrySize
200 0x0000d81f 0x0000d81f 10  11   .rodata ascii   operator&&
201 0x0000d82a 0x0000d82a 9   10   .rodata ascii   decimal32
202 0x0000d834 0x0000d834 18  19   .rodata ascii   CIE ID is not zero
203 0x0000d84f 0x0000d84f 33  34   .rodata ascii   thread-local wrapper routine for
204 0x0000d871 0x0000d871 40  41   .rodata ascii   thread-local initialization routine
for
205 0x0000d89a 0x0000d89a 8   9    .rodata ascii   sizeof (
206 0x0000d8a3 0x0000d8a3 10  11   .rodata ascii   operator|=
207 0x0000d8ae 0x0000d8ae 9   10   .rodata ascii   __uuidof(
208 0x0000d8b8 0x0000d8b8 14  15   .rodata ascii   std::allocator
209 0x0000d8c7 0x0000d8c7 9   10   .rodata ascii   allocator
210 0x0000d8d1 0x0000d8d1 6   7    .rodata ascii   struct
211 0x0000d8d8 0x0000d8d8 71  72   .rodata ascii   libunwind: malformed DW_CFA_def_cfa_
register DWARF unwind, reg too big\n
212 0x0000d920 0x0000d920 52  53   .rodata ascii   Can't binary search on variable leng
th encoded data.
213 0x0000d95d 0x0000d95d 49  50   .rodata ascii   terminate_handler unexpectedly threw
 an exception
214 0x0000d992 0x0000d992 9   10   .rodata ascii   operator,
215 0x0000d99c 0x0000d99c 9   10    .rodata ascii   decimal16
216 0x0000d9a6 0x0000d9a6 8   9     .rodata ascii   noexcept
217 0x0000d9af 0x0000d9af 11  12    .rodata ascii   getRegister
218 0x0000d9bb 0x0000d9bb 51  52    .rodata ascii   DW_EH_PE_datarel is invalid with a d
atarelBase of 0
219 0x0000d9f9 0x0000d9f9 16  17    .rodata ascii   unknown register
220 0x0000da0c 0x0000da0c 14  15    .rodata ascii   basic_iostream
221 0x0000da1b 0x0000da1b 5   6     .rodata ascii   std::
222 0x0000da21 0x0000da21 9   10    .rodata ascii    volatile
223 0x0000da2b 0x0000da2b 6   7     .rodata ascii   throw(
224 0x0000da32 0x0000da32 29  30    .rodata ascii   Pure virtual function called
225 0x0000da50 0x0000da50 18  19    .rodata ascii   std::bad_exception
226 0x0000da63 0x0000da63 27  28    .rodata ascii   DW_OP_piece not implemented
227 0x0000df91 0x0000df91 5   6     .rodata ascii   \t\t\t\t\t
228 0x0000df97 0x0000df97 25  26    .rodata ascii   \t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\
t\t\t\t\t\t\t
229 0x0000dfb1 0x0000dfb1 5   6     .rodata ascii   \t\t\t\t\t
230 0x0000dfc2 0x0000dfc2 4   5     .rodata ascii   Ocv}
231 0x0000e135 0x0000e135 14  15    .rodata ascii   \t7\t\t\t\t\t\t\t\t\t\t\t\t
232 0x0000e150 0x0000e150 32  65    .rodata utf16le PPPPPPPPPPPPPPPPPPPPPPPPPPPPP~PP\
233 0x0000e194 0x0000e194 17  35    .rodata utf16le ddddddddddddddddd
234 0x0000e1c0 0x0000e1c0 9   19    .rodata utf16le ddddddddd
235 0x0000e27a 0x0000e27a 49  50    .rodata ascii    (N12_GLOBAL__N_116itanium_demangle1
1SpecialNameE
236 0x0000e2ac 0x0000e2ac 39  40    .rodata ascii   N12_GLOBAL__N_116itanium_demangle4No
deE
237 0x0000e2d4 0x0000e2d4 57  58    .rodata ascii   N12_GLOBAL__N_116itanium_demangle21C
torVtableSpecialNameE
238 0x0000e30e 0x0000e30e 43  44    .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Na
meTypeE
239 0x0000e33a 0x0000e33a 46  47    .rodata ascii   N12_GLOBAL__N_116itanium_demangle10N
estedNameE
240 0x0000e369 0x0000e369 60  61    .rodata ascii   N12_GLOBAL__N_116itanium_demangle24F
orwardTemplateReferenceE
241 0x0000e3a6 0x0000e3a6 50  51    .rodata ascii   N12_GLOBAL__N_116itanium_demangle14I
ntegerLiteralE
242 0x0000e3d9 0x0000e3d9 43  44    .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Bo
olExprE
243 0x0000e405 0x0000e405 55  56    .rodata ascii   N12_GLOBAL__N_116itanium_demangle16F
loatLiteralImplIfEE
244 0x0000e43d 0x0000e43d 55  56    .rodata ascii   N12_GLOBAL__N_116itanium_demangle16F
loatLiteralImplIdEE
245 0x0000e475 0x0000e475 55  56    .rodata ascii   N12_GLOBAL__N_116itanium_demangle16F
loatLiteralImplIeEE
246 0x0000e4ad 0x0000e4ad 49  50    .rodata ascii   N12_GLOBAL__N_116itanium_demangle13S
tringLiteralE
247 0x0000e4df 0x0000e4df 51  52    .rodata ascii   N12_GLOBAL__N_116itanium_demangle15U
nnamedTypeNameE
248 0x0000e513 0x0000e513 62  63    .rodata ascii   N12_GLOBAL__N_116itanium_demangle26S
yntheticTemplateParamNameE
249 0x0000e552 0x0000e552 57  58    .rodata ascii   N12_GLOBAL__N_116itanium_demangle21T
ypeTemplateParamDeclE
250 0x0000e58c 0x0000e58c 60  61    .rodata ascii   N12_GLOBAL__N_116itanium_demangle24N
onTypeTemplateParamDeclE
251 0x0000e5c9 0x0000e5c9 61  62    .rodata ascii   N12_GLOBAL__N_116itanium_demangle25T
emplateTemplateParamDeclE
```

```
252 0x0000e607 0x0000e607 57  58   .rodata ascii   N12_GLOBAL__N_116itanium_demangle21T
emplateParamPackDeclE
253 0x0000e641 0x0000e641 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15C
losureTypeNameE
254 0x0000e675 0x0000e675 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10L
ambdaExprE
255 0x0000e6a4 0x0000e6a4 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15I
ntegerCastExprE
256 0x0000e6d8 0x0000e6d8 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13F
unctionParamE
257 0x0000e70a 0x0000e70a 43  44   .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Fo
ldExprE
258 0x0000e736 0x0000e736 58  59   .rodata ascii   N12_GLOBAL__N_116itanium_demangle22P
arameterPackExpansionE
259 0x0000e771 0x0000e771 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10B
inaryExprE
260 0x0000e7a0 0x0000e7a0 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10P
refixExprE
261 0x0000e7cf 0x0000e7cf 43  44   .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Ca
stExprE
262 0x0000e7fb 0x0000e7fb 43  44   .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Ca
llExprE
263 0x0000e827 0x0000e827 50  51   .rodata ascii   N12_GLOBAL__N_116itanium_demangle14C
onversionExprE
264 0x0000e85a 0x0000e85a 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10D
eleteExprE
265 0x0000e889 0x0000e889 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13Q
ualifiedNameE
266 0x0000e8bb 0x0000e8bb 43  44   .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Dt
orNameE
267 0x0000e8e7 0x0000e8e7 58  59   .rodata ascii   N12_GLOBAL__N_116itanium_demangle22C
onversionOperatorTypeE
268 0x0000e922 0x0000e922 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15L
iteralOperatorE
269 0x0000e956 0x0000e956 55  56   .rodata ascii   N12_GLOBAL__N_116itanium_demangle19G
lobalQualifiedNameE
270 0x0000e98e 0x0000e98e 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10M
emberExprE
271 0x0000e9bd 0x0000e9bd 54  55   .rodata ascii   N12_GLOBAL__N_116itanium_demangle18A
rraySubscriptExprE
272 0x0000e9f4 0x0000e9f4 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10B
racedExprE
273 0x0000ea23 0x0000ea23 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15B
racedRangeExprE
274 0x0000ea57 0x0000ea57 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12I
nitListExprE
275 0x0000ea88 0x0000ea88 47  48   .rodata ascii   N12_GLOBAL__N_116itanium_demangle11P
ostfixExprE
276 0x0000eab8 0x0000eab8 42  43   .rodata ascii   N12_GLOBAL__N_116itanium_demangle7Ne
wExprE
277 0x0000eae3 0x0000eae3 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13E
nclosingExprE
278 0x0000eb15 0x0000eb15 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15C
onditionalExprE
279 0x0000eb49 0x0000eb49 55  56   .rodata ascii   N12_GLOBAL__N_116itanium_demangle19S
izeofParamPackExprE
```

```
280 0x0000eb81 0x0000eb81 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13N
odeArrayNodeE
281 0x0000ebb3 0x0000ebb3 44  45   .rodata ascii   N12_GLOBAL__N_116itanium_demangle9Th
rowExprE
282 0x0000ebe0 0x0000ebe0 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10U
UIDOfExprE
283 0x0000ec0f 0x0000ec0f 63  64   .rodata ascii   N12_GLOBAL__N_116itanium_demangle27E
xpandedSpecialSubstitutionE
284 0x0000ec4f 0x0000ec4f 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12C
torDtorNameE
285 0x0000ec80 0x0000ec80 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10A
biTagAttrE
286 0x0000ecaf 0x0000ecaf 57  58   .rodata ascii   N12_GLOBAL__N_116itanium_demangle21S
tructuredBindingNameE
287 0x0000ece9 0x0000ece9 44  45   .rodata ascii   N12_GLOBAL__N_116itanium_demangle9Lo
calNameE
288 0x0000ed16 0x0000ed16 55  56   .rodata ascii   N12_GLOBAL__N_116itanium_demangle19S
pecialSubstitutionE
289 0x0000ed4e 0x0000ed4e 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13P
arameterPackE
290 0x0000ed80 0x0000ed80 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12T
emplateArgsE
291 0x0000edb1 0x0000edb1 56  57   .rodata ascii   N12_GLOBAL__N_116itanium_demangle20N
ameWithTemplateArgsE
292 0x0000edea 0x0000edea 52  53   .rodata ascii   N12_GLOBAL__N_116itanium_demangle16S
tdQualifiedNameE
293 0x0000ee1f 0x0000ee1f 56  57   .rodata ascii   N12_GLOBAL__N_116itanium_demangle20T
emplateArgumentPackE
294 0x0000ee58 0x0000ee58 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12E
nableIfAttrE
295 0x0000ee89 0x0000ee89 52  53   .rodata ascii   N12_GLOBAL__N_116itanium_demangle16F
unctionEncodingE
296 0x0000eebe 0x0000eebe 44  45   .rodata ascii   N12_GLOBAL__N_116itanium_demangle9Do
tSuffixE
297 0x0000eeeb 0x0000eeeb 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12N
oexceptSpecE
298 0x0000ef1c 0x0000ef1c 56  57   .rodata ascii   N12_GLOBAL__N_116itanium_demangle20D
ynamicExceptionSpecE
299 0x0000ef55 0x0000ef55 48  49   .rodata ascii   N12_GLOBAL__N_116itanium_demangle12F
unctionTypeE
300 0x0000ef86 0x0000ef86 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13O
bjCProtoNameE
301 0x0000efb8 0x0000efb8 53  54   .rodata ascii   N12_GLOBAL__N_116itanium_demangle17V
endorExtQualTypeE
302 0x0000efee 0x0000efee 43  44   .rodata ascii   N12_GLOBAL__N_116itanium_demangle8Qu
alTypeE
303 0x0000f01a 0x0000f01a 51  52   .rodata ascii   N12_GLOBAL__N_116itanium_demangle15P
ixelVectorTypeE
304 0x0000f04e 0x0000f04e 46  47   .rodata ascii   N12_GLOBAL__N_116itanium_demangle10V
ectorTypeE
305 0x0000f07d 0x0000f07d 44  45   .rodata ascii   N12_GLOBAL__N_116itanium_demangle9Ar
rayTypeE
306 0x0000f0aa 0x0000f0aa 55  56   .rodata ascii   N12_GLOBAL__N_116itanium_demangle19P
ointerToMemberTypeE
307 0x0000f0e2 0x0000f0e2 58  59   .rodata ascii   N12_GLOBAL__N_116itanium_demangle22E
laboratedTypeSpefTypeE
```

```
308 0x0000f11d 0x0000f11d 47  48   .rodata ascii   N12_GLOBAL__N_116itanium_demangle11P
ointerTypeE
309 0x0000f14d 0x0000f14d 49  50   .rodata ascii   N12_GLOBAL__N_116itanium_demangle13R
eferenceTypeE
310 0x0000f17f 0x0000f17f 56  57   .rodata ascii   N12_GLOBAL__N_116itanium_demangle20P
ostfixQualifiedTypeE
311 0x0000f21d 0x0000f21d 5   6    .rodata ascii   KKKK6
312 0x0000f290 0x0000f290 32  33   .rodata ascii   N10__cxxabiv116__shim_type_infoE
313 0x0000f2b1 0x0000f2b1 33  34   .rodata ascii   N10__cxxabiv117__class_type_infoE
314 0x0000f2d3 0x0000f2d3 33  34   .rodata ascii   N10__cxxabiv117__pbase_type_infoE
315 0x0000f2f5 0x0000f2f5 35  36   .rodata ascii   N10__cxxabiv119__pointer_type_infoE
316 0x0000f319 0x0000f319 36  37   .rodata ascii   N10__cxxabiv120__function_type_infoE
317 0x0000f33e 0x0000f33e 45  46   .rodata ascii   N10__cxxabiv129__pointer_to_member_t
ype_infoE
318 0x0000f388 0x0000f388 39  40   .rodata ascii   N10__cxxabiv123__fundamental_type_in
foE
319 0x0000f3c0 0x0000f3c0 4   5    .rodata ascii   PKDn
320 0x0000f453 0x0000f453 4   5    .rodata ascii   PKDh
321 0x0000f483 0x0000f483 4   5    .rodata ascii   PKDu
322 0x0000f48f 0x0000f48f 4   5    .rodata ascii   PKDs
323 0x0000f49b 0x0000f49b 4   5    .rodata ascii   PKDi
324 0x0000f4a0 0x0000f4a0 33  34   .rodata ascii   N10__cxxabiv117__array_type_infoE
325 0x0000f4c2 0x0000f4c2 32  33   .rodata ascii   N10__cxxabiv116__enum_type_infoE
326 0x0000f4e3 0x0000f4e3 36  37   .rodata ascii   N10__cxxabiv120__si_class_type_infoE
327 0x0000f508 0x0000f508 37  38   .rodata ascii   N10__cxxabiv121__vmi_class_type_info
E
328 0x0000f52e 0x0000f52e 12  13   .rodata ascii   St9exception
329 0x0000f53b 0x0000f53b 17  18   .rodata ascii   St13bad_exception
330 0x0000f54d 0x0000f54d 12  13   .rodata ascii   St9bad_alloc
331 0x0000f55a 0x0000f55a 24  25   .rodata ascii   St20bad_array_new_length
332 0x0000f573 0x0000f573 12  13   .rodata ascii   St9type_info
333 0x0000f580 0x0000f580 11  12   .rodata ascii   St8bad_cast
334 0x0000f58c 0x0000f58c 14  15   .rodata ascii   St10bad_typeid
335 0x0000f610 0x0000f610 30  124  .rodata utf32le DPppppppppppppppppppppppppppppp
336 0x0000f6a4 0x0000f6a4 4   20   .rodata utf32le P|`l
337 0x0000f6da 0x0000f6da 6   14   .rodata utf16le \e,7L`|
338 0x0000f76a 0x0000f76a 5   6    .rodata ascii   +\n:IE
339 0x0000f808 0x0000f808 32  66   .rodata utf16le &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
0   0x000cca31 0x000cea31 4   5    .data   ascii   STSN
1   0x000cca40 0x000cea40 17  18   .data   ascii   G1ARyk_p}ooUroh}r
...
70  0x000cd304 0x000cf304 4   5    .data   ascii   hi\t@
71  0x000cd30e 0x000cf30e 14  16   .data   utf8    ā#Gaj}j$gjel$X blocks=Latin Extended
-B,Basic Latin
72  0x000cd31e 0x000cf31e 20  21   .data   ascii   ybel0B"Gaj}j$gjel$Be
73  0x000cd333 0x000cf333 6   7    .data   ascii   nlny0\v
74  0x000cd340 0x000cf340 10  11   .data   ascii   ?=,\b9;39?=
75  0x000cd34b 0x000cf34b 4   5    .data   ascii   6>7X
...
195 0x000ce8b0 0x000d08b0 11  12   .data   ascii   2V{t~hus~5t
196 0x000ce8bc 0x000d08bc 97  98   .data   ascii   n5Ohs|AVp{l{5v{t}5Inhst}|Vp{l{5v{t}5
Inhst}|AVp{l{5v{t}5Inhst}|Vp{l{5v{t}5Inhst}|3V{t~hus~5~{n{x{i
197 0x000ce91e 0x000d091e 8   9    .data   ascii   5Yohiuh|
...
225 0x000cef2e 0x000d0f2e 31  32   .data   ascii   P-]ahl~h-nlaa-dcdy%$-bc-`ldc-ye
226 0x000cef4e 0x000d0f4e 5   6    .data   ascii   hli#\r
```

```
227 0x000cef58 0x000d0f58 19  20    .data    ascii    ALVUID\%l mzzq@lwjk
228 0x000cef74 0x000d0f74 5   6     .data    ascii    _VKZ\
229 0x000cef82 0x000d0f82 13  14    .data    ascii    wXMPO\q\UI\K9
```

-s：symbols

```
➜  arm64-v8a rabin2 -s libtacker.so
[Symbols]
nth paddr      vaddr      bind   type size  lib name                               dem
angled
―――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――――
――――――
48  0x00044ce8 0x00044ce8 GLOBAL FUNC 6608      .datadiv_decode16117807209816376729
49  0x00078a04 0x00078a04 GLOBAL FUNC 2696      .datadiv_decode9901940071257331957
...
73  0x000a8884 0x000a8884 GLOBAL FUNC 4         .datadiv_decode11706101414295225912
74  0x000aa438 0x000aa438 GLOBAL FUNC 1436      JNI_OnLoad
75  0x00026d98 0x00026d98 GLOBAL FUNC 18656     .datadiv_decode12335027288954124723
...
117 0x000a5e74 0x000a5e74 GLOBAL FUNC 4         .datadiv_decode5454406552017557296
1   0x000c9240 0x000c9240 GLOBAL FUNC 16        imp.__cxa_finalize
2   0x000c9250 0x000c9250 GLOBAL FUNC 16        imp.__cxa_atexit
3   0x000c9260 0x000c9260 GLOBAL FUNC 16        imp.__android_log_print
4   0x000c9270 0x000c9270 GLOBAL FUNC 16        imp.__stack_chk_fail
5   0x000c9280 0x000c9280 GLOBAL FUNC 16        imp.memset
6   0x000c9290 0x000c9290 GLOBAL FUNC 16        imp.strncpy
7   0x000c92a0 0x000c92a0 GLOBAL FUNC 16        imp.strncat
8   0x000c92b0 0x000c92b0 GLOBAL FUNC 16        imp.pthread_self
9   0x000c92c0 0x000c92c0 GLOBAL FUNC 16        imp.malloc
10  0x000c92d0 0x000c92d0 GLOBAL FUNC 16        imp.free
11  0x000c92e0 0x000c92e0 GLOBAL FUNC 16        imp.posix_memalign
12  ---------- ---------- GLOBAL OBJ  16        imp.__sF
13  0x000c92f0 0x000c92f0 GLOBAL FUNC 16        imp.vfprintf
14  0x000c9300 0x000c9300 GLOBAL FUNC 16        imp.fputc
15  0x000c9310 0x000c9310 GLOBAL FUNC 16        imp.vasprintf
16  0x000c9320 0x000c9320 GLOBAL FUNC 16        imp.android_set_abort_message
17  0x000c9330 0x000c9330 GLOBAL FUNC 16        imp.openlog
18  0x000c9340 0x000c9340 GLOBAL FUNC 16        imp.syslog
19  0x000c9350 0x000c9350 GLOBAL FUNC 16        imp.closelog
20  0x000c9360 0x000c9360 GLOBAL FUNC 16        imp.abort
21  0x000c9370 0x000c9370 GLOBAL FUNC 16        imp.strlen
22  0x000c9380 0x000c9380 GLOBAL FUNC 16        imp.realloc
23  0x000c9390 0x000c9390 GLOBAL FUNC 16        imp.memmove
24  0x000c93a0 0x000c93a0 GLOBAL FUNC 16        imp.__memmove_chk
25  0x000c93b0 0x000c93b0 GLOBAL FUNC 16        imp.__strlen_chk
26  0x000c93c0 0x000c93c0 GLOBAL FUNC 16        imp.memchr
27  0x000c93d0 0x000c93d0 GLOBAL FUNC 16        imp.__vsnprintf_chk
28  0x000c93e0 0x000c93e0 GLOBAL FUNC 16        imp.memcpy
29  0x000c93f0 0x000c93f0 GLOBAL FUNC 16        imp.pthread_mutex_lock
30  0x000c9400 0x000c9400 GLOBAL FUNC 16        imp.pthread_mutex_unlock
31  0x000c9410 0x000c9410 GLOBAL FUNC 16        imp.calloc
32  0x000c9420 0x000c9420 GLOBAL FUNC 16        imp.strcmp
33  0x000c9430 0x000c9430 GLOBAL FUNC 16        imp.pthread_getspecific
```

```
34  0x000c9440 0x000c9440 GLOBAL FUNC 16        imp.pthread_once
35  0x000c9450 0x000c9450 GLOBAL FUNC 16        imp.pthread_setspecific
36  0x000c9460 0x000c9460 GLOBAL FUNC 16        imp.pthread_key_delete
37  0x000c9470 0x000c9470 GLOBAL FUNC 16        imp.pthread_key_create
38  0x000c9480 0x000c9480 GLOBAL FUNC 16        imp.getauxval
39  0x000c9490 0x000c9490 GLOBAL FUNC 16        imp.__system_property_get
40  0x000c94a0 0x000c94a0 GLOBAL FUNC 16        imp.strncmp
41  0x000c94b0 0x000c94b0 GLOBAL FUNC 16        imp.fprintf
42  0x000c94c0 0x000c94c0 GLOBAL FUNC 16        imp.fflush
43  0x000c94d0 0x000c94d0 GLOBAL FUNC 16        imp.pthread_rwlock_wrlock
44  0x000c94e0 0x000c94e0 GLOBAL FUNC 16        imp.pthread_rwlock_unlock
45  0x000c94f0 0x000c94f0 GLOBAL FUNC 16        imp.dl_iterate_phdr
46  0x000c9500 0x000c9500 GLOBAL FUNC 16        imp.pthread_rwlock_rdlock
47  0x000c9510 0x000c9510 GLOBAL FUNC 16        imp.fwrite
```

-S : sections

```
➜  arm64-v8a rabin2 -S libtacker.so
[Sections]

nth paddr          size vaddr         vsize perm type        name
──────────────────────────────────────────────────────────────────
0   0x00000000      0x0 0x00000000      0x0 ---- NULL
1   0x00000238     0x98 0x00000238     0x98 -r-- NOTE        .note.android.ident
2   0x000002d0     0x24 0x000002d0     0x24 -r-- NOTE        .note.gnu.build-id
3   0x000002f8    0xb10 0x000002f8    0xb10 -r-- DYNSYM      .dynsym
4   0x00000e08     0xec 0x00000e08     0xec -r-- GNU_VERSYM  .gnu.version
5   0x00000ef4     0x40 0x00000ef4     0x40 -r-- GNU_VERNEED .gnu.version_r
6   0x00000f38    0x1ec 0x00000f38    0x1ec -r-- GNU_HASH    .gnu.hash
7   0x00001124    0x3b8 0x00001124    0x3b8 -r-- HASH        .hash
8   0x000014dc    0xc19 0x000014dc    0xc19 -r-- STRTAB      .dynstr
9   0x000020f8   0x8850 0x000020f8   0x8850 -r-- RELA        .rela.dyn
10  0x0000a948    0x450 0x0000a948    0x450 -r-- RELA        .rela.plt
11  0x0000ad98   0x1960 0x0000ad98   0x1960 -r-- PROGBITS    .gcc_except_table
12  0x0000c6f8   0x3434 0x0000c6f8   0x3434 -r-- PROGBITS    .rodata
13  0x0000fb2c   0x1dbc 0x0000fb2c   0x1dbc -r-- PROGBITS    .eh_frame_hdr
14  0x000118e8   0x8cd4 0x000118e8   0x8cd4 -r-- PROGBITS    .eh_frame
15  0x0001a5c0  0xaec60 0x0001a5c0  0xaec60 -r-x PROGBITS    .text
16  0x000c9220    0x300 0x000c9220    0x300 -r-x PROGBITS    .plt
17  0x000c9520   0x2eb8 0x000ca520   0x2eb8 -rw- PROGBITS    .data.rel.ro
18  0x000cc3d8     0x10 0x000cd3d8     0x10 -rw- FINI_ARRAY  .fini_array
19  0x000cc3e8    0x230 0x000cd3e8    0x230 -rw- INIT_ARRAY  .init_array
20  0x000cc618    0x1d0 0x000cd618    0x1d0 -rw- DYNAMIC     .dynamic
21  0x000cc7e8     0xc0 0x000cd7e8     0xc0 -rw- PROGBITS    .got
22  0x000cc8a8    0x188 0x000cd8a8    0x188 -rw- PROGBITS    .got.plt
23  0x000cca30   0x25d8 0x000cea30   0x25d8 -rw- PROGBITS    .data
24  0x000cf008      0x0 0x000d1010    0xad0 -rw- NOBITS      .bss
25  0x000cf008     0xc6 0x00000000     0xc6 ---- PROGBITS    .comment
26  0x000cf0ce    0x104 0x00000000    0x104 ---- STRTAB      .shstrtab
```

-SS : segments

```
➜  arm64-v8a rabin2 -SS libtacker.so
[Segments]

nth paddr          size vaddr         vsize perm type name
―――――――――――――――――――――――――――――――――――――――――――――――――――――――――
0   0x00000040    0x1f8 0x00000040    0x1f8 -r-- MAP  PHDR
1   0x00000000  0xc9520 0x00000000  0xc9520 -r-x MAP  LOAD0
2   0x000c9520   0x3510 0x000ca520   0x3510 -rw- MAP  LOAD1
3   0x000cca30   0x25d8 0x000cea30   0x30b0 -rw- MAP  LOAD2
4   0x000cc618    0x1d0 0x000cd618    0x1d0 -rw- MAP  DYNAMIC
5   0x000c9520   0x3510 0x000ca520   0x3ae0 -r-- MAP  GNU_RELRO
6   0x0000fb2c   0x1dbc 0x0000fb2c   0x1dbc -r-- MAP  GNU_EH_FRAME
7   0x00000000      0x0 0x00000000      0x0 -rw- MAP  GNU_STACK
8   0x00000238     0xbc 0x00000238     0xbc -r-- MAP  NOTE
9   0x00000000     0x40 0x00000000     0x40 -rw- MAP  ehdr
```

rabin2的help语法

```
➜  ~ rabin2 -h
Usage: rabin2 [-AcdeEghHiIjlLMqrRsSUvVxzZ] [-@ at] [-a arch] [-b bits] [-B addr]
              [-C F:C:D] [-f str] [-m addr] [-n str] [-N m:M] [-P[-P] pdb]
              [-o str] [-O str] [-k query] [-D lang mangledsymbol] file
 -@ [addr]       show section, symbol or import at addr
 -A              list sub-binaries and their arch-bits pairs
 -a [arch]       set arch (x86, arm, .. or <arch>_<bits>)
 -b [bits]       set bits (32, 64 ...)
 -B [addr]       override base address (pie bins)
 -c              list classes
 -cc             list classes in header format
 -C [fmt:C:D]    create [elf,mach0,pe] with Code and Data hexpairs (see -a)
 -d              show debug/dwarf information
 -D lang name    demangle symbol name (-D all for bin.demangle=true)
 -e              program entrypoint
 -ee             constructor/destructor entrypoints
 -E              globally exportable symbols
 -f [str]        select sub-bin named str
 -F [binfmt]     force to use that bin plugin (ignore header check)
 -g              same as -SMZIHVResizcld -SS -SSS -ee (show all info)
 -G [addr]       load address . offset to header
 -h              this help message
 -H              header fields
 -i              imports (symbols imported from libraries)
 -I              binary info
 -j              output in json
 -k [sdb-query]  run sdb query. for example: '*'
 -K [algo]       calculate checksums (md5, sha1, ..)
 -l              linked libraries
 -L [plugin]     list supported bin plugins or plugin details
 -m [addr]       show source line at addr
 -M              main (show address of main symbol)
 -n [str]        show section, symbol or import named str
 -N [min:max]    force min:max number of chars per string (see -z and -zz)
 -o [str]        output file/folder for write operations (out by default)
 -O [str]        write/extract operations (-O help)
 -p              show always physical addresses
 -P              show debug/pdb information
 -PP             download pdb file for binary
 -q              be quiet, just show fewer data
 -qq             show less info (no offset/size for -z for ex.)
 -Q              show load address used by dlopen (non-aslr libs)
 -r              radare output
 -R              relocations
 -s              symbols
 -S              sections
 -SS             segments
 -SSS            sections mapping to segments
 -t              display file hashes
 -T              display file signature
 -u              unfiltered (no rename duplicated symbols/sections)
```

```
  -U               resoUrces
  -v               display version and quit
  -V               show binary version information
  -w               display try/catch blocks
  -x               extract bins contained in file
  -X [fmt] [f] .. package in fat or zip the given files and bins contained in file
  -z               strings (from data section)
  -zz              strings (from raw bins [e bin.str.raw=1])
  -zzz             dump raw strings to stdout (for huge files)
  -Z               guess size of binary program
 Environment:
  R2_NOPLUGINS:      1|0|               # do not load shared plugins (speedup loading)
  RABIN2_CHARSET:    e cfg.charset      # set default value charset for -z strings
  RABIN2_DEBASE64:   e bin.str.debase64 # try to debase64 all strings
  RABIN2_DEMANGLE=0:e bin.demangle      # do not demangle symbols
  RABIN2_DMNGLRCMD:  e bin.demanglercmd # try to purge false positives
  RABIN2_LANG:       e bin.lang         # assume lang for demangling
  RABIN2_MAXSTRBUF:  e bin.str.maxbuf   # specify maximum buffer size
  RABIN2_PDBSERVER:  e pdb.server       # use alternative PDB server
  RABIN2_PREFIX:     e bin.prefix       # prefix symbols/sections/relocs with a specific
 string
  RABIN2_STRFILTER:  e bin.str.filter   # r2 -qc 'e bin.str.filter=??' -
  RABIN2_STRPURGE:   e bin.str.purge    # try to purge false positives
  RABIN2_SYMSTORE:   e pdb.symstore     # path to downstream symbol store
  RABIN2_SWIFTLIB:   1|0|               # load Swift libsto demangle (default: true)
  RABIN2_VERBOSE:    e bin.verbose      # show debugging messages from the parser
```

- 注： `rabin2 --help` 只能查看到精简的参数，没有参数含义介绍

  ```
  ➜  ~ rabin2 --help
  Usage: rabin2 [-AcdeEghHiIjlLMqrRsSUvVxzZ] [-@ at] [-a arch] [-b bits] [-B addr]
                [-C F:C:D] [-f str] [-m addr] [-n str] [-N m:M] [-P[-P] pdb]
                [-o str] [-O str] [-k query] [-D lang mangledsymbol] file
  ```

rabin2的man手册

```
RABIN2(1)                                  BSD General Commands Manual
                     RABIN2(1)

NAME
     RABIN2 — Binary program info extractor

SYNOPSIS
     rabin2 [-AceghHiIsSMzlpRrLxvhqQV] [-a arch] [-b bits] [-B addr] [-C fmt:C:[D]] [-D
 lang sym|-] [-f subbin]
            [-k query] [-K algo] [-O binop] [-o str] [-m addr] [-@ addr] [-n str] [-X f
mt file ...] file

DESCRIPTION
     This program allows you to get information about ELF/PE/MZ and CLASS files in a si
mple way.

     All those commandline flags are also available under the i command in radare2. Typ
e i? for help.

     -@ addr     Show information (symbol, section, import) of the given address

     -A          List sub-binaries and their associated arch-bits pairs

     -a arch     Set arch (x86, arm, .. accepts underscore for bits x86_32)

     -b bits     Set bits (32, 64, ...)

     -B addr     Override baddr

     -c          List classes

     -cc         List classes in header format

     -C [fmt:C[:D]]
                 Create [elf,mach0,pe] for arm and x86-32/64 tiny binaries where 'C' is
 an hexpair list of the code
                 bytes and ':D' is an optional concatenation to describe the bytes for
the data section.

     -d          Show debug/dwarf information

     -D lang symbolname|-
                 Demangle symbol name (or - to read from stdin) for lang (cxx, swift, j
ava, cxx, ..)

     -e          Show entrypoints for disk and on-memory

     -ee         Show constructor/destructors (extended entrypoints)

     -f subbin   Select sub-binary architecture. Useful for fat-mach0 binaries
```

```
       -F binfmt    Force to use that bin plugin (ignore header check)

       -g           Show all possible information

       -G addr      Load address . offset to header

       -h           Show usage help message.

       -H           Show header fields (see ih command in r2)

       -I           Show binary info (iI in r2)

       -i           Show imports (symbols imported from libraries) (ii)

       -j           Output in json

       -k query     Perform SDB query on loaded file

       -K algo      Select a rahash2 checksum algorithm to be performed on sections listin
g (and maybe others in the
                    future) i.e 'rabin2 -K md5 -S /bin/ls'

       -l           List linked libraries to the binary

       -L           List supported bin plugins

       -M           Show address of 'main' symbol

       -m addr      Show source line reference from a given address

       -N minlen:maxlen
                    Force minimum and maximum number of chars per string (see -z and -zz).
if (strlen>minlen && (!maxlen
                    || strlen<=maxlen))

       -n str       Show information (symbol, section, import) at string offset

       -o str       Output file/folder for write operations (out by default)

       -O binop     Perform binary operation on target binary (dump, resize, change sectio
ns, ...) see '-O help' for
                    more information

       -p           Disable VA. Show physical addresses

       -P           Show debug/pdb information

       -PP          Download pdb file for binary

       -q           Be quiet, just show fewer data

       -qq          Show less info (no offset/size for -z for ex.)

       -Q           Show load address used by dlopen (non-aslr libs)

       -r           Show output in radare format
```

```
     -R           Show realocations

     -s           Show exported symbols

     -S           Show sections

     -u           Unfiltered (no rename duplicated symbols/sections)

     -v           Show version information

     -V           Show binary version information

     -x           Extract all sub binaries from a fat binary (f.ex: fatmach0)

     -X format file ...
                  Package a fat or zip containing all the files passed (fat, zip)

     -z           Show strings inside .data section (like gnu strings does)

     -Z           Guess size of binary program

     -zz          Shows strings from raw bins

     -zzz         Dump raw strings to stdout (for huge files)

ENVIRONMENT
     RABIN2_LANG same as r2 -e bin.lang for rabin2

     RABIN2_DEMANGLE demangle symbols

     RABIN2_MAXSTRBUF same as r2 -e bin.maxstrbuf for rabin2

     RABIN2_DEBASE64 try to decode all strings as base64 if possible

     RABIN2_STRFILTER same as r2 -e bin.strfilter for rabin2

     RABIN2_STRPURGE same as r2 -e bin.strpurge for rabin2

EXAMPLES
     List symbols of a program

       $ rabin2 -s a.out

     Get offset of symbol

       $ rabin2 -n _main a.out

     Get entrypoint

       $ rabin2 -e a.out

     Load symbols and imports from radare2

       $ r2 -n /bin/ls
       [0x00000000]> . !rabin2 -prsi $FILE
```

```
SEE ALSO
     rahash2(1), rafind2(1), radare2(1), radiff2(1), rasm2(1), rax2(1), rsc2(1), ragg2(1
), rarun2(1),

AUTHORS
     Written by pancake <pancake@nopcode.org>.

                                                     Sep 29, 2016
```

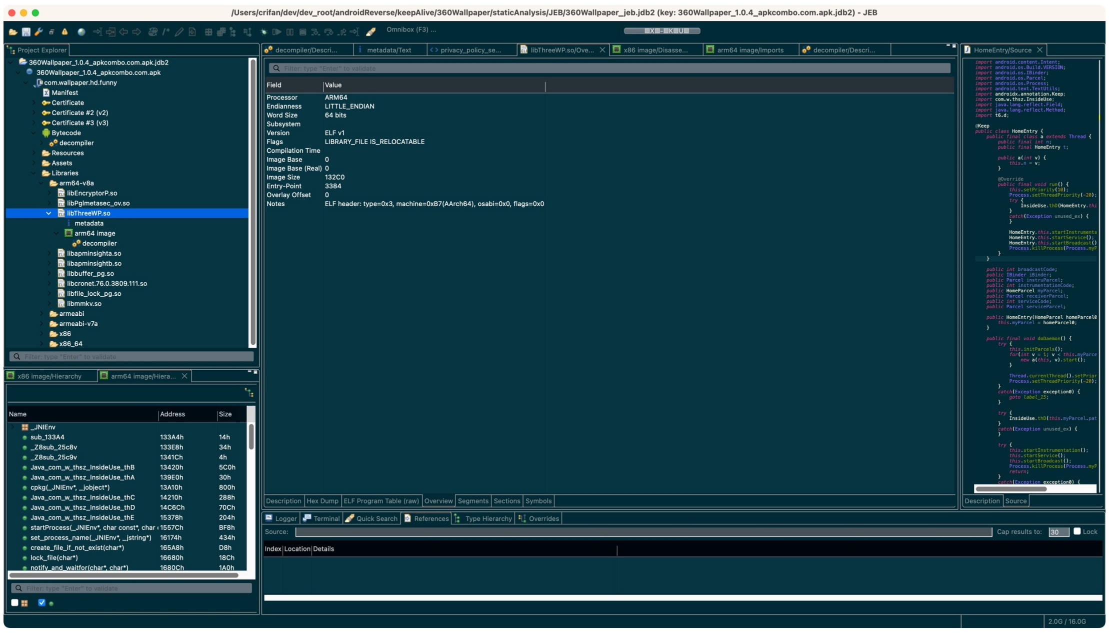
Android专用

JEB

JEB是个安卓逆向的利器。

JEB中也集成了，用于解析ELF格式的so库文件的功能。

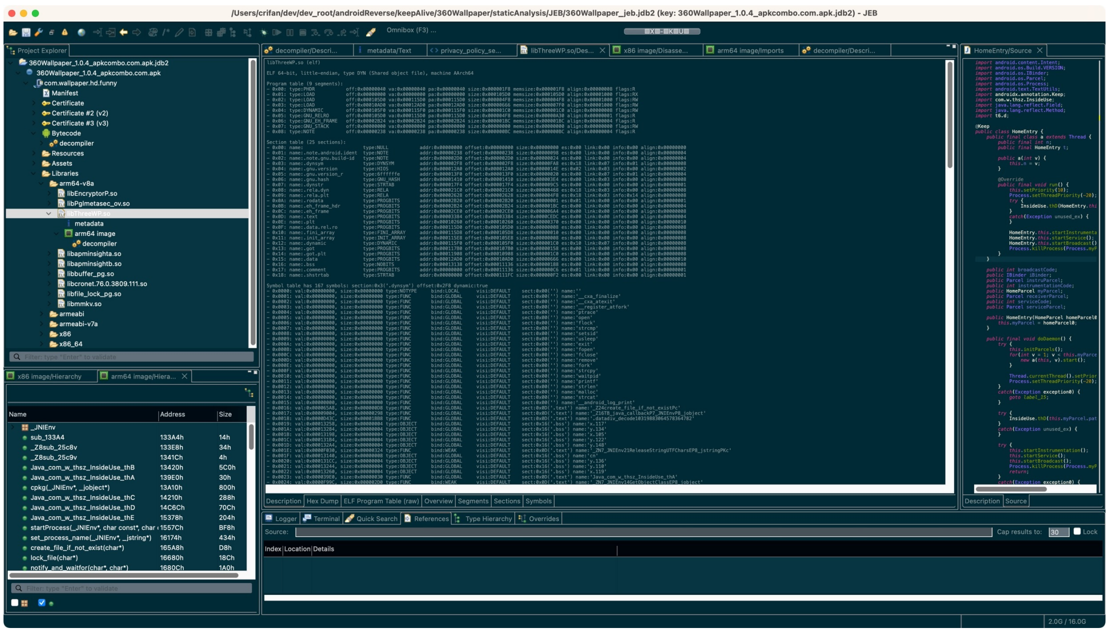
此处列出，JEB解析ELF格式的so库文件的相关内容：

- ELF的so库文件
  - Overview
    - 

/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

Field	Value
Processor	ARM64
Endianness	LITTLE_ENDIAN
Word Size	64 bits
Subsystem	
Version	ELF v1
Flags	LIBRARY_FILE IS_RELOCATABLE
Compilation Time	
Image Base	0
Image Base (Real)	0
Image Size	132C0
Entry-Point	3384
Overlay Offset	0
Notes	ELF header: type=0x3, machine=0xB7(AArch64), osabi=0x0, flags=0x0

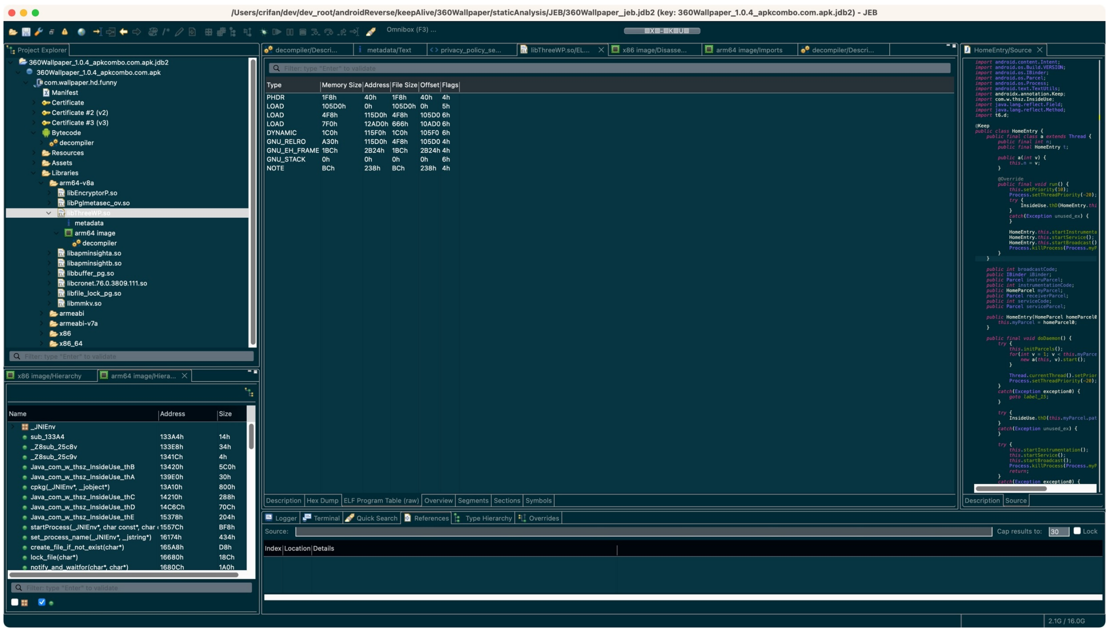
Name	Address	Size
_JNIEnv		
sub_133A4	133A4h	14h
_Z8sub_25c8v	133E8h	34h
_Z8sub_25c9v	1341Ch	4h
Java_com_w_thsz_InsideUse_thB	13420h	5C0h
Java_com_w_thsz_InsideUse_thA	139E0h	30h
cpkg(_JNIEnv*, _jobject*)	13A10h	800h
Java_com_w_thsz_InsideUse_thC	14210h	288h
Java_com_w_thsz_InsideUse_thD	14C6Ch	70Ch
Java_com_w_thsz_InsideUse_thE	15378h	204h
startProcess(_JNIEnv*, char const*, char	1557Ch	BF8h
set_process_name(_JNIEnv*, _jstring*)	16174h	434h
create_file_if_not_exist(char*)	165A8h	D8h
lock_file(char*)	16680h	18Ch
notify_and_waitfor(char*, char*)	1680Ch	1A0h

  - Description
    - 

/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

```
libThreeWP.so (elf)

ELF 64-bit, little-endian, type DYN (Shared object file), machine AArch64

Program table (9 segments):
- 0x00: type:PHDR         off:0x00000040 va:0x00000040 pa:0x00000040 size:0x000001F8 memsize:0x000001F8 align:0x00000008 flags:R
- 0x01: type:LOAD         off:0x00000000 va:0x00000000 pa:0x00000000 size:0x000105D0 memsize:0x000105D0 align:0x00001000 flags:RX
- 0x02: type:LOAD         off:0x000105D0 va:0x000115D0 pa:0x000115D0 size:0x000004F8 memsize:0x000004F8 align:0x00001000 flags:RW
- 0x03: type:LOAD         off:0x00010AD0 va:0x00012AD0 pa:0x00012AD0 size:0x00000666 memsize:0x000007F0 align:0x00001000 flags:RW
- 0x04: type:DYNAMIC      off:0x000105F0 va:0x000115F0 pa:0x000115F0 size:0x000001C0 memsize:0x000001C0 align:0x00000008 flags:RW
- 0x05: type:GNU_RELRO    off:0x000105D0 va:0x000115D0 pa:0x000115D0 size:0x000004F8 memsize:0x00000A30 align:0x00000001 flags:R
- 0x06: type:GNU_EH_FRAME off:0x00002B24 va:0x00002B24 pa:0x00002B24 size:0x000001BC memsize:0x000001BC align:0x00000004 flags:R
- 0x07: type:GNU_STACK    off:0x00000000 va:0x00000000 pa:0x00000000 size:0x00000000 memsize:0x00000000 align:0x00000000 flags:RW
- 0x08: type:NOTE         off:0x00000238 va:0x00000238 pa:0x00000238 size:0x000000BC memsize:0x000000BC align:0x00000004 flags:R

Section table (25 sections):
- 0x00: name:                    type:NULL          addr:0x00000000 offset:0x00000000 size:0x00000000 es:0x00 link:0x00 info:0x00 align:0x00000000
- 0x01: name:.note.android.ident type:NOTE          addr:0x00000238 offset:0x00000238 size:0x00000098 es:0x00 link:0x00 info:0x00 align:0x00000004
- 0x02: name:.note.gnu.build-id  type:NOTE          addr:0x000002D0 offset:0x000002D0 size:0x00000024 es:0x00 link:0x00 info:0x00 align:0x00000004
- 0x03: name:.dynsym             type:DYNSYM        addr:0x000002F8 offset:0x000002F8 size:0x00000FA8 es:0x18 link:0x07 info:0x01 align:0x00000008
- 0x04: name:.gnu.version        type:HIOS          addr:0x000012A0 offset:0x000012A0 size:0x0000014E es:0x02 link:0x03 info:0x00 align:0x00000002
- 0x05: name:.gnu.version_r      type:6ffffffe      addr:0x000013F0 offset:0x000013F0 size:0x00000020 es:0x00 link:0x07 info:0x01 align:0x00000004
- 0x06: name:.gnu.hash           type:GNU_HASH      addr:0x00001410 offset:0x00001410 size:0x000003E4 es:0x00 link:0x03 info:0x00 align:0x00000008
- 0x07: name:.dynstr             type:STRTAB        addr:0x000017F4 offset:0x000017F4 size:0x000009C5 es:0x00 link:0x00 info:0x00 align:0x00000001
- 0x08: name:.rela.dyn           type:RELA          addr:0x000021C0 offset:0x000021C0 size:0x00000468 es:0x18 link:0x03 info:0x00 align:0x00000008
- 0x09: name:.rela.plt           type:RELA          addr:0x00002628 offset:0x00002628 size:0x000004F8 es:0x18 link:0x03 info:0x14 align:0x00000008
- 0x0A: name:.rodata             type:PROGBITS      addr:0x00002B20 offset:0x00002B20 size:0x00000001 es:0x01 link:0x00 info:0x00 align:0x00000001
- 0x0B: name:.eh_frame_hdr       type:PROGBITS      addr:0x00002B24 offset:0x00002B24 size:0x000001BC es:0x00 link:0x00 info:0x00 align:0x00000004
- 0x0C: name:.eh_frame           type:PROGBITS      addr:0x00002CE0 offset:0x00002CE0 size:0x000006A4 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x0D: name:.text               type:PROGBITS      addr:0x00003384 offset:0x00003384 size:0x0000CEDC es:0x00 link:0x00 info:0x00 align:0x00000004
- 0x0E: name:.plt                type:PROGBITS      addr:0x00010260 offset:0x00010260 size:0x00000370 es:0x00 link:0x00 info:0x00 align:0x00000010
- 0x0F: name:.data.rel.ro        type:PROGBITS      addr:0x000115D0 offset:0x000105D0 size:0x00000008 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x10: name:.fini_array         type:FINI_ARRAY    addr:0x000115D8 offset:0x000105D8 size:0x00000010 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x11: name:.init_array         type:INIT_ARRAY    addr:0x000115E8 offset:0x000105E8 size:0x00000008 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x12: name:.dynamic            type:DYNAMIC       addr:0x000115F0 offset:0x000105F0 size:0x000001C0 es:0x10 link:0x07 info:0x00 align:0x00000008
- 0x13: name:.got                type:PROGBITS      addr:0x000117B0 offset:0x000107B0 size:0x00000158 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x14: name:.got.plt            type:PROGBITS      addr:0x00011908 offset:0x00010908 size:0x000001C0 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x15: name:.data               type:PROGBITS      addr:0x00012AD0 offset:0x00010AD0 size:0x00000666 es:0x00 link:0x00 info:0x00 align:0x00000010
- 0x16: name:.bss                type:NOBITS        addr:0x00013138 offset:0x00011136 size:0x00000188 es:0x00 link:0x00 info:0x00 align:0x00000008
- 0x17: name:.comment            type:PROGBITS      addr:0x00000000 offset:0x00011136 size:0x000000C6 es:0x01 link:0x00 info:0x00 align:0x00000001
- 0x18: name:.shstrtab           type:STRTAB        addr:0x00000000 offset:0x000111FC size:0x000000F2 es:0x00 link:0x00 info:0x00 align:0x00000001

Symbol table has 167 symbols: section:0x3('.dynsym') offset:0x2F8 dynamic:true
- 0x0000: val:0x00000000, size:0x00000000 type:NOTYPE    bind:LOCAL     visi:DEFAULT    sect:0x00('') name:''
- 0x0001: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'__cxa_finalize'
- 0x0002: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'__cxa_atexit'
- 0x0003: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'__register_atfork'
- 0x0004: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'ptrace'
- 0x0005: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'open'
- 0x0006: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'flock'
- 0x0007: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'strcmp'
- 0x0008: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'setsid'
- 0x0009: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'usleep'
- 0x000A: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'exit'
- 0x000B: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'fopen'
- 0x000C: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'fclose'
- 0x000D: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'remove'
- 0x000E: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'fork'
- 0x000F: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'strcpy'
- 0x0010: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'waitpid'
- 0x0011: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'printf'
- 0x0012: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'strlen'
- 0x0013: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'malloc'
- 0x0014: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'strcat'
- 0x0015: val:0x00000000, size:0x00000000 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x00('') name:'__android_log_print'
- 0x0016: val:0x000065A8, size:0x000000D8 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x0D('.text') name:'_Z24create_file_if_not_existPc'
- 0x0017: val:0x00009004, size:0x00000298 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x0D('.text') name:'_Z16TB_java_callbackP7_JNIEnvP8_jobject'
- 0x0018: val:0x0000D43C, size:0x00001BB8 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x0D('.text') name:'.datadiv_decode10319883064578364782'
- 0x0019: val:0x00013258, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'x.117'
- 0x001A: val:0x00013284, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'y.134'
- 0x001B: val:0x00013198, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'x.105'
- 0x001C: val:0x000131B4, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'y.122'
- 0x001D: val:0x000132A4, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'y.148'
- 0x001E: val:0x0000F030, size:0x00000324 type:FUNC      bind:WEAK      visi:DEFAULT    sect:0x0D('.text') name:'_ZN7_JNIEnv21ReleaseStringUTFCharsEP8_jstringPKc'
- 0x001F: val:0x00013140, size:0x00000008 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'cn'
- 0x0020: val:0x000131CC, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'y.136'
- 0x0021: val:0x00013244, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'y.110'
- 0x0022: val:0x00013260, size:0x00000004 type:OBJECT    bind:GLOBAL    visi:DEFAULT    sect:0x16('.bss') name:'x.119'
- 0x0023: val:0x000039E0, size:0x00000030 type:FUNC      bind:GLOBAL    visi:DEFAULT    sect:0x0D('.text') name:'Java_com_w_thsz_InsideUse_thA'
- 0x0024: val:0x0000F99C, size:0x000002D0 type:FUNC      bind:WEAK      visi:DEFAULT    sect:0x0D('.text') name:'_ZN7_JNIEnv14GetObjectClassEP8_jobject'
```

  - ELF Program Table (raw)

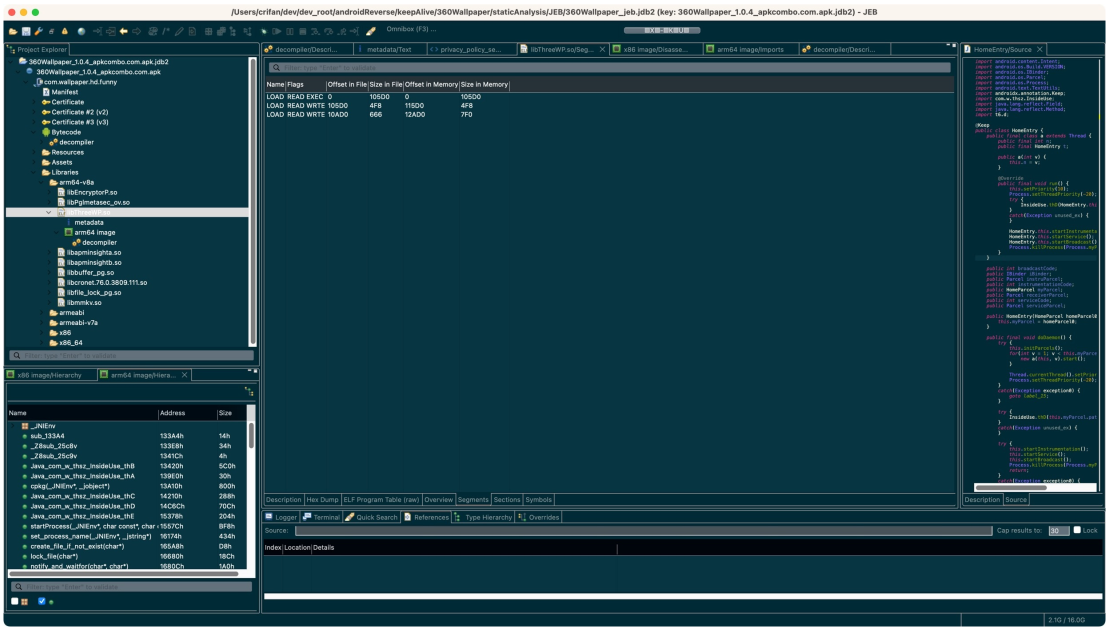


/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

Type	Memory Size	Address	File Size	Offset	Flags
PHDR	1F8h	40h	1F8h	40h	4h
LOAD	105D0h	0h	105D0h	0h	5h
LOAD	4F8h	115D0h	4F8h	105D0	6h
LOAD	7F0h	12AD0h	666h	10AD0	6h
DYNAMIC	1C0h	115F0h	1C0h	105F0	6h
GNU_RELRO	A30h	115D0h	4F8h	105D0	4h
GNU_EH_FRAME	1BCh	2B24h	1BCh	2B24h	4h
GNU_STACK	0h	0h	0h	0h	6h
NOTE	BCh	238h	BCh	238h	4h

Name	Address	Size
_JNIEnv		
sub_133A4	133A4h	14h
_Z8sub_25c8v	133E8h	34h
_Z8sub_25c9v	1341Ch	4h
Java_com_w_thsz_InsideUse_thB	13420h	5C0h
Java_com_w_thsz_InsideUse_thA	139E0h	30h
cpkg(_JNIEnv*, _jobject*)	13A10h	800h
Java_com_w_thsz_InsideUse_thC	14210h	288h
Java_com_w_thsz_InsideUse_thD	14C6Ch	70Ch
Java_com_w_thsz_InsideUse_thE	15378h	204h
startProcess(_JNIEnv*, char const*, char	1557Ch	BF8h
set_process_name(_JNIEnv*, _jstring*)	16174h	434h
create_file_if_not_exist(char*)	165A8h	D8h
lock_file(char*)	16680h	18Ch
notify_and_waitfor(char*, char*)	1680Ch	1A0h

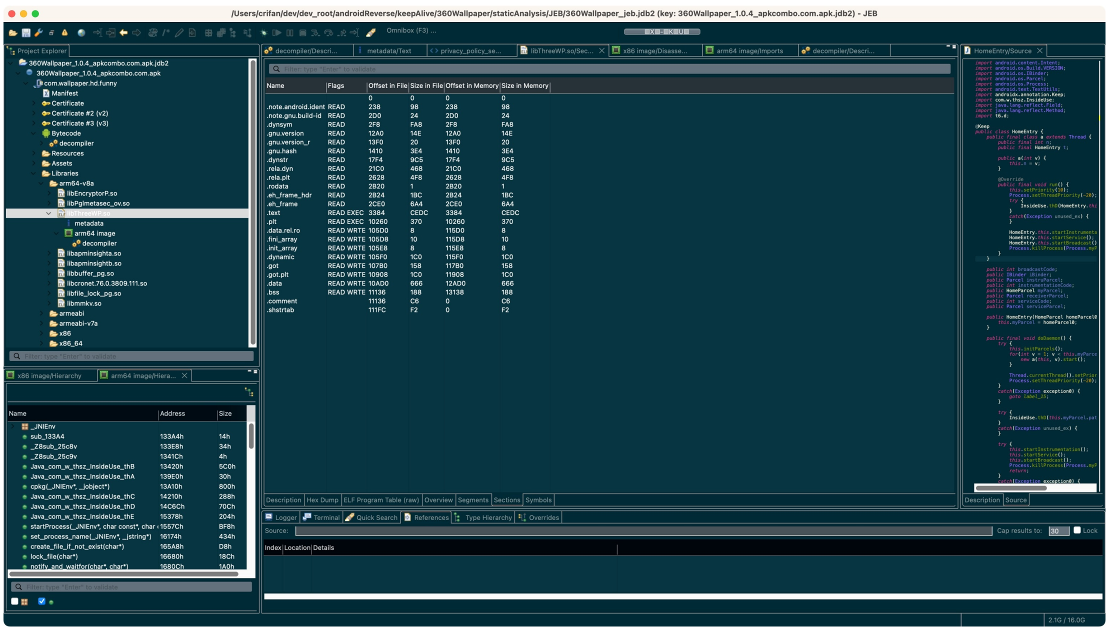
- Segements



/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

Name	Flags	Offset in File	Size in File	Offset in Memory	Size in Memory
LOAD	READ EXEC	0	105D0	0	105D0
LOAD	READ WRTE	105D0	4F8	115D0	4F8
LOAD	READ WRTE	10AD0	666	12AD0	7F0

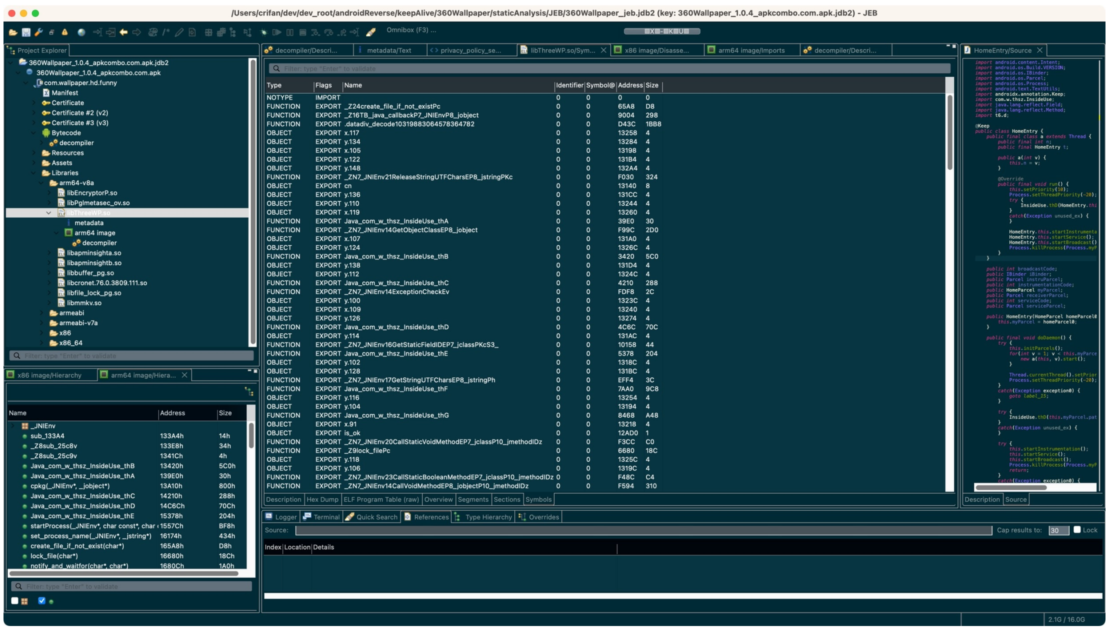
- Sections

- 

/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

Name	Flags	Offset in File	Size in File	Offset in Memory	Size in Memory
		0	0	0	0
.note.android.ident	READ	238	98	238	98
.note.gnu.build-id	READ	2D0	24	2D0	24
.dynsym	READ	2F8	FA8	2F8	FA8
.gnu.version	READ	12A0	14E	12A0	14E
.gnu.version_r	READ	13F0	20	13F0	20
.gnu.hash	READ	1410	3E4	1410	3E4
.dynstr	READ	17F4	9C5	17F4	9C5
.rela.dyn	READ	21C0	468	21C0	468
.rela.plt	READ	2628	4F8	2628	4F8
.rodata	READ	2B20	1	2B20	1
.eh_frame_hdr	READ	2B24	1BC	2B24	1BC
.eh_frame	READ	2CE0	6A4	2CE0	6A4
.text	READ EXEC	3384	CEDC	3384	CEDC
.plt	READ EXEC	10260	370	10260	370
.data.rel.ro	READ WRTE	105D0	8	115D0	8
.fini_array	READ WRTE	105D8	10	115D8	10
.init_array	READ WRTE	105E8	8	115E8	8
.dynamic	READ WRTE	105F0	1C0	115F0	1C0
.got	READ WRTE	107B0	158	117B0	158
.got.plt	READ WRTE	10908	1C0	11908	1C0
.data	READ WRTE	10AD0	666	12AD0	666
.bss	READ WRTE	11136	188	13138	188
.comment		11136	C6	0	C6
.shstrtab		111FC	F2	0	F2

Name	Address	Size
_JNIEnv		
sub_133A4	133A4h	14h
_Z8sub_25c8v	133E8h	34h
_Z8sub_25c9v	1341Ch	4h
Java_com_w_thsz_InsideUse_thB	13420h	5C0h
Java_com_w_thsz_InsideUse_thA	139E0h	30h
cpkg(_JNIEnv*, _jobject*)	13A10h	800h
Java_com_w_thsz_InsideUse_thC	14210h	288h
Java_com_w_thsz_InsideUse_thD	14C6Ch	70Ch
Java_com_w_thsz_InsideUse_thE	15378h	204h
startProcess(_JNIEnv*, char const*, char (	1557Ch	BF8h
set_process_name(_JNIEnv*, _jstring*)	16174h	434h
create_file_if_not_exist(char*)	165A8h	D8h
lock_file(char*)	16680h	18Ch
notify_and_waitfor(char*, char*)	1680Ch	1A0h

- Symbols
  - 

/Users/crifan/dev/dev_root/androidReverse/keepAlive/360Wallpaper/staticAnalysis/JEB/360Wallpaper_jeb.jdb2 (key: 360Wallpaper_1.0.4_apkcombo.com.apk.jdb2) - JEB

Type	Flags	Name	Identifier	Symbol@	Address	Size
NOTYPE	IMPORT		0	0	0	0
FUNCTION	EXPORT	_Z24create_file_if_not_existPc	0	0	65A8	D8
FUNCTION	EXPORT	_Z16TB_java_callbackP7_JNIEnvP8_jobject	0	0	9004	298
FUNCTION	EXPORT	.datadiv_decode10319883064578364782	0	0	D43C	1BB8
OBJECT	EXPORT	x.117	0	0	13258	4
OBJECT	EXPORT	y.134	0	0	13284	4
OBJECT	EXPORT	x.105	0	0	13198	4
OBJECT	EXPORT	y.122	0	0	131B4	4
OBJECT	EXPORT	y.148	0	0	132A4	4
FUNCTION	EXPORT	_ZN7_JNIEnv21ReleaseStringUTFCharsEP8_jstringPKc	0	0	F030	324
OBJECT	EXPORT	cn	0	0	13140	8
OBJECT	EXPORT	y.136	0	0	131CC	4
OBJECT	EXPORT	y.110	0	0	13244	4
OBJECT	EXPORT	x.119	0	0	13260	4
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thA	0	0	39E0	30
FUNCTION	EXPORT	_ZN7_JNIEnv14GetObjectClassEP8_jobject	0	0	F99C	2D0
OBJECT	EXPORT	x.107	0	0	131A0	4
OBJECT	EXPORT	y.124	0	0	1326C	4
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thB	0	0	3420	5C0
OBJECT	EXPORT	y.138	0	0	131D4	4
OBJECT	EXPORT	y.112	0	0	1324C	4
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thC	0	0	4210	288
FUNCTION	EXPORT	_ZN7_JNIEnv14ExceptionCheckEv	0	0	FDF8	2C
OBJECT	EXPORT	y.100	0	0	1323C	4
OBJECT	EXPORT	x.109	0	0	13240	4
OBJECT	EXPORT	y.126	0	0	13274	4
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thD	0	0	4C6C	70C
OBJECT	EXPORT	y.114	0	0	131AC	4
FUNCTION	EXPORT	_ZN7_JNIEnv16GetStaticFieldIDEP7_jclassPKcS3_	0	0	10158	44
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thE	0	0	5378	204
OBJECT	EXPORT	y.102	0	0	1318C	4
OBJECT	EXPORT	y.128	0	0	131BC	4
FUNCTION	EXPORT	_ZN7_JNIEnv17GetStringUTFCharsEP8_jstringPh	0	0	EFF4	3C
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thF	0	0	7AA0	9C8
OBJECT	EXPORT	y.116	0	0	13254	4
OBJECT	EXPORT	y.104	0	0	13194	4
FUNCTION	EXPORT	Java_com_w_thsz_InsideUse_thG	0	0	8468	A48
OBJECT	EXPORT	x.91	0	0	13218	4
OBJECT	EXPORT	is_ok	0	0	12AD0	1
FUNCTION	EXPORT	_ZN7_JNIEnv20CallStaticVoidMethodEP7_jclassP10_jmethodIDz	0	0	F3CC	C0
FUNCTION	EXPORT	_Z9lock_filePc	0	0	6680	18C
OBJECT	EXPORT	y.118	0	0	1325C	4
OBJECT	EXPORT	y.106	0	0	1319C	4
FUNCTION	EXPORT	_ZN7_JNIEnv23CallStaticBooleanMethodEP7_jclassP10_jmethodIDz	0	0	F48C	C4
FUNCTION	EXPORT	_ZN7_JNIEnv14CallVoidMethodEP8_jobjectP10_jmethodIDz	0	0	F594	310

Name	Address	Size
_JNIEnv		
sub_133A4	133A4h	14h
_Z8sub_25c8v	133E8h	34h
_Z8sub_25c9v	1341Ch	4h
Java_com_w_thsz_InsideUse_thB	13420h	5C0h
Java_com_w_thsz_InsideUse_thA	139E0h	30h
cpkg(_JNIEnv*, _jobject*)	13A10h	800h
Java_com_w_thsz_InsideUse_thC	14210h	288h
Java_com_w_thsz_InsideUse_thD	14C6Ch	70Ch
Java_com_w_thsz_InsideUse_thE	15378h	204h
startProcess(_JNIEnv*, char const*, char (	1557Ch	BF8h
set_process_name(_JNIEnv*, _jstring*)	16174h	434h
create_file_if_not_exist(char*)	165A8h	D8h
lock_file(char*)	16680h	18Ch
notify_and_waitfor(char*, char*)	1680Ch	1A0h

更多细节详见：解析so库文件 · 安卓逆向利器：JEB

解析修改

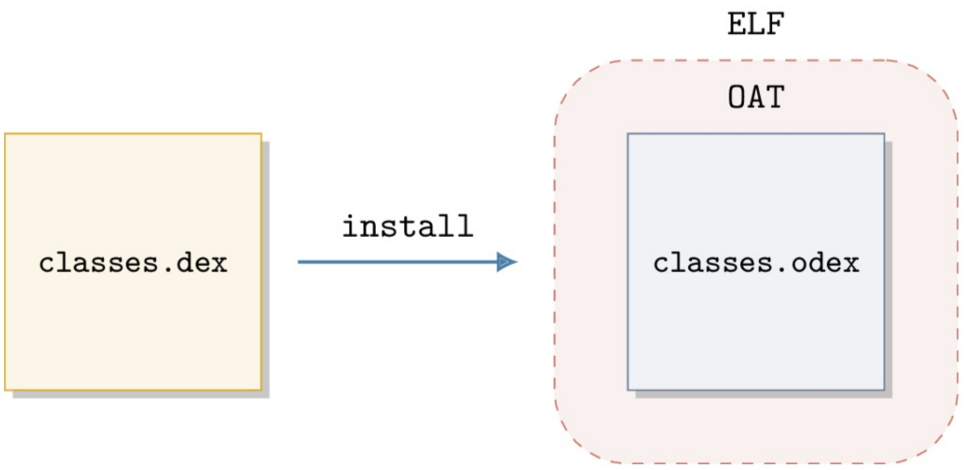
用于解析和修改ELF格式的工具有：

- LIEF

LIEF

- LIEF
    - 概述：用于查看解析和编辑修改（ `ELF` / `PE` / `MachO` / `Android` 等）各种通用的可执行文件格式的库
    - 详解
        - LIEF · 可执行文件格式
            - LIEF用法举例 · 可执行文件格式

Android中的ELF

- Android中的ELF格式
  - Android在通用的Linux中的ELF的基础上，进一步扩展
    - `.dex` / `.oat` : `ELF` + 扩展的section
    - `.dex` 被转换成 `.odex`
      - odex是外部是ELF头，内部包裹了个OAT格式
      - 

  - 解析Android的ELF格式
    - 详见
      - Android · 可执行文件格式

dex格式

dex文件就是ELF格式的：

```
$ file snet.dex
snet.dex: ELF 64-bit LSB shared object, ARM aarch64, version 1 (GNU/Linux), dynamically
 linked, stripped
```

- 注意
  - 不可轻信后缀
    - `.dex` 可能是 `DEX` 或 `OAT`
    - `.odex` 是 `OAT`
    - `.oat` 是 `OAT`

OAT格式

- 把java转成OAT的过程

。

附录

下面列出相关参考资料。

参考资料

- 【整理】ELF文件格式
- 【整理】ELF相关：.bss节
- 【已解决】Mac M2 Max中安装readelf
- 【记录】Mac中用readelf查看ELF的so库二进制文件信息
- 【记录】Mac中用objdump查看ELF的so库二进制文件信息
- 【记录】用rabin2查看ELF的so库文件信息
- 【未解决】安卓逆向：查看ELF的so库二进制信息的工具
- 【未解决】Mac中是否有readelf的GUI图形界面版本
- 【已解决】给已有libtacker.so去改动信息
- 【已解决】用LIEF去修改ELF的so中的部分信息
-
- 查看信息和导出字符串 · iOS逆向开发：静态分析 (crifan.org)
- 解析so库文件 · 安卓逆向利器：JEB
-
- 2007.14266.pdf (zzm7000.github.io)
- Executable and Linkable Format - Wikipedia
- elf(5) — Linux manual pages (courier-mta.org)
- ELF Header (sco.com)
- Special Sections (oracle.com)
- Executable and Linkable Format (ELF) (netmeister.org)
- s.eresi-project.org/inc/articles/elf-rtld.txt
- [原创]Android so(ELF)文件解析-Android安全-看雪-安全社区|安全招聘|kanxue.com
- 使用readelf和objdump解析目标文件 - 江召伟 - 博客园 (cnblogs.com)
- 13. readelf elf文件格式分析 — Linux Tools Quick Tutorial (linuxtools-rst.readthedocs.io)
- ELF文件 及 nm & readelf & objdump 使用与对比 - 简书 (jianshu.com)
- ELF文件分析之0 – 简介和分析工具 | Simple (cedar-renjun.github.io)
- linux下强大的ELF文件分析工具 -- readelf*elf*解析工具悟OO道的博客-CSDN博客
- 二进制分析工具 - 阿宅の小窝 (zaxtyson.cn)
- objdump(1) - Linux manual page (man7.org)
- readelf 和 objdump 例子详解及区别 （ELF文件说明）_objdump readelf_Hani_97的博客-CSDN博客
- Linux中objdump的使用 | Ivanzz (ivanzz1001.github.io)
- 常用的分析ELF文件的命令（readelf、objdump及od） - 王瓦斯的春天 - 博客园 (cnblogs.com)
- 14. objdump 二进制文件分析 — Linux Tools Quick Tutorial (linuxtools-rst.readthedocs.io)
- ELF文件 及 nm & readelf & objdump 使用与对比 - 简书 (jianshu.com)
- ELF for the ARM Architecture
- DWARF Debugging Information Format
- Hardened/GNU stack quickstart - Gentoo Wiki
- Dynamic Linking
- Program Header
- ELF Header
- RolandMcGrath/BuildID - Fedora Project Wiki
- Releases/FeatureBuildId - Fedora Project Wiki

- Airs – Ian Lance Taylor » Executable stack
- rabin2 - r2wiki
- Rabin2 - The Official Radare2 Book
- Linux 修改 ELF 解决 glibc 兼容性问题 (qq.com)
- ELF中可以被修改又不影响执行的区域-腾讯云开发者社区-腾讯云 (tencent.com)
-