

Descriptif de la structure du code développé : principales fonctions et fichiers correspondants

I - PHASE-1 :

elf_header.c :

get_header_shoff, get_header_phoff, etc. : Récupèrent et convertissent les champs de l'en-tête ELF (e.g., e_shoff, e_phoff) en utilisant ntohl ou ntohs pour gérer l'ordre des octets.

get_header_type_name, get_header_machine_name, get_header_OSABI_name : Convertissent les valeurs numériques des champs e_type, e_machine et EI_OSABI en descriptions textuelles.

get_header : Lit l'en-tête ELF et retourne une structure ELF allouée dynamiquement.

valider_elf : Vérifie si le fichier est un ELF valide en comparant les valeurs magiques.

afficher_entete_elf : Affiche les informations de l'en-tête ELF.

elf_section.c :

get_section_offset, get_section_size, get_section_addr, etc. : Récupèrent et convertissent les champs des en-têtes de section.

get_sections : Charge les en-têtes de section et retourne un tableau dynamique.

get_shstrtab : Extrait la table des noms de sections.

get_section_name : Retourne le nom d'une section à partir de son index.

get_section_type_name : Convertit le type de section en une chaîne lisible.

afficher_section_headers : Affiche les informations des sections.

elf_section_contenu.c :

afficher_contenu_section : Affiche le contenu d'une section en hexadécimal et ASCII.

afficher_section_par_index : Affiche le contenu d'une section par son index.

afficher_section_par_nom : Affiche le contenu d'une section par son nom.

read_section_content : Lit et affiche le contenu d'une section en fonction de l'index ou du nom.

elf_table_symbole.c :

get_symbol_type : Retourne une description du type de symbole.

get_symbol_binding : Retourne le type de liaison d'un symbole.

get_symbol_visibility : Retourne la visibilité du symbole.

get_symbol_name : Extrait le nom du symbole.

get_symbol_value : Retourne l'adresse virtuelle du symbole.

get_symbol_size : Retourne la taille du symbole.

get_symbol_shndx : Retourne l'index de la section associée au symbole.

afficher_table_symboles : Affiche les informations des symboles dans un tableau.

elf_relocation.c :

get_relocation_type : Retourne une description du type de réimplantation.

print_relocation_info : Affiche les informations détaillées des entrées de réimplantation.

elf_readelf.c :

main : Appelle les fonctions d'affichage en fonction des arguments.

II - PHASE-2 :

process_rel.c

set_header * (e.g., set_header_shoff, set_header_phoff, etc.) : Ces fonctions définissent les champs de l'en-tête ELF (e.g., e_shoff, e_phoff) en utilisant htonl ou htons pour gérer l'ordre des octets.

set_section * (e.g., set_section_name, set_section_type, etc.) : Ces fonctions définissent les champs d'une section ELF (e.g., sh_name, sh_type) en utilisant htonl ou htons.

setsymvalue : Définit la valeur d'un symbole (st_value) en gérant l'ordre des octets.

setsymstindx : Définit les informations d'un symbole (st_info, st_shndx).

setsize : Définit la taille d'une structure de symboles.

get_sh_name : Récupère le nom d'une section à partir de son en-tête.

update_symbol_addresses : Met à jour les adresses des symboles en fonction des sections allouables (e.g., .text, .data, .bss).

apply_relocation : Applique la réimplantation à une section cible en fonction du type de réimplantation (e.g., R_ARM_ABS32, R_ARM_JUMP24).

process_and_remove_relocation_sections : Traite et supprime les sections de réimplantation (SHT_REL) du fichier ELF, met à jour les symboles et écrit un nouveau fichier ELF modifié.

main : Point d'entrée du programme. Lit un fichier ELF, traite les sections de réimplantation, et génère un fichier ELF modifié.