

Projet d'éditeur de liens au format ELF

Journal de bord

Lundi 4 Janvier

Recherche documentaire.

Afin de mieux appréhender le projet, nous avons regardé les ressources techniques fournies et d'autres ressources glanées sur Internet. Nous avons planifié le travail et commencé la mise en œuvre du projet.

Mardi 5 Janvier

Programmation de la Phase 1 du Projet.

On démarre la programmation du projet et on continue les recherches documentaires (2 groupes distincts)
L'étape 1 est fini. Cependant, il y a eu un ajustement suite aux contraintes imposées dans le projet (se limiter aux ELF 32 bits).

Mercredi 6 Janvier

Seconde version de l'étape 1, programmation de l'étape 2.

Afin d'être plus simple et efficace, une seconde version de l'étape 1 a été réalisée. Tout d'abord sans utiliser le fichier elf.h et ses fonctions, avant de réintégrer ce fichier. En parallèle, l'étape 2 commence à être programmée (2 groupes distincts).

Reprise par la suite de la 2ème version pour continuer à programmer la 2ème étape.
On rencontre un problème lors de la finalisation du code : le tableau est bien affiché et bien indenté, cependant la section 7 « `.rel.debug_info` » rencontre des problèmes de récupération d'information sur les colonnes adresse, décalage et Fanions (Flags). Il reste à corriger ces problèmes pour pouvoir considérer l'étape 2 comme complétée.
On peut voir également que le texte n'est pas présent sur les Fanions et les types.

Jeudi 7 Janvier

Finition de 2ème étape et programmation de la 3ème, 4ème et 5ème étape.

Il y a eu deux groupes travaillant en parallèle : Dans un premier temps, pendant la matinée, un premier travail consistait à achever le travail de mise en forme du résultat de la phase 2 afin d'avoir le même affichage qu'avec l'outil "readelf -S".

Dans un deuxième temps, la seconde tâche consistait à continuer la phase 1 par la 3ème étape puis la 4ème étape en les affichant de la même façon, mise en forme comprise, que la fonction « readelf » le fait.

-- Note : Lors de la réalisation de la 4ème étape, le premier groupe a rencontré un problème : il est impossible d'intégrer toutes les chaînes de caractères affichées par la fonction « readelf » car, il semblerait que la documentation ne liste pas toutes les options de fichier que propose la fonction.

Fin de l'étape 4 et poursuite de la programmation de l'étape 5 à deux groupes.

-- Note : Dans l'étape 3, le sujet précise "la section choisie pourra être désignée par son numéro ou par son nom." Nous le faisons uniquement à partir de son numéro.

-- Note : Réorganisation des étapes du projet en modules (.h) et non sur un fichier unique, Ajout d'un Readme et d'un Makefile.

Vendredi 8 Janvier

Finition de l'étape 5.

Après plusieurs heures de travail, la partie 5 est terminée. La principale difficulté nécessita, pour la résoudre, que nous comprenions comment récupérer les données afin d'obtenir le même affichage que la fonction « readelf ».

Dans un deuxième temps, un second groupe, s'est appliqué à mettre en forme le programme afin de le rendre compatible avec des fichiers "Big endian". Cette partie du travail n'est pas encore achevée au soir du vendredi car ce travail dépendait en partie des progrès du premier groupe.

Préparation de la phase 2 et de l'implémentation des options (par exemple « -x », « -s » ou « -r »).

Lundi 11 Janvier

Implémentation des Options, documentation, tests. Début de la phase 2, début de l'étape 6.

Tout d'abord les options ont été implémentées afin de ne plus avoir à écrire en dur dans le programme les fonctions à tester. Le fichier README a été corrigé pour refléter les modifications.

Ensuite, nous nous sommes séparés en deux groupes de travail. Le premier entame l'étape 6 pendant que le second écrit une documentation du programme en utilisant le logiciel Doxygene. Cependant cette tâche de documentation a été reportée suite à l'audit de code qui conseille de se concentrer davantage sur les tests. Cet audit a d'ailleurs relevé des possibilités de factorisations de fonctions du code que nous n'avions pas exploité et nous a permis de comprendre la nécessité d'utiliser un tableau (array) afin de pouvoir éditer les globales lors de la phase 2.

En fin de journée, l'étape 6 est sur le point d'être terminée (quelques bugs demeurent) et implémentation des codes de test pour la soutenance est partiellement achevée.

-- Note: Ajouts de tests dans le programme.

Mardi 12 Janvier

Finition de l'étape 6, commencement de l'étape 7 et fonctionnalités.

Nous nous sommes répartis le travail entre deux équipes : le premier corrige les bugs restant du code émis hier. Dès que cela est fait, ce groupe est passé à la programmation de l'étape 7.

Parallèlement, la seconde équipe s'est attelée à la documentation "Doxygen" du code et à l'ajout de fonctionnalités : une option permettant d'utiliser les fonctions développées à l'étape 6 ainsi qu'une gestion du renommage dynamique du fichier sortant (sans les sections REL).

En fin de journée, le groupe s'est reconstitué pour continuer l'étape 7.

Mercredi 13 Janvier

Finition de l'étape 7, commencement de l'étape 8 et rédaction de documents annexes .

Le groupe se scinde en deux groupes de travail. Pendant qu'une équipe termine l'étape 7 en rencontrant plusieurs difficultés et réalise en fait une partie de l'étape 8 dans le même temps, l'autre moitié du groupe commence à rédiger les documents annexes à

rendre avec le projet : les organigrammes de la structure du projet et des fonctionnalités. Ensuite les 2 groupes de travail ont fusionné afin d'achever ensemble les étapes 7 et 8, lesquelles sont terminées à la fin de la journée.

-- Note : Mise à jour de la documentation Doxygen.

Jeudi 14 Janvier

Finition de l'étape 8, Programmation de l'étape 9, annexes et Préparation de la soutenance.

Le groupe se scinde 2 espaces de travail, le premier s'occupe de finir l'étape 8 pendant que le second commence à prévoir la soutenance du lendemain.

Plus tard, le 2 groupes travaillent respectivement sur la programmation de l'étape 9 et sur la finalisation des annexes avec l'ajout du diagrammes des tâches.