

Projet PROG5

Tom CLEMENT/Henri GROS/Thibault PONÇON
Théo KOEHLER/Raphaël DEMOULIN

Decembre 2022 - Janvier 2023

Lien Github : <https://github.com/Karvyz/Prog5-linker>

Contents

1	Mode d'emploi	2
1.1	Copier et compiler le projet	2
1.2	Lecture d'un fichier ELF	2
2	Structure du code	3
2.1	Disponibilité des fonctionnalités	3
2.2	Bogues connu	3
3	Les tests	3
4	Journal de bord	4
4.1	Journal explicatif	4
4.2	Diagramme de Gantt	6
5	Conclusion	6

1 Mode d'emploi

1.1 Copier et compiler le projet

Installation

- Cloner le projet :

```
git clone https://github.com/Karvyz/Prog5-linker
```

- Se déplacer dans le repertoire du projet :

```
cd Prog5-linker
```

Initialisation

```
./configure  
make  
sudo make install
```

Nettoyage

```
make clean  
make distclean
```

1.2 Lecture d'un fichier ELF

Utilisation

```
./read_elf [options] FILE
```

```
Display information about the given ELF file  
-H —help      Display this information  
-h             Display the ELF header  
-S             Display the sections headers  
-s             Display the symbol table  
-x <num|text>  Display the content of the section <num|text>  
-r             Display the relocation table  
-F FILE2       Fusion of FILE2's sections in FILE
```

Réaliser les tests

-Pour faire les test, exécuter les commandes suivantes :

```
make check
```

2 Structure du code

2.1 Disponibilité des fonctionnalités

- Étape 1 : Achievée, Oracle réalisé
- Étape 2 : Achievée, Oracle réalisé
- Étape 3 : Achievée, Oracle réalisé
- Étape 4 : Achievée, Oracle réalisé
- Étape 5 : Achievée, Oracle réalisé
- Étape 6 : Achievée, Oracle en cours : les test sur la memoire sont finis mais les test sur les résultats sont en conception
- Étape 7 : Inachevée, n'arrive pas à acceder corectement au *data* du type *sectionFusion*
- Étape 8 : Inachevée, manque la table des symboles fusionner pour pouvoir continuer.
- Étape 9 : Inachevée, écrit correctement le header dans un fichier de sortie

2.2 Bogues connu

Lors de l'exécution des tests automatiques, on peut voir que le test *mem_fusion* ne passe pas car il y a encore des erreurs avec valgrind sur cette partie mais cela n'empêche pas l'exécution normale de la partie fusion.

3 Les tests

Nous avons créer un oracle qui effectue tous les tests (étape 1 à 6) pour tous les fichiers ELF stockés dans Exemple_ELF_Files.
On y teste les fuites mémoires (avec valgrind) et le résultat des fonctions.
La commande *make check* permet de retourner tous les tests effectués.
Chaque test est écrit en shell et vérifie caractère par caractère si nos fonctions renvoi la même chose que les fonctions readelf (-h -S -x <num|name> -s -r) :

- *test_headers.sh* = *fichier* test pour readelf -h
- *test_section_headers.sh* = fichier test pour readelf -S
- *test_content_section.sh* = fichier test pour readelf -x
- *test_symbole.sh* = fichier test pour readelf -s
- *test_relocations.sh* = fichier test pour readelf -r

4 Journal de bord

4.1 Journal explicatif

14/12/22

Lecture et compréhension du sujet

Tout début de l'étape 1

15/12/22 :

Répartition du travail fait :

étape 1 : tom

étape 2 : théo / raphaël

étape 3 : tom

étape 4 : tom

étape 5 : thibault

étape 6 : tom

étape 7 : raphaël

étape 8 : thibault

étape 9 : thibault

les oracles : henri

CR : théo / raphaël

Lecture bit à bit, Big-Endian (incorrecte)

Création de l'en-tête (done)

Affichage de l'en-tête (done)

Etape 1 presque finie

19/12/22 :

Correction de la lecture en big endian et

de la fonction d'affichage de l'en-tête

=>Étape 1 (done)

Étape 2 : Lecture des tables des sections (done)

Étape 3 : Affichage du contenu d'une section (done)

20/12/22 :

Etape 4 : Affichage de la table des symboles et des détails relatifs

à chaque symbole (done)

03/01/22 :

Étape 2 : affichage des données (done)

Oracle 1 (done)

04/01/22 :

Audit de code : ok => possible simplification de l'affichage
de l'étape 2

Étape 6 en cours

Oracle 3 (done)

Étape 5 presque finie

05/01/22 :

Étape 5 (done)

Oracle étape 2 et 5

Simplification étape 2 (done)

Étape 6 toujours en cours, découverte de problème

06/01/22 :

Oracle étape 3 (done)

Construction d'un oracle qui teste tous les oracles ainsi que tous les valgrinds

Commence à régler les fuites mémoires (free)

Étape 6 : fusion OK mais problème affichage

09/01/22 : Étape 1 à 5 : fuites mémoires réglés

Étape 6 : fuites mémoires réglées

10/01/22 :

Étape 6 (done) (hormis valgrind)

11/01/22 :

Nétoyage du code

CR

4.2 Diagramme de Gantt

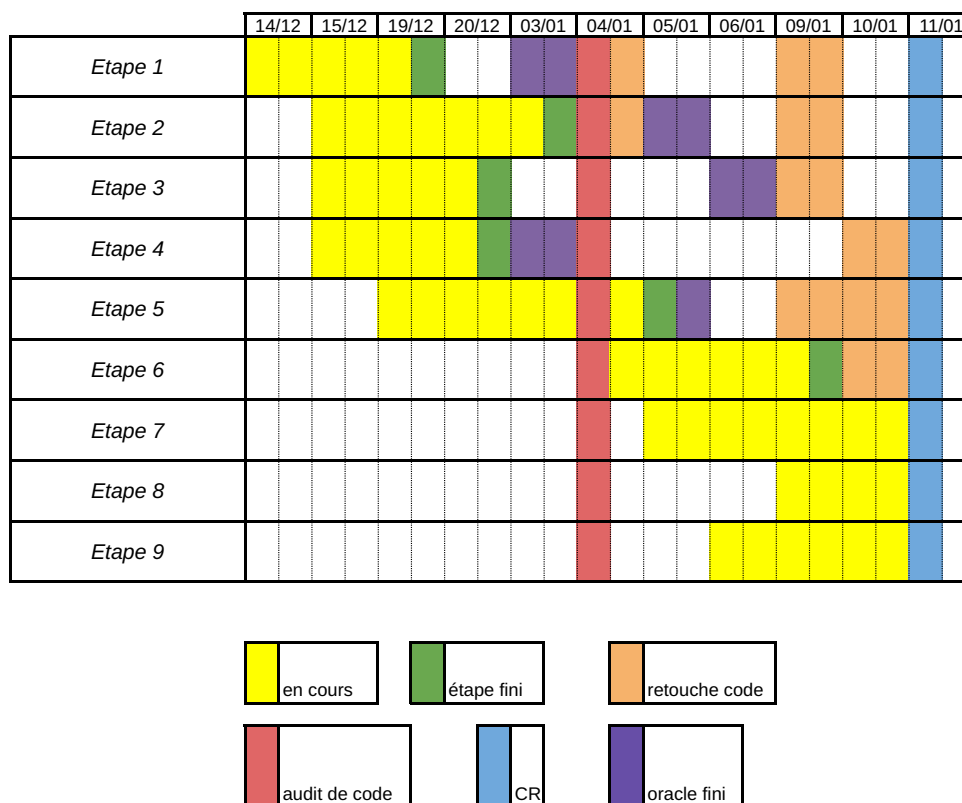


Figure 1: Diagramme de Gantt

5 Conclusion

Lors de ce projet nous avons pu expérimenter la programmation sur une durée étendue avec un groupe plus conséquent que les projets précédents. Cela nous a permis de voir les difficultés de répartition de travail entre les différents membres.

Nous avons acquis plein de compétences dans un domaine que nous ne pensions pas explorer un jour.