

COMPTE RENDU

Marouane AIT-HANINE | Omar BARKOK | Félix IRADUKUNDA

Contexte et Objectifs :

L'objectif de ce projet est de développer un programme capable d'effectuer la phase de réimplantation d'un éditeur de liens, spécifiquement pour des fichiers binaires translatables au format ELF ciblant l'architecture ARM en mode big endian. La réimplantation consiste à attribuer des adresses absolues aux symboles et à corriger les références dans les sections de code et de données, afin de produire un fichier exécutable non translatable.

Structure du Code :

Le programme principal **my_readelf** utilisé pour exécuter toutes les étapes implémentées.

Un module **my_elf** qui contient les interfaces implémentant les différentes étapes.

Bref mode d'emploi :

Pour pouvoir exécuter le programme principal, et donc tester à la main les différentes fonctions implémentées, il faut d'abord configurer l'environnement et les exemples puis compiler le programme principal avec make.

Pour afficher un message d'aide avec les différentes options et modes d'utilisation, il suffit d'exécuter ce dernier en ligne de commande sans rajouter d'argument.

Liste des fonctionnalités :

Implémentées:

- les fonctions d'affichage (étapes 1-5)
- les étapes 6-8 de la phase 2

Manquantes:

- Interfaçage avec le simulateur ARM
- Production d'un fichier exécutable non relogeable

Liste bugs connue non résolus :

- **Bug test:**
testeur3.sh. Le script shell est censé comparer les résultats de notre programme de l'étape 3 ce qu'il fait bien, sauf dans certains cas. Le cas où une section n'a pas de contenu.

Liste et description des tests effectués

Pour la phase 1 on a développé 5 tests en langage shell automatisés:

- *testeur1.sh
- *testeur2.sh
- *testeur3.sh
- *testeur4.sh
- *testeur5.sh

Elles ont été appliquées à l'ensemble des exemples fournis dans le répertoire `Example_loader`. Chaque `testeur*.sh` correspond à l'étape qu'on cherche à tester.

Description

Les scripts ont le même squelette, mais avec des différences dépendant du cible.

La base des tests est de comparer (en utilisant la commande: `diff`) deux fichiers `.txt` (`RESULT` et `RESULTARM`) qui contiennent les sorties de nos fonctions d'affichage du module **my_elf.c** et celles du programme **arm-none-eabi-readelf** respectivement. On a décidé de comparer que des parties pertinentes en coupant les sorties à l'aide des outils `awk` et `sed`.

Comment lancer les tests

Le lancement des tests se fait tout simplement en tapant le nom du testeur dans le terminal. Exemple: **./testeur4.sh**

Conclusion des tests

Les tests sont réussis si le script n'affiche rien ou s'il affiche que des traits (-----) ce qui veut dire que les deux fichiers .txt sont équivalents. Si les tests échouent, le script affiche les lignes qui sont différentes entre les deux fichiers .txt correspondant au sortie des deux programmes (arm-none-eabi et my_readelf).