Compte Rendu Projet Programmation

Descriptif de la structure du code développé

getELF.c
Ensemble de fonctions permettant de lire les différentes parties d'un fichier objet et renvoie une structure de données propre à chaque partie du fichier ELF.
La fonction lireFichierELF() utilise les autres fonctions de getELF.c pour lire et stocker en mémoire un fichier ELF entier.

getELFHeader.c
Programme qui prend en argument un fichier objet. La fonction afficherELFHeader() utilise une fonction de getELF.c pour lire le ELF header du fichier et l'affiche au format de readelf -h.

getSectionHeaderTable.c
Programme qui prend en argument un fichier objet.
La fonction afficherSectionHeaderTable() se déplace aux bons endroits dans le fichier avant d'appeler les différentes fonctions de getELF.c pour lire le section header table du fichier et l'affiche au format de readelf -S.

getSection.c
Programme qui prend en argument un numéro/nom de section et un fichier objet.
La fonction afficherSectionNum() se déplace aux bons endroits dans le fichier avant d'appeler les différentes fonctions de getELF.c pour lire la section au numéro donné et l'affiche au format de readelf -x.
Si c'est le nom de la section qui a été donné, la fonction trouverNumSection() renvoie le numéro de la section correspondante.

getSymbolTable.c
Programme qui prend en argument un fichier objet.
La fonction afficherSymbolTable() se déplace aux bons endroits dans le fichier avant d'appeler les différentes fonctions de getELF.c pour lire la table des symboles du fichier et l'affiche au format de readelf -s.

getRelocationTable.c
Programme qui prend en argument un fichier objet.
La fonction afficherRelocationTable() se déplace aux bons endroits dans le fichier avant d'appeler les différentes fonctions de getELF.c pour lire les tables de réimplantations du fichier et les affichent au format de readelf -r.

fusionELF.c
Programme qui prend en argument deux fichiers objet.
Le fonction main() utilise une fonction de getELF.c pour stocker en mémoire les deux fichiers objet, déclare la structure correspondant au fichier résultat et alloue la mémoire nécessaire à la fusion.
La fonction FusionSectionsCode() va fusionner les sections de code des deux fichiers données en argument en remplissant la structure du fichier résultat.
La fonction Fusionrenumerotationsymboles() va fusionner la table des symboles des deux fichiers données en argument en remplissant la structure du fichier résultat.
La fonction afficherELF() prend la structure du fichier résultat en argument et affiche son contenu.

Liste des fonctionnalités implémentées et manquantes

Implémentées
- lecture et stockage d'un fichier ELF en mémoire
- affichage d'un fichier ELF au format de readelf pour toutes les étapes de la phase 1
- fusion des sections de code et de la table des symboles de deux fichiers objets
- affichage du fichier ELF fusionné dans la console (sections de code + table des symboles)

Manquantes
- correction des numéros des sections et des valeurs des symboles
- fusion et correction des tables de réimplantations
- production d'un fichier résultat au format ELF

Liste des éventuels bogues connus mais non résolus

Aucun bug révélé lors de nos tests.

Liste et description des tests effectués

Le script shell test.sh parcourt tous les fichiers du répertoire de tests donné en argument. Pour chaque fichier de test, tous les programmes de la phase 1 sont exécutés et l'affichage résultant est comparé à celui de readelf.
Pour les tests du programme getSection, on incrémente une variable pour exécuter avec tous les numéros de section jusqu'à recevoir une erreur indiquant que nous les avons toutes testées.
Les fichiers tests sont créés à partir de 9 programmes C de 5 à 250 lignes.

Pour lancer les tests :
1 - se déplacer dans le dossier « tests » puis exécuter `./compile.sh` pour compiler tous les fichiers tests
2 - revenir dans le dossier source puis exécuter `make all`
3 – exécuter la commande :
`./test.sh . ./tests/`
Les différences entre nos affichages et ceux de readelf seront affichées en vert et en rouge. Lors des tests du programme getSection, le message d'erreur de readelf apparaît car on teste les sections une par une jusqu'à un index invalide. Le second message d'erreur est celui du programme getSection, qui imite l'affichage de readelf.