Introduction to ELF

Contents

- What is ELF?
  - Magic number
- ELF Header

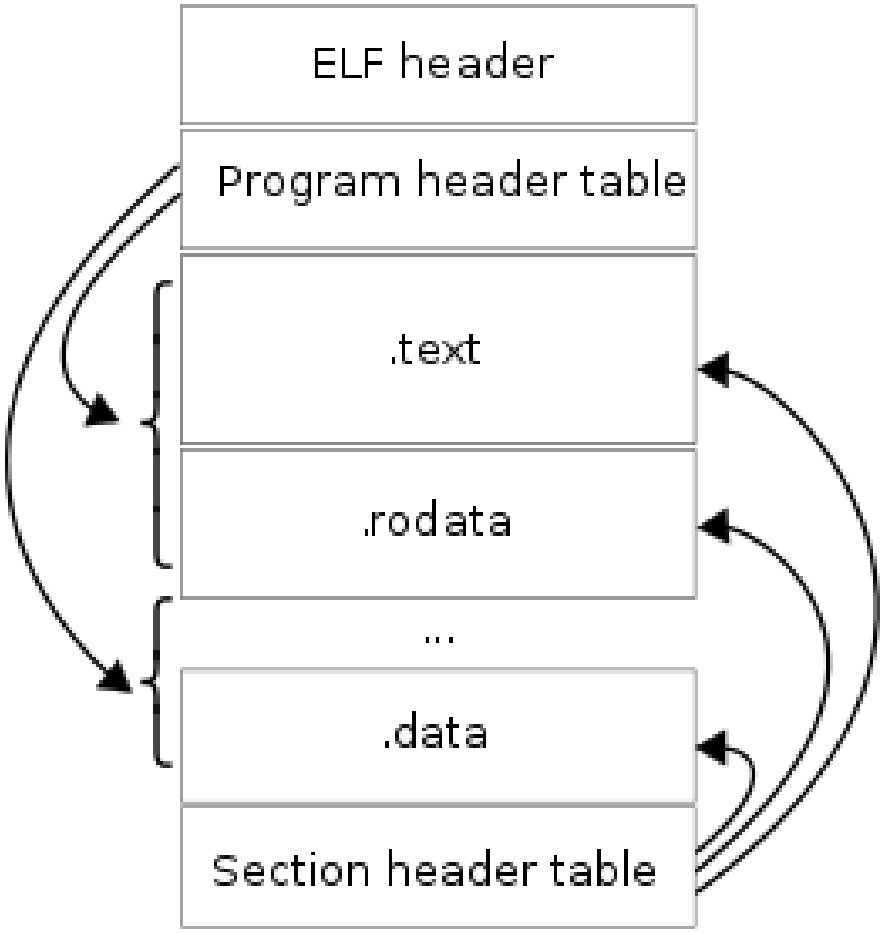
What is ELF?

- Executable and Linkable Format

- Common standard file format for
  - Executable file
  - Object code
  - Shared libraries
  - Core dump

What is ELF?

- Made up with of one ELF header, followed by file data
- `readelf` tool can read / display information about ELF files
- The ELF header is 52 bytes long for 32 bit, 64 bytes long for 64 bit addresses



(1)

What is ELF?

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ ls
foo1  foo1.c  foo2  foo2.c  output1  output2
hatchling@DESKTOP-23UPC6C:~/play$ readelf -h foo1
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:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x4003e0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          6384 (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:         28
  Section header string table index: 27
hatchling@DESKTOP-23UPC6C:~/play$
```

ELF header

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ readelf -l foo1

Elf file type is EXEC (Executable file)
Entry point 0x4003e0
There are 9 program headers, starting at offset 64

Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R      0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000006d0 0x00000000000006d0  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x0000000000000220 0x0000000000000228  RW     0x200000
  DYNAMIC        0x0000000000000e20 0x0000000000600e20 0x0000000000600e20
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000020 0x0000000000000020  R      0x4
  GNU_EH_FRAME   0x0000000000000594 0x0000000000400594 0x0000000000400594
                 0x000000000000003c 0x000000000000003c  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1
```

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ readelf -S foo1
There are 28 section headers, starting at offset 0x18f0:

Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .interp           PROGBITS         0000000000400238  00000238
       000000000000001c  0000000000000000   A       0     0     1
  [ 2] .note.ABI-tag     NOTE             0000000000400254  00000254
       0000000000000020  0000000000000000   A       0     0     4
  [ 3] .gnu.hash         GNU_HASH         0000000000400278  00000278
       000000000000001c  0000000000000000   A       4     0     8
  [ 4] .dynsym           DYNSYM           0000000000400298  00000298
       0000000000000060  0000000000000018   A       5     1     8
  [ 5] .dynstr           STRTAB           00000000004002f8  000002f8
       000000000000003f  0000000000000000   A       0     0     1
  [ 6] .gnu.version      VERSYM           0000000000400338  00000338
       0000000000000008  0000000000000002   A       4     0     2
  [ 7] .gnu.version_r    VERNEED          0000000000400340  00000340
       0000000000000020  0000000000000000   A       5     1     8
  [ 8] .rela.dyn         RELA             0000000000400360  00000360
       0000000000000030  0000000000000018   A       4     0     8
  [ 9] .rela.plt         RELA             0000000000400390  00000390
       0000000000000018  0000000000000018  AI       4    21     8
  [10] .init             PROGBITS         00000000004003a8  000003a8
       0000000000000017  0000000000000000  AX       0     0     4
  [11] .plt              PROGBITS         00000000004003c0  000003c0
       0000000000000020  0000000000000010  AX       0     0    16
  [12] .text             PROGBITS         00000000004003e0  000003e0
       0000000000000192  0000000000000000  AX       0     0    16
```

What is ELF?

Magic number

- Data used to identify the contents of a file
- In Unix-like system, filename extension is not mandatory
- Therefore, we use `file` command to read / interpret magic numbers
- Similar to MIME type

Magic number

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ ls
foo1  foo1.c  foo2  foo2.c  output1  output2
hatchling@DESKTOP-23UPC6C:~/play$ file foo1
foo1: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
for GNU/Linux 3.2.0, not stripped
hatchling@DESKTOP-23UPC6C:~/play$
```

What is ELF?

- Program header table
  - Tells the system how to create a process image:
  - A copy of the process at a given point in time
  - Files used to execute a program must have one

- Section header table
  - Contains information describing the file’s sections
  - Files used during linking must have one

ELF header

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ ls
foo1  foo1.c  foo2  foo2.c  output1  output2
hatchling@DESKTOP-23UPC6C:~/play$ readelf -h foo1
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:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x4003e0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          6384 (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:         28
  Section header string table index: 27
hatchling@DESKTOP-23UPC6C:~/play$
```

hatchling@DESKTOP-23UPC6C: ~/play

```
hatchling@DESKTOP-23UPC6C:~/play$ readelf -h fool
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:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x4003e0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          6384 (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:         28
  Section header string table index: 27
hatchling@DESKTOP-23UPC6C:~/play$
```

hatchling@DESKTOP-23UPC6C: ~/play

```
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 3e00 0100 0000 e003 4000 0000 0000  ..>.......@.....
00000020: 4000 0000 0000 0000 f018 0000 0000 0000  @...............
00000030: 0000 0000 4000 3800 0900 4000 1c00 1b00  ....@.8...@.....
00000040: 0600 0000 0400 0000 4000 0000 0000 0000  ........@.......
00000050: 4000 4000 0000 0000 4000 4000 0000 0000  @.@.....@.@.....
00000060: f801 0000 0000 0000 f801 0000 0000 0000  ................
00000070: 0800 0000 0000 0000 0300 0000 0400 0000  ................
00000080: 3802 0000 0000 0000 3802 4000 0000 0000  8.......8.@.....
00000090: 3802 4000 0000 0000 1c00 0000 0000 0000  8.@.............
000000a0: 1c00 0000 0000 0000 0100 0000 0000 0000  ................
000000b0: 0100 0000 0500 0000 0000 0000 0000 0000  ................
000000c0: 0000 4000 0000 0000 0000 4000 0000 0000  ..@.......@.....
000000d0: d006 0000 0000 0000 d006 0000 0000 0000  ................
000000e0: 0000 2000 0000 0000 0100 0000 0600 0000  .. .............
000000f0: 100e 0000 0000 0000 100e 6000 0000 0000  ..........`.....
00000100: 100e 6000 0000 0000 2002 0000 0000 0000  ..`..... .......
00000110: 2802 0000 0000 0000 0000 2000 0000 0000  (......... .....
00000120: 0200 0000 0600 0000 200e 0000 0000 0000  ........ .......
00000130: 200e 6000 0000 0000 200e 6000 0000 0000   .`..... .`.....
00000140: d001 0000 0000 0000 d001 0000 0000 0000  ................
00000150: 0800 0000 0000 0000 0400 0000 0400 0000  ................
00000160: 5402 0000 0000 0000 5402 4000 0000 0000  T.......T.@.....
00000170: 5402 4000 0000 0000 2000 0000 0000 0000  T.@..... .......
00000180: 2000 0000 0000 0000 0400 0000 0000 0000  ................
00000190: 50e5 7464 0400 0000 9405 0000 0000 0000  P.td............
000001a0: 9405 4000 0000 0000 9405 4000 0000 0000  ..@.......@.....
000001b0: 3c00 0000 0000 0000 3c00 0000 0000 0000  <.......<.......
000001c0: 0400 0000 0000 0000 51e5 7464 0600 0000  ........Q.td....
fool.x
```

ELF header

Overall

- ELF is common standard file format used in Unix-like system
- Made up of one ELF header, followed by file data
- Magic number is [0x7F, ‘E’(0x45), ‘L’(0x4C), ‘F’(0x46)]
- Difference between section vs. segment?

Citation

- https://refspecs.linuxfoundation.org/elf/elf.pdf

- (1) - https://en.wikipedia.org/wiki/File:Elf-layout--en.svg