

Projet PROG5 – Année 2022-2023
Réalisation d'un éditeur de liens – Phase de fusion

Romain ALVES - Romain BOSSY - Victoria FIORI
Lancelin FOUSSADIER - Jules SEBAN

Journal de bord

Date du début du projet : 14/12/2022

14/12/2022 :

Découverte du sujet du projet et des modalités en amphithéâtre de présentation et par la lecture du polycopié. Première approche de l'étape 1 de la partie 1.

15/12/2022 :

Avancées sur les étapes 1 et 2 de la partie 1(Tout le groupe).

16/12/2022 :

Étape 1 terminée. Avancées sur les étapes 2 et 3 de la partie 1. Début de résolution du problème du passage en big endian, jusqu'à maintenant en little endian sur nos machines.

19/12/2022 :

Étapes 2 et 3 terminées, intégration des étapes 1 à 3 dans le main et résolution du problème du passage en big endian.

20/12/2022 :

Oracles 1(Romain Bossy), 2 (Lancelin et Victoria) et 4 terminés. Étape 4 terminée (Jules).

Du 21/12/2022 au 02/01/2023 :

interruption pédagogique.

03/01/2023 :

Précision des read error et début de l'étape 5 (Lancelin).

04/01/2023 :

Passage à l'audit de code, vision de notre avancée et échange à propos des problèmes rencontrés. Objectif : corriger nos erreurs de parcours et continuer d'avancer. Soucis au niveau de la complexité des oracles, implémentés en C jusqu'à maintenant. Cette tâche pourrait être simplifiée si les oracles sont écrits dans un autre langage de programmation comme le Bash, ou bien le Python. Poursuite de l'étape 5 et de la création de l'oracle de l'étape 3 (Romain Alves).

05/01/2023 :

Étape 5 et son oracle (Lancelin et Jules) ainsi que l'étape 6 (Jules) terminés.
Avancés dans l'oracle de l'étape 3 (Romain Alves).

06/01/2023 :

Oracles de l'étape 3 (Romain Alves) et de l'étape 6 (Romain Bossy) terminés.
Correction d'une erreur dans l'oracle de l'étape 5. Erreur relevée sur l'étape 2, besoin de la modifier.

07/01/2023 :

Problème d'étape 2 réglé (Lancelin et Jules), étape 7 en cours (Romain Bossy)

09/01/2023 :

Ébauche de l'étape 7 réalisée par Romain Bossy, oracle de l'étape 6 corrigé par Jules, oracle de l'étape 2 corrigé par Lancelin, diaporama de présentation réalisé par Victoria et Romain Alves. Victoria a des problèmes de production de fichier.o

10/01/2023 :

Étape 7 et son oracle en cours d'écriture, début de l'étape 8 et résolution de bugs généraux au projet par Romain Bossy, Jules et Lancelin. Poursuite du diaporama de présentation et productions des documents à rendre par Romain Alves et Victoria.

11/01/2023 :

Jour de fin de projet, la deadline est fixée à 12h. Dernière ligne droite, dernières modifications, résolutions de bugs et implémentations dans les étapes 8, 9 et ajout d'un test.

Mode d'emploi :

Se référer au fichier *README.md* dans le projet.
La version de readELF 2.38 est nécessaire.

Descriptif de la structure du code :

Partie *utils.c* → Ici, les fonctions implémentées sont pour faciliter le travail de ELF.c.
Partie *ELF.c* → Sont présentes les fonctions de lectures et d'affichages pour les différentes étapes implémentées.
Partie *oracle.c* → Permet de tester les différentes étapes de manière automatique avec des fichiers tests.

Partie *tests* → Contient 6 fichiers de test en ARM numérotés de 1 à 6 afin de tester les différentes étapes implémentées en lien avec les oracles contenus dans la partie *oracle*.

Fonctionnalités implémentées :

Dans la phase 1 :

- Étape 1 → lecture et affichage des différents champs de l'en-tête d'un fichier ELF ainsi que son oracle pour tester.
- Étape 2 → lecture et affichage de la table des sections d'un fichier ELF ainsi que son oracle.
- Étape 3 → lecture et affichage du contenu de l'une des sections d'un fichier ELF ainsi que son oracle.
- Étape 4 → lecture et affichage de la table des symboles d'un fichier ELF ainsi que son oracle.
- Étape 5 → lecture et affichage des tables de réimplantation d'un fichier ELF pour machine ARM ainsi que son oracle.

Dans la phase 2 :

- Étape 6 → concaténation des sections de code de deux fichiers binaires translatables et mémorisation du possible numéro de section du second fichier et de l'offset où a eu lieu la concaténation. Ainsi que son oracle.
- Étape 7 → construction de la table des symboles du fichier de sortie. Ainsi que son oracle.

Fonctionnalités obsolètes :

Dans la phase 2 :

- Étape 8 → fusion et correction des entrées des tables de réimplantation des fichiers en entrée. Ainsi que son oracle.
- Étape 9: écriture du résultat de la fusion dans un fichier au format ELF. Le contenu de ce fichier de résultat est illisible sauf son header.

Fonctionnalité manquante :

Dans la phase 2 :

Oracle de l'étape 9.

Liste des bogues connus :

Lorsqu'on écrit le résultat de la fusion de deux fichiers ou le contenu d'un fichier ELF correct, on peut lire le header du fichier de résultat mais pas les autres parties comme la table des sections et la table des symboles.

Tests effectués :

Utilisation des tests fournis et ajout de nouveaux tests. Les tests sont effectués sur file1.o et file2.o, les tests de 1 à 6, sur oracle.o et ELF.o. La commande à utiliser est `“./oracle ./Examples_fusion/file1.o”` ou bien `“./oracle ./Examples_fusion/file2.o”`

Fichiers de tests :

test1.o → Ce test consiste à retourner la valeur d'une variable.

test2.o → Ce test consiste à savoir si une valeur a bien été assignée à une variable.

test3.o → Ce test consiste à savoir si une valeur et un texte ont bien été assignés chacun à une variable.

test4.o → Ce test vérifie que l'adresse d'un tableau a bien été assignée à une variable

test5.o → teste si un message est bien retourné via l'utilisation de la macro `print_message`

test6.o → cherche à avoir la variable `r0` égale à 2

testdata.o :

- créer 2 variables globales "buf" de 40 bytes et "len" pour la taille de la chaîne de caractère.
- charger l'adresse de début de "buf" dans `r0`
- charger l'adresse de début de "len" dans `r1`
- calcule l'adresse où stocker la prochaine donnée dans buf en fonction de la valeur de len
- charge 0 dans `r3` pour stocker un octet de donnée
- stock cette donnée dans l'emplacement pointé par `r2` dans buf
- incrémente len de 1
- sauvegarde la nouvelle valeur de len
- retourne à l'appelant