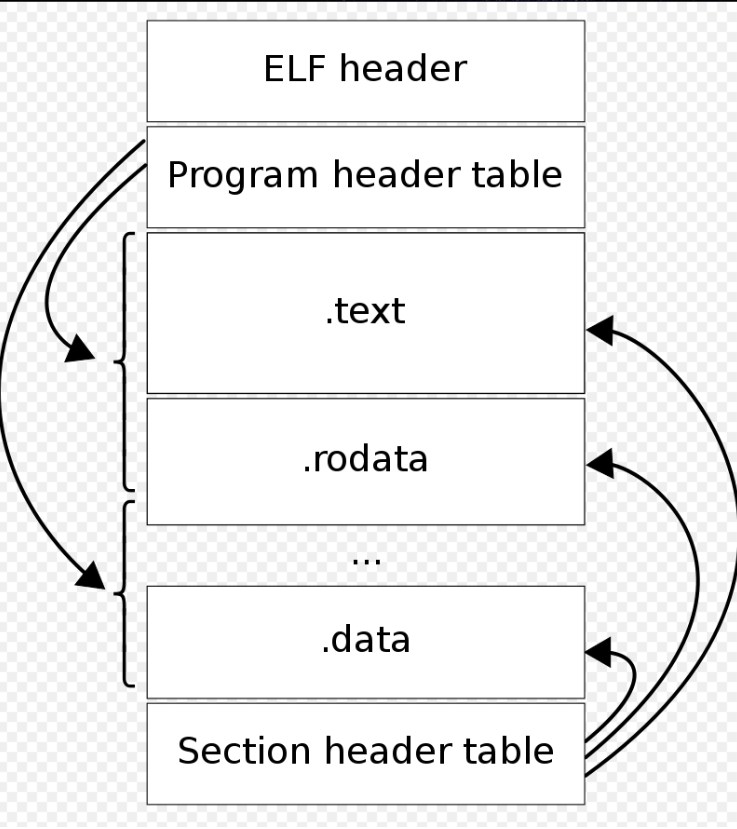
ELF

1. ELF là gì?

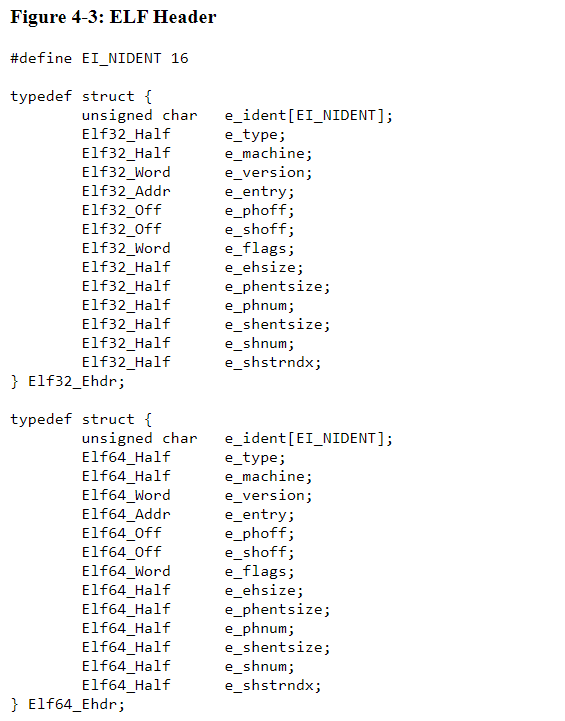
Elf viết tắt của Executable and Link Format – định dạng có thể thực thi và có thể liên kết

Cấu trúc của ELF



1. File header

Đây là phần đầu của file ELF có độ dài 52 byte (32bit) và 64 byte(64bit)



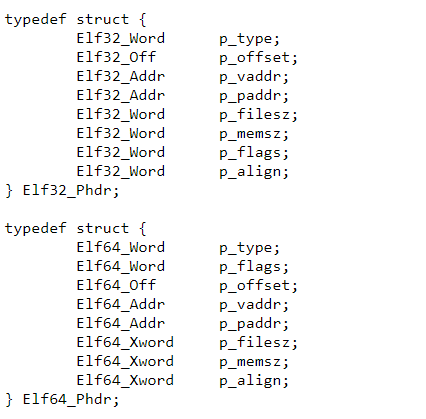
Ý nghĩa các trường

* e\_ident đánh dấu đây là tệp đối tượng cung cấp dữ liệu để giải mã
* e\_ident[EI\_MAG0] -> e\_ident[EI\_MAG3] :nếu là file ELF sẽ có giá trị (0x 7F 45 4C 46)
* e\_ident[EI\_CLASS] : có giá trị là 1 nếu đây là file 32bit là 2 nếu 64 bit
* e\_ident[EI\_DATA] : có giá trị 1 với với kiểu dữ liệu đầu nhỏ (little endianness), 2 nếu kiểu dữ liệu đầu to (big endianness)
* e\_ident[EI\_VERSION]: có giá trị 1 cho phiên bản gốc và phiên bản hiện tại của ELF
* e\_ident[EI\_OSABI] : xác định hệ điều hành
* e\_ident[EI\_ABIVERSION]: xác định phien bản ABI
* e\_ident[EI\_PAD] : đánh dấu các byte ko sử dụng trong e\_ident và có giá trị là 0
* e\_type xác định loại tệp đối tượng
* e\_machine: xác định cấu trúc máy tính
* e\_version: có giá trị 1 cho phiên bản gốc của ELF
* e\_entry : cung cấp địa chỉ ảo mà hệ thống chuyển quyền điều khiển
* e\_phoff : cung cấp địa chỉ bắt đầu program header
* e\_shoff : cung cấp địa chỉ bắt đầu section header
* e\_flags :cờ
* e\_ehsize: kích thước của ELF header 64 byte với 64 bit và 52 byte với 32 bit
* e\_phentsize: chứa kích thước program header
* e\_phnum: chứa số lượng mục trong program header
* e\_shentsize: chứa kích thước section header
* e\_shnum: chứa số lượng mục trong section header
* e\_shstrndx: cung cấp section chứa tên các section

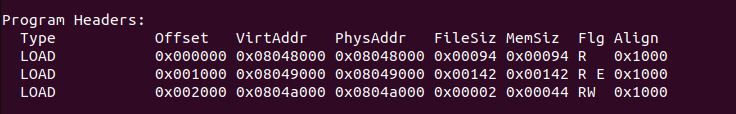
1. Program Header

Program header table là 1 mảng cấu trúc, mỗi cấu trúc mô tả một phân đoạn (segment) hoặc thông tin khác mà hệ thống cần để chuẩn bị chương trình thực thi

Trong ELF 32bit và ELF 64 bit có các trường giống nhau nhưng bố cục khác nhau (ở p\_glags) và kích thước nhau

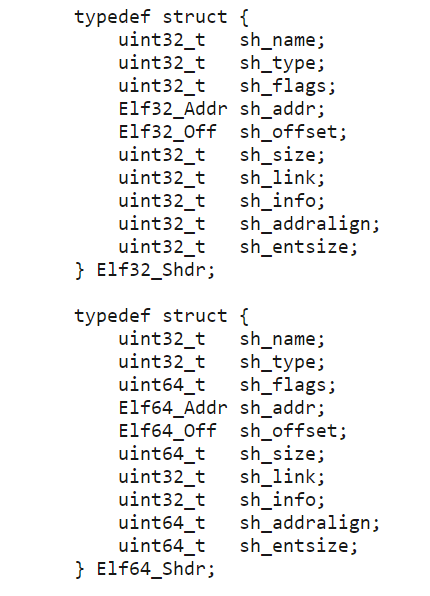


* p\_type : xác định loại phân đoạn
* p\_flags: xác định cờ của phân đoạn
* p\_offset: cung cấp địa chỉ phân đoạn
* p\_vaddr: địa chỉ ảo của phân đoạn trong bộ nhớ
* p\_paddr: địa chỉ thực của phân đoạn
* p\_filesz: kích thước phân đoạn tính bằng byte trong file image
* p\_memsz : kích thước phân đoạn tính bằng byte trong memory
* p\_align: xác định căn chỉnh hay không. Nếu 0 và 1 là không cần căn chỉnh.



1. Section header

Section header table chứa tất cả thông tin cần thiết để xác định vị trí từng phần(section) của tệp

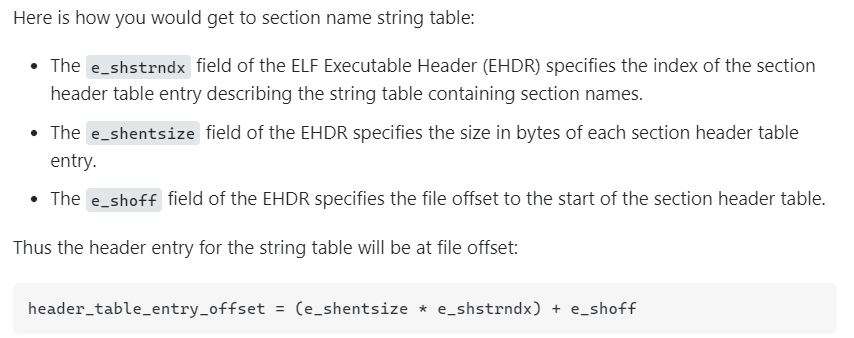


Cấu trúc giữa ELF 32bit và 64bit là giống nhau (khác nhau ở kích thước)

* sh\_name : chứa thông tin để tìn ra tên của section
* sh\_type : xác định loại section
* sh\_flags: xác định cờ của section
* sh\_addr: địa chỉ ảo cúa setion trong memory
* sh\_offset: kích thước của section trong file image
* sh\_size: kích thước tính bằng byte của section trong file image
* sh\_link: liên kết chỉ mục bảng tiêu đề phần
* sh\_info: thông tin bổ sung của section
* sh\_addralign: chứa liên kết cần thiết của section
* sh\_entsize: Chứa kích thướng tính bằng byte của section. Nếu section không có kích thước cố định giá trị là 0

\*cách tìm vị trí chứa tên các section

Tính header\_table\_entry\_offset



Offset\_name\_section = sh\_name + header\_table\_entry\_offset