Projet logiciel
Éditeur de liens - Phase de fusion

Extrant Robin, Omel Jocelyn , Raynaud Paul, Sorin Gaëtan et Visage Yohan

L'objectif de ce projet est de créer un programme capable de réaliser la phase de fusion d'un éditeur de lien. Cela permet de fusionner deux fichiers au format objet (de type ELF32 pour ARM) en un nouveau fichier.

Pour cela, le projet est décomposé en deux phases :

— Phase 1 : Lecture d'un fichier ELF32, interprétation et affichage des données collectées

— Phase 2 : Fusion des données issues de deux fichiers distincts

1 Compilation des programmes

Afin de compiler les fichiers C de notre projet, nous avons recours à l'outil CMake. Il suffit de l'utiliser ainsi, après s'être placé dans le répertoire du projet :

```
cmake .
make
```

2 Utilisation des programmes

Les exécutables produits se trouvent dans le même répertoire. On peut les utiliser comme ceci :

2.1 readelf

Cet utilitaire est similaire à son homonyme : il permet d'afficher sous forme structurée les informations d'un fichier au format ELF.

```
./readelf OPTIONS fichier1 [fichier2 ...]

Les options sont :
  -h, --file-header      Affiche l'en-tête du fichier ELF
  -S, --section-headers  Affiche les sections de l'en-tête
  -x, --hex-dump         Affiche le contenu en hexadécimal d'une section donnée
  -s, --syms             Affiche la table des symboles
```

```
  -r, --relocs           Affiche les réalocations (si présentes)
  -A, --all              Similaire à -h -S -s -r
  -H, --help             Affiche cette aide et quitte
```

2.2 fusion

Cet autre utilitaire permet de fusionner deux fichiers .o, à l'instar de la commande *ld -r -o fichier_sortie fichier_entrée1 fichier_entrée2.*

```
./fusion fichier_entrée1 fichier_entrée2 fichier_sortie

Ce programme ne dispose pas d'options.
La variable d'environnement DEBUG_FUSION sert à afficher ce que le programme fait.
```

3 Organisation du code

Le code est organisé en une partie commune (une bibliothèque locale), comprenant un fichier pour chacune des étapes de la première phase :

- L'étape 1 : la récupération de l'en-tête ELF
- L'étape 2 : la récupération de l'en-tête des sections
- L'étape 4 : la récupération de la table des symboles
- L'étape 5 : la récupération des tables de réimplémentations

Ensuite, il y a deux parties spécifiques dédiées à l'affichage et la fusion, qui sont respectivement *readelf* et *fusion.*

4 Fonctionnalités

Le programme *readelf* est capable de :

- Lire des fichiers au format ELF32 (peu importe l'architecture, du moment qu'elle est 32 bits)
- Lire l'en-tête du fichier, en affichant les informations mentionnées ci-dessus
- Récupérer les tables de symboles dynamiques (DYNSYM)
- Lire les tables de réimplémentations avec addend explicite (RELA)
- Reconnaître de façon exhaustive tous les types (voir type_strings.h)
- Convertir les données au format Big Endian sur une machine Little Endian et vice-versa

En revanche, il n'est pas capable de faire le café !

Le programme **fusion** est capable de :

- Fusionner les sections PROGBITS
- Fusionner et corriger les tables de symboles
- Fusionner et corriger les tables de réimplémentations
- Produire un nouveau fichier .o, correspondant à la concaténation des deux fichiers d'entrée

Mais il n'est pas capable de :

— Adapter les sections propres à ARM (comme .ARM-attributes)
— Gérer les addend explicites (RELA)

5 Bugs connus

— Problème au niveau des addends lors de la réimplémentation. Ce qui rend impossible la compilation ultérieure.