

Aymerick Beauvais
Anis Belahadji
Eliott Dutronc
Clement Heresaz
Quentin-Gerome Omel
Virgil Lacondemine

Projet L3 Informatique – Semestre 5 Groupe 10

Documentation complémentaire

Table des matières

Mode d'emploi	3
Structure du code.....	4
Liste des fonctionnalités.....	5
Liste des bogues	6
Tests	6
Journal de bord.....	7

Mode d'emploi

Pour compiler l'utilitaire développé, il suffit d'exécuter le Makefile fournit avec la commande « make ». Afin d'être sûr que la compilation s'effectue correctement, il est recommandé d'effectuer la commande « make clean » avant chaque nouvelle compilation.

La partie affichage du programme s'utilise avec l'exécutable « main » crée lors de la compilation, suivit des options suivantes :

- -h [nom du fichier].o : Affichage de l'en-tête du fichier passé en paramètre.
- -S [nom du fichier].o : Affichage de la table des sections du fichier passé en paramètre.
- -x [numéro de section/nom de section] [nom du fichier].o : Affiche la section de numéro ou de nom passé en paramètre du fichier passé en paramètre.
- -s [nom du fichier].o : Affichage de la table des symboles du fichier passé en paramètre.
- -r [nom du fichier].o : Affichage de la table des réimplantations du fichier passé en paramètre.
- -f [nom du fichier 1].o [nom du fichier 2].o [nom du fichier final].o : fusionne les fichiers 1 et 2 dans un fichier final. Les fichiers 1 et 2 doivent être compilés au préalable avec les commandes suivantes :
 - arm-none-eabi-gcc -mno-thumb-interwork -S [nom du fichier].c
 - arm-none-eabi-as -o [nom du fichier].o [nom du fichier].s

Structure du code

La structure de fichier du code est la suivante :

- **display.c** : contient toutes les fonctions d'affichage du programme.
- **info.c** : contient les fonctions de renommage utile pour l'affichage afin de faciliter la lisibilité des informations.
- **reader.c** : contient les fonctions de récupération des différentes parties d'un fichier elf (header, section header, etc) et les alloues en mémoire pour qu'elles puissent être utilisées par la suite.
- **main.c** : contient la fonction main du programme. On peut faire appel aux fonctions présentes dans tous les autres fichiers depuis main.c. C'est ce même fichier qui devient l'exécutable lors de la compilation.
- **fusion.c** : gère la fusion de deux fichiers au format elf.
- **Makefile** : fichier qui sert à la compilation du programme.

Liste des fonctionnalités

Réalisées	Non réalisées
Affichage de l'en-tête	Fusion, renumérotation et correction des symboles
Affichage de la table des sections	Fusion et correction des tables de réimplantations
Affichage du contenu d'une section	Production d'un fichier résultat au format ELF
Affichage de la table des symboles	
Affichage de la table de réimplantation	
Fusion et renumérotation des sections	

Liste des bogues

Lors de la fusion, les offsets ne sont pas g rer correctement.

Tests

- Commande sans argument : Affiche les options disponibles.
- Fichier inexistant pass  en param tre de la fonction : Affiche un message d'erreur pour pr venir l'utilisateur que le fichier n'existe pas.
- Le fichier pass  en param tre n'est pas au format ELF : Affiche un message d'erreur pour pr venir l'utilisateur que le fichier n'est pas au format ELF.
- Ex cution du programme avec l'option -x sans passer de nom ou de num ro de section en param tre : Affiche un message d'erreur   l'utilisateur pour l'informer de l'utilisation de l'option.
- Ex cution du programme avec l'option -x avec un mauvais num ro ou nom de section pass  en param tre : Affiche un message d'erreur   l'utilisateur pour lui sp cifier que le nom ou le num ro de section est incorrect.
- Ex cution du programme avec un trop grand nombre d'argument : Affiche un message d'erreur   l'utilisateur pour l'informer de l'utilisation de l'option.

Le programme a aussi  t  ex cut  avec plusieurs fichiers de taille diff rente et de complexit  diff rente.

Journal de bord

Mardi 03 Janvier 2017 :

- Découverte du sujet pour tous les membres du groupe.
- Lecture et compréhension de la documentation fournie.
- Début du développement du programme avec l'affichage de l'en-tête d'un fichier elf.

Mercredi 04 Janvier 2017 :

- Fin du développement de l'affichage de l'en-tête d'un fichier elf.
- Divers test sont effectués sur cette fonctionnalité.
- Début du développement de l'affichage de la table des sections.

Jeudi 05 Janvier 2017 :

- Fin du développement de l'affichage de la table des sections.
- Divers test sont effectués sur cette fonctionnalité.
- Début et fin du développement de l'affichage du contenu d'une section.
- Divers test sont effectués sur cette fonctionnalité.
- Début du développement de l'affichage de la table des symboles.

Vendredi 06 Janvier 2017 :

- Fin du développement de l'affichage de la table des symboles.
- Divers test sont effectués sur cette fonctionnalité.
- Début et fin du développement de l'affichage de la table de réimplantation.
- Divers test sont effectués sur cette fonctionnalité.

Lundi 09 Janvier 2017 :

- Audit de code.
- Travail de factorisation sur le code de la première partie du projet.
- Début du développement de la fusion et renumérotation des sections de deux fichiers elf.

Mardi 10 Janvier 2017 :

- Factorisation du code terminé.
- Développement de la fusion et renumérotation des sections de deux fichiers elf.
- Début de l'écriture des documents complémentaire demandés.

Mercredi 11 Janvier 2017 :

- Fin du développement de la fusion et renumérotation des sections de deux fichiers elf.
- Divers test sont effectués sur cette fonctionnalité.

Jeudi 12 Janvier 2017 :

Réagencement des fichiers code.

Finalisation de la fusion et renumérotation des sections de deux fichiers elf.

Début du développement de l'étape 7 sans succès.

Début du développement de l'étape 8 sans succès.

Mise à jour du compte rendu.

Vendredi 13 Janvier 2017 :

Finalisation du compte rendu.

Fin du développement et présentation du projet lors de la soutenance.