

Custom ELF Loader intended for pedagogic use

build passing  62%

[GitHub Repository](#)

Telecharger

testElfReader `bash git clone https://github.com/FederAndInk/ProjetS5.git`

Pour compiler

`bash ./compile.sh`

lance les tests à la fin de la compilation: `bash ./launchTest.sh`

Executer

`bash build/src/<program>` program: - readelf - ElfLinker (not all rels completed) - ArmRun (can't be run after ElfLinker, rels not completed) - ArmRunnerExample (see ./sample_run.sh)

To do

- [x] Affichage de l'en-tête
- [x] Affichage de la table des sections et des détails relatifs à chaque section
- [X] Affichage du contenu d'une section
- [X] Affichage de la table des symboles et des détails relatifs à chaque symbole
- [X] Affichage des tables de réimplantation et des détails relatifs à chaque entrée
- [X] Renumérotation des sections
- [X] Correction des symboles
- [] Réimplantations de type RARMABS*
- [X] Réimplantation de type RARMJUMP24 et R ARM CALL
- [X] Interfaçage avec le simulateur ARM (not tested yet)
- [] Exécution à l'aide du simulateur arm-eabi-run.

Tests

`bash ./launchTest.sh ./testReadElf.sh` 0. **launchTest.sh** : script permettant de lancer les tests unitaires contenus dans les fichiers testElfWriter.c et testElfReader.c et de résumer les réussites/échecs des tests.

1. **testReadElf.sh** : script effectuant les tests fonctionnels : appel du readelf du projet avec toutes les options possibles sur tous les fichiers dans le répertoire bin et comparaison des sorties avec les sorties de la commande readelf du système.
2. **testIsElf** Sert à vérifier si le fichier lu est bien au format attendu (ELF ou non).
3. **testElfRead32** Sert à vérifier la bonne lecture d'un caractère codé sur 4 octets dans un endroit précis

4. **testElfRead16** Sert à vérifier la bonne lecture d'un caractère codé sur 2 octets dans un endroit précis (vérifiable) du fichier, notamment si c'est bien un fichier ELF, si oui : si le caractère lu est correct, si les endianness ont été bien gérées par les fonctions et si la position du curseur est correcte après la lecture (s'il est bien déplacé de 2 octets).
5. **testElfReadUC** Sert à vérifier la bonne lecture d'un caractère codé sur 1 octet dans un endroit précis (vérifiable) du fichier, notamment si c'est bien un fichier ELF, si oui : si le caractère lu est correct et si la position du curseur est correcte après la lecture (s'il est bien déplacé de 1 octet).
6. **testElfReadUCs** *Sert à vérifier la bonne lecture de plusieurs caractères codés sur 1 octet dans un endroit précis (vérifiable) du fichier (avec un offset), notamment si c'est bien un fichier ELF, si oui : si les caractères lus sont corrects et si la position du curseur est correcte après la lecture (s'il est bien déplacé de nbcaractères_lus octets).*
7. **testElfWriteUC** Sert à vérifier la bonne écriture d'un caractère codé sur 1 octet dans une section du fichier différente de celles du mot magique (les 16 premiers octets, afin de ne pas modifier la nature du fichier ELF pour de futurs tests).
8. **testElfWrite16** Sert à vérifier la bonne écriture d'un caractère connu codé sur 2 octets dans une section du fichier différente de celles du mot magique (les 16 premiers octets, afin de ne pas modifier la nature du fichier ELF pour de futurs tests) et de taille suffisamment grande pour contenir 2 octets, on vérifie aussi la bonne gestion de l'endianness.
9. **testElfWrite32** Sert à vérifier la bonne écriture d'un caractère connu codé sur 4 octets dans une section du fichier différente de celles du mot magique (les 16 premiers octets, afin de ne pas modifier la nature du fichier ELF pour de futurs tests) et de taille suffisamment grande pour contenir 4 octets, on vérifie aussi la bonne gestion de l'endianness.

Journal

Un [journal](#) à été maintenu tout au long du projet disponible sur la page du repo.

Docs

- [Elf reading](#)
- [Elf structure](#)