

Log du projet GROUPE 10

03/01/22 :

Matin:

-Prise de connaissance du sujet (tout le monde)

-- tâches :

*E1 :
Hugolin
E2 :
Ayoub
David
G  rome
E3 :
Th  o
E4 :
Antonin
E5 :
Zeinabou

Apr  s midi:

-Premi  re version de E2 : *Premier commit Etape2 :*

-- fichier :

+ affichage_table_section.c

-- fonction dans affichage_table_section.c :

```
+ Elf32_Shdr *affiche_tab_sect(FILE *f, Elf32_Ehdr header)
+ void afficher_tableau(Elf32_Shdr * Tab, Elf32_Ehdr header, FILE *f)
+ int main()
```

-Premi  re version de E1: *Etape 1 :*

-- fichier :

```
+ readHeader.c
* arm_runner_example.c
```

-- fonction dans readHeader.c :

```
+ Elf32_Ehdr * readHeader(const char * file)
```

04/01/22 :

Matin:

-Patch E1 : *small fix* :

```
* Elf32_Ehdr * readHeader(const char * file)
```

Après midi:

-Patch E2 : *static* :

```
* Elf32_Shdr *affiche_tab_sect(FILE *f, Elf32_Ehdr header)
* void afficher_tableau(Elf32_Shdr * Tab, Elf32_Ehdr header, FILE *f)
```

-Patch E1, E2 : *E1 + E2 clean* :

-- fichier :

```
- affichage_table_section.c
- readHeader.c
+ readSectionheader.c
+ readSectionheader.h
+ readELFHeader.c
+ readELFHeader.h
+ readSectionTest.sh
* arm_runner_example.c
* MakeFile.am
```

-- fonction readSectionheader.c, readSectionheader.h :

```
+ Elf32_Shdr *loadTabSectionHeader(FILE *f, Elf32_Ehdr header)
+ void printSectionHeader(Elf32_Shdr * Tab, Elf32_Ehdr header, FILE *f)
+ int readSections(const char * file, int affichage)
```

-- fonction readELFHeader.c, readELFHeader.h :

```
+ Elf32_Ehdr * readHeader(const char * file)
+ Elf32_Ehdr * readHeaderAffichage(const char * file)
```

-Première version de E3: Etape 1 + 2 + 3 :

-- fichier :

- + readSectionContent.c
- + readSectionContent.h
- * MakeFile.am
- * arm_runner_example.c
- * readSectionHeader.h
- * readSectionHeader.c

-- fonction readSectionContent.c, readSectionContent.h :

- + void readRawSectionContent(const char * f_char)

-- fonction readSectionHeader.c, readSectionHeader.h :

- int readSections(const char * file, int affichage)
- + int readSectionsHeader(const char * file, int affichage)

-Première version de E4: Etape 4 CLEAN :

-- tâches :

E4 :

Antonin
Gérome
David
Hugolin
Ayoub

-- fichier :

- * MakeFile.am
- * arm_runner_example.c
- + readSymTable.c
- + readSymTable.h
- * readSectionHeader.h
- * readSectionHeader.c

-- fonction readSymTable.c, readSymTable.h :

- + Elf32_Sym *loadSymTable(FILE *f, Elf32_Shdr *sectionHeader, Elf32_Ehdr header, int * nb_symboles)
- + void printSymTable(Elf32_Sym * Tab, Elf32_Shdr * sectionHeader, Elf32_Ehdr header, FILE *f, int nb_symboles)
- + Elf32_Sym * readSymTable(const char * file, int affichage)

-- fonction readSectionHeader.c, readSectionHeader.h :

- int readSectionsHeader(const char * file, int affichage)
- + Elf32_Shdr *readSectionsHeader(const char * file, int affichage)

05/01/22 :

-- tâches :

E5:
Zeinabou
David
Hugolin
G  rome

-Premi  re version de E5: *Etape 5 CLEAN :*

-- fichier :

```
* MakeFile.am
* arm_runner_example.c
* readSymTable.c
+ readRelocTable.c
+ readRelocTable.h
+ readRelocTable.sh
* readSectionHeader.c
```

-- fonction readRelocTable.C, readRelocTable.h :

```
+ Elf32_Rel ** loadRelocTable(FILE *f, Elf32_Shdr * sectionHeader,
Elf32_Ehdr header, TableauSectionReloc * tabSecRel)
+ void printRelocTable(FILE *f, Elf32_Rel ** tabRel, Elf32_Sym * Tab,
Elf32_Shdr * sectionHeader, Elf32_Ehdr header, TableauSectionReloc *
tabSecRel)
+ Elf32_Rel **readRelocTable(const char * file, int affichage)
```

-- fonction readSymTable.c :

```
* Elf32_Sym *loadSymTable(FILE *f, Elf32_Shdr *sectionHeader,
Elf32_Ehdr header, int * nb_symboles)
```

-Premi  re version de E6: *Etape 6 temporaire :*

-- fichier :

```
* MakeFile.am
* LaunchTest.sh
* arm_runner_example.c
* readSectionContent.c
+ renumSection.c
+ renumSection.h
```

-- fonction readSectionContent.c :

```
* void readRawSectionContent(const char * f_source)
```

-- fonction renumSection.c, renumSection.h :

```
+ void writeSectionContentFile(const char * f_source, const char *  
f_dest)
```

06/01/22 :

-- tâches :

A partir de ce moment, les tâches sont réalisées tel que :

(*Codeur* : étudiant qui rédige du code, *Soutien*: étudiant aidant les étudiants *Codeur*,

Recherche : étudiant recherchant dans les docs)

Codeur :

Gérome

David

Hugolin

Soutien :

Zeinabou

Ayoub

Recherche :

Antonin

Théo

-Avancement E6: Etape 6 temporaire2 :

-- fichier :

```
* renumSection.c
```

```
* renumSection.h
```

-- fonction renumSection.c, renumSection.h :

```
* void writeSectionContentFile(const char * f_source, const char *  
f_dest)
```

07/01/22 :

-Optimisation Global: Etape 6 ULTRA CLEAN :

-Ajout d'une structure à passer dans les paramètres des fonctions, pour éviter la duplications de ligne de code.

-Ajout d'un "main" des fonctions ajoutée, séparer de ARM_runner_example

(ARM_runner.c)

-Renommage des fonctions

-Meilleur gestion de la mémoire

10/01/22 :

-E6 final et première version de E7: Etape 6 NOT SO CLEAN :

-- fichier :

```
* MakeFile.am
* arm_runner_example.c
+ implantation.c
+ implantation.h
```

-- fonction implantation.c, implantation.h :

```
* void deleteRel(Elf32_Main * ELF)
* void correctSymTable(FILE * f, Elf32_Main * ELF, uint32_t text,
uint32_t data)
```

11/01/22 :

-Optimisation général: refactoring main:

-- fichier :

```
* ARM_runner.c
* implantation.c
* readHeaderTest.sh
* readRelocTable.sh
* readSectionContent.c
* readSectionTest.sh
* readSymTable.sh
```

-- fonction readSectionContent.c :

```
* void printSectionContent(Elf32_Main * ELF)
```

-- fonction implantation.c :

```
* void deleteRel(Elf32_Main * ELF)
```

-Optimisation et Commentaire : Nettoyage & commentaire & Debug :

-- fichier :

tous fichiers et fonction rédigées par les étudiants *Codeur*

12/01/22 :

-Rédaction des documents demandés