

JOURNAL DE PROGRESSION

13 décembre 2024 :

Nous avons commencé par lire le sujet en groupe pour bien comprendre ce qu'on devait faire. Ça nous a permis de clarifier les objectifs et de nous mettre d'accord sur une méthode de travail.

Nous avons réparti les tâches et décidé de découper le projet en étapes pour mieux s'organiser.

Nous avons attaqué les étapes 1, 2 et 3 :

- **Étape 1** : Affichage de l'en-tête. Nous avons eu des soucis avec les formats big endian et little endian, mais après quelques recherches, nous avons réussi à régler ça.
- **Étape 2** : Affichage de la table des sections. Cette partie s'est bien passée après avoir surmonté les problèmes de l'étape 1.
- **Étape 3** : Affichage du contenu d'une section. Nous avons commencé, mais on a rencontré des problèmes d'affichage, surtout avec les options pour choisir entre l'indice et le nom de la section.

14 décembre 2024 :

- **Étape 3** : Nous avons réglé les soucis d'affichage liés à la gestion des options. C'était surtout des erreurs d'espaces et de gestion des noms de sections. Une fois corrigés, nous avons fini cette étape.
- **Étape 4** : Début de l'affichage de la table des symboles. Il a fallu approfondir notre compréhension des fichiers ELF, mais nous avons avancé grâce à des recherches supplémentaires.

15 décembre 2024 :

- **Nettoyage du code** : Nous avons structuré et commenté le code des étapes 1 et 2 pour qu'il soit plus clair et facile à maintenir.
- **Étape 5** : Début du travail sur la table de réimplantation. C'était assez compliqué, car il fallait bien comprendre le fonctionnement des symboles et leur réimplantation dans les fichiers ELF.

17 décembre 2024 :

- **Étape 5** : Après plusieurs tests et ajustements, nous avons enfin terminé l'affichage de la table de réimplantation.
- **Programme principal** : nous avons ajouté le fichier elf_readelf avec la fonction main pour gérer les options. Le Makefile a été modifié pour inclure tout ça.

18 décembre 2024 :

- **Nettoyage du code** : Journée dédiée à simplifier et clarifier le code des étapes 3, 4 et 5. Nous avons supprimé le code inutile, ajouté des commentaires, et restructuré le tout.
- **Révision** : Nous avons relu tout le phase 1 pour vérifier s'il restait des erreurs ou des améliorations possibles. À part quelques petits ajustements, le code était propre.

19 décembre 2024 :

- **Audit de code** : Nous avons présenté notre projet au prof pour un retour. Le feedback a été positif : le code est bien structuré et compréhensible. Pas besoin de gros changements, donc nous avons pu continuer tranquillement.

30 décembre 2024 :

- **Étape 6** : Nous rencontrons des difficultés avec le renommage des sections, plusieurs problèmes ont été identifiés, notamment une confusion entre l'espace chargé en mémoire RAM et le contenu des fichiers.
- **Tests** : Nous avons commencé à tester l'étape 1 pour vérifier que tout fonctionnait bien.
- **Documentation** : Début de la rédaction des comptes rendus pour documenter ce que nous avons fait jusqu'ici.

1er janvier 2025 :

- **Tests** : Suite des tests pour les étapes 1, 2 et 3. Tout fonctionne, mais nous avons identifié quelques petits détails à améliorer.
- **Étape 6** : Nous avons des problèmes pour gérer la mise à jour des offsets des nouvelles sections, et l'encodage BE/LE pose également des difficultés. De plus, la mise à jour de la table des chaînes est problématique, car sa taille n'est connue qu'à la fin, et les offsets sont calculés en fonction de la taille de ces sections, notamment celle-ci.
- **Documentation** : nous avons terminé de décrire la phase 1 (structure du code, principales fonctions, et fichiers). Nous avons aussi fini le mode d'emploi pour expliquer comment compiler et utiliser la phase 1.

2 janvier 2025 :

- **Étape 6** : nous avons enfin terminé ! C'était dur, mais nous sommes contents du résultat.
- **Étape 7** : Correction des symboles et attribution des adresses absolues aux sections

3 janvier 2025 :

- **Étape 8 et 9** : Discussion autour de l'énoncé, et lecture de la documentation.
- **Étape 7** : Finalisation de l'étape

4 janvier 2025 :

- **Étape 6 et 7:** Réorganisation du code afin d'améliorer sa lisibilité.
- **Étape 8 et 9:** Découpage du code en modules, début du travail sur ces derniers.

5 janvier 2025 :

- **Étape 8 et 9:** Nous avons terminé les étapes 8 et 9, mais elles n'ont pas été testées car l'étape 10 n'est pas encore au point.
- **Étape 10 :** Nous avons commencé l'étape 10 , nous avons débuté par revoir le sujet du projet pour comprendre comment utiliser le simulateur.

6 janvier 2025 :

- **Automatisation des tests :** Nous avons automatisé les tests pour les étapes 1 à 7. Ça nous fait gagner du temps et permet de vérifier facilement si tout fonctionne.
- **Nettoyage des fichiers :** Nous avons trié le répertoire, supprimé les fichiers inutiles et ajouté des spécifications aux fonctions pas encore documentées.
- **Étape 10 :** Nous avons finalisé l'étape. Ce qui nous a permis de détecter les problèmes présents dans les étapes 8 et 9.
- **Étape 8-9:** Nous avons repéré de nombreux bugs liés à la réimplantation du type R_ARM_ABS*.

7 janvier 2025 :

- **Nettoyage des fichiers :** Nous avons trié le répertoire, supprimé les fichiers inutiles et ajouté des spécifications aux fonctions pas encore documentées.
- **Automatisation des tests :** On a automatisé les tests pour les étapes 1 à 7. Ça nous fait gagner du temps et permet de vérifier facilement si tout fonctionne.
- **Étape 9 :** ÇA FONCTIONNE !!!!!!!!!!!.
- **Révision :** nettoyage des codes de la phases 2 et révision de code des deux phases

8 janvier 2025 :

- **Présentation :** préparation à la soutenance et mise au propre du rendu.