

Projet Prog5

Groupe 10

ARNAUD Tom
BESSON Corentin
COMTOIS Tommy
DODE Etienne
GRONDIN Denis
PIET Emilien

MODE D'EMPLOI

- Se placer dans TEETDC/elf_linker-1.0

Pour compiler le code :

- Exécuter la commande make

Pour exécuter le code :

- Exécuter la commande ./ELFRead [nom_fichier] pour exécuter la Phase 1
- Exécuter la commande ./ELFFusion [nom_fichier1] [nom_fichier2] [fichier_de_sortie] pour exécuter la Phase 2 (les fichiers fich1.o et fich2.o dans Examples_loader/ sont prévus a cet effet)

STRUCTURE DU CODE

Le code de la phase 1 et 2 sont divisés en 7 fichiers sources .c ayant respectivement chacun un fichier header .h pour chaque étape nommées « partieX.c/h ».

Chaque header contient le prototype de chaque fonction permettant les différentes lectures et réécritures ainsi que le type de données qu'elle prend en paramètres, une description de ce qu'elle réalise et sa valeur de retour.

Chaque fichier source de la phase 1 contient la définition des fonctions de son fichier header. Il contient une fonction de lecture et une fonction d'affichage. Chaque fonction d'affichage est divisée en sous fonctions, lesquelles sont appelées dans la fonction d'affichage principale.

Le projet comporte 2 fichiers 'main' qui servent à tester la phase 1 (lecture et affichage) et la phase 2 (fusion).

partie1.c :

- la fonction de lecture de l'en-tête du fichier : *'elf_read_entete'*
- la fonction d'affichage de l'entete : *'afficherHeader'*.

Partie2.c :

- la fonction de lecture l'en-tête de la Table des Sections : *'lectureTableSectionHeader'*
- la fonction de création d'une String Table : *'createStrTab'*
- la fonction d'affichage de la Table des Sections *'printTableSection'*.

Partie3.c :

- la fonction d'affichage du contenu d'une section : *'affichage_contenu_section'*

Partie4.c :

- la fonction de lecture de la table des symboles : *'readSymTab'*
- la fonction d'affichage de la table des symboles : *'afficherTabSymb'*

Partie5.c :

- la fonction de lectures des tables de relocations *'getRelTable'*,
- la fonction d'affichage d'une table de relocation *'afficherRelTables'*
- la fonction principale d'affichage *'affiche_toute_reloc'*.

ELFRead.c : fichier source contenant la fonction main, il contient l'initialisation des variables et l'appel des fonctions.

ELFFusion.c : le main et toutes ses fonctions sont dans le même fichier ,il permet de fusionner deux fichier.o en un troisième fichier .

FONCTIONNALITÉS IMPLÉMENTÉES ET MANQUANTES

1.Affichage de l'en-tête : Toutes

2.Affichage de la table des sections : Toutes

3,Affichage du contenu d'une section : Toutes

4.Affichage de la table des symboles : Toutes

5.Affichage des tables de réimplantation : Toutes

6.Fusion et renumérotation des sections : Toutes

7.Fusion,renumérotation et correction des symboles : Toutes

8.Fusion,renumérotation et correction des tables de réimplantations : Toutes

9.Production d'un fichier résultat au format ELF : Toutes

non implémenté :

- les fichiers gérés sont aux formats ELF_32 et little endian
- Pas de gestion des addends (dans le cadre des tables de relocation)
- Dans le même sens, pas de gestion des addends à la réécriture.
- pas de vérification du format de fichier donné

BOGUES CONNUS MAIS NON RÉSOLUS

-le fichier fusionné ne peut pas être transformé en exécutable

POINTS IMPORTANTS

- dans la table des symboles nous avons des symboles 'PROTECTED' alors que les outils donnés ont des 'DEFAULT' mais la différence est due à la différence des versions .

-manque d'optimisation de la table des sections(simple concaténation)

TESTS EFFECTUES

Pour exécuter les tests :

(les parties à tester doivent déjà être compiler)

-se mettre dans le répertoire Test

- Exécuter la commande make

Pour exécuter tous les tests :

- Exécuter la commande ./allTest.sh

nous partons du principe que seule des fichier ELF sont donner en argument

1.Affichage de l'en-tête :

test : vérifier la bonne lecture d'un header

donner un fichier ELF.

./partie1,sh [nom_fichier_elf]

affichage du header

2.Affichage de la table des sections :

test : vérifier la bonne lecture de la table des sections

donner un fichier ELF.

./partie2,sh [nom_fichier_elf]

affichage de la table des sections

3,Affichage du contenu d'une section :

test : vérifier la bonne lecture d'une section

donner un fichier ELF.

./partie3,sh [nom_fichier] [numéro_de_section]

affichage de la section

4.Affichage de la table des symboles :

test : vérifier la bonne lecture de la table des symboles

donner un fichier ELF.

./partie4,sh [nom_fichier_elf]

affichage de la table des symboles

5.Affichage des tables de réimplantations :

test : vérifier la bonne lecture des tables des réimplantations

donner un fichier ELF.

./partie5,sh [nom_fichier_elf]

affichage de la table des réimplantations

6.Fusion :

Toute la partie 2 n'a pas de tests automatisés , mais nous faisons des tests en créant un fichier avec ELFFusion et un autre fichier avec les outils donnés sur les mêmes fichiers de base et ensuite nous faisons une comparaison entre ces deux fichiers à l'aide du module de lecture(ELFRead).