

Compte-rendu

Introduction :

L'objectif de ce projet est de créer un programme capable de réaliser la phase de fusion d'un éditeur de lien (Linker). Cela nous permet de fusionner deux fichiers au format ELF dans un seul fichier. Le projet est composé en 2 phases :

- Phase 1 : Lecture d'un fichier ELF32, et l'affichage
- Phase 2 : Fusion des données et modification

Mode d'emploi :

Compilation

Pour compiler les fichiers C de notre projet, on doit taper ./configure, et après make, après qu'on soit dans le bon répertoire. Après on aura le programme exécutable readelf.

exécution :

`./readelf [option] fichierELF`

en lançant ce programme on obtiendra l'affichage

Les options sont :

- h : affichage du header (l'en-tête du fichier ELF)
- S : affichage de la table des sections
- x : affichage du contenu d'une section en tapant son numero ou son nom
- s : affichage de table de symboles
- r : affichage de relocations
- T : Obtenir le nombre de Section de fichier elf indiqué

`./readelf fichier1 fichier2 -f`

pour fusionner fichier1 et fichier2 au format ELF en un seul fichier

`bash AutoTest.sh`

pour test automatique

Description du code

Programme principal

Le programme est réparti en plusieurs fichiers où readelf.c en est le principal. Il contient simplement le main qui est divisé en deux parties : l'une comprend des différents **affichages** dans lesquels on peut choisir ce qu'on veut afficher selon un

fichier fourni au format ELF, l'autre est pour **fusionner** deux fichiers donnés en argument au format ELF si on a 4 arguments sur la commande.

Initialisation dans `all_elf_func.c`

Ces fonctions correspondent à toutes les déclarations des structures, l'allocation de leur mémoire pour mémoriser temporairement le header, sections, symboles, relocations et deux strings tables(.shstrtab et .strtab). Et changement en little endian éventuellement.

En fin on a une fonction pour **libérer** la mémoire des informations du fichier elf.

On a des fonctions séparées dans des fichiers :

header.c sections.c symbole.c relocation.c

pour implémenter le **lecture** et **l'affichage** des différentes parties du fichier ELF.

header.c

```
//affichage de l'en -tête
void affiche_Magic(Elf32_info elf);
void affiche_Classe(Elf32_info elf);
void affiche_Type(Elf32_info elf);
void affiche_DATA(Elf32_info elf);
void affiche_Version(Elf32_info elf);
void affiche_Machine(Elf32_info elf);
// si coder en big endian, transformer en little endian pour l'affichage correcte
void setto_little_endian(Elf32_info *elf);
//etape 1 Affichage de l'en - tête
void afficheHeader(Elf32_info elf);
```

section.c

```
// Stocke les informations relatives aux sections dans la structure elf.
void lire_Section_table(Elf32_info *elf, FILE *file);
// Alloue la mémoire pour stocker la table contenant les noms des sections.
void init_strtable(Elf32_info *elf, FILE *file);
// Liste de type de section
char * getSectionType(Elf32_Shdr sh);
// Liste de flags de section
char * get_elf_section_flags(Elf32_Word sh_flags);
// si coder en big endian, transformer en little endian
void sectionto_little_endian(Elf32_info *elf, int i);
//etape 2 Affichage de la table des sections
void afficheTableSection(Elf32_info elf, FILE *file);
//etape 3 Affichage du contenu d'une section
void afficher_contenu_section(Elf32_info elf, FILE* fsource);
```

symbole.c

```
//Stocke les informations relatives à la table des symboles dans la structure elf.
void read_Sym(Elf32_info *elf, FILE *file);
//Alloue la mémoire pour stocker la table contenant les noms des symboles.
```

```
void read_SymTable(Elf32_info *elf, FILE *file);  
//Affichage la valeur de symbole  
uint16_t print_val_sym(Elf32_info elf, FILE *file, int k);
```

```
//etape 4 Affichage de la table des symboles et des détails relatifs à chaque symbole  
void aff_s(Elf32_info elf, FILE *file);
```

relocation.c

```
//Stocke les informations relatives à la table des relocations dans la structure elf.  
void read_Rel(Elf32_info *elf, FILE *file);  
//etape 5 Affichage des tables de réimplantation et des détails relatifs à chaque entrée.  
void aff_r(Elf32_info elf, FILE *file);
```

swap.c

Des fonctions de transformer entre BIG ENDIAN et LITTLE ENDIAN.

Definitions et structures des codes

elf.h : des types de fichier elf en 32 bits.

baseElf32.h : une structure de type Elf32_info qui contient header, sections, strtable, symtab, reldtab.

Fusion des deux fichiers.c

```
void p2e6e8(Elf32_info elfo, Elf32_info elft);  
void p2e7(Elf32_info elfo, Elf32_info elft);  
void p2(Elf32_info elfo, Elf32_info elft);
```