

# Liste des Tests

## Comment exécuter des tests et les scripts !?

- 1 - Se positionner dans le dossier **PROG5**
- 2 - Pour compiler tous les tests assembleurs exécutez **./compile\_tests.sh** (peut necessiter les permissions avant l'exécution : **chmod +x compile\_tests.sh**)
- 3 - Pour tester chaque étape exécutez **./tests.sh <num étape>**
- 4 - Pour supprimer les fichiers **.o** et **.o\_modified** des dossiers contenant les tests exécutez :  
**./supprimer\_objet\_tests.sh**

**NOTE** : pour avoir les **.o\_modified** dans les tests respectivement des étapes 6 ou 7 , exécutez **./tests/TEST\_ETAPE6/supprimer\_REL.sh** ou **./tests/TEST\_ETAPE7/supprimer\_REL.sh**

## Etape 1 - Affichage de l'en-tête

### **Objectif**

Vérifier que notre programme peut lire et afficher correctement les en-têtes des fichiers ELF valide pour ARM 32 bits en le comparant avec l'affichage de readelf.

### **Façon de Conclusion**

Si les résultats sont identiques, le test est réussi. En cas de différences, le script indique la différence en rouge.

## Etape 2 - Affichage des sections ELF

### **Objectif**

Vérifier l'affichage correct des sections **.text**, **.data**, **.symtab**, **.strtab**, ...etc

### **Façon de Conclusion**

Le script signale les différences en rouge (erreurs) et en vert les éléments communs.

## Etape 3 - Affichage du contenu d'une section

### **Objectif**

Vérifier l'affichage correct du contenu d'une section

### **Façon de Conclusion**

Le script signale les différences en rouge (erreurs) et en vert les éléments communs.

## Etape 4 : Comparaison des Tables des Symboles

### **Objectif**

Comparer les tables des symboles extraites par readelf et elf\_readelf.

### **Façon de Conclusion**

Le script signale les différences en rouge (erreurs) et en vert les éléments communs.

## Etape 5 - Affichage des tables de réimplantation

### **Objectif**

Lire et Afficher les tables de réimplantation d'un fichier ELF pour machine ARM

### **Façon de Conclusion**

Le script signale les différences en rouge (erreurs) et en vert les éléments communs.

## Etape 6 - Renumérotation des sections

### **Objectif**

Ces tests valident la bonne génération et la cohérence d'un fichier ELF, depuis la gestion des sections .text, .data et .bss, en passant par l'appel de fonctions externes, la référence croisée entre sections, la détection d'un fichier vide.

### **Façon de Conclusion**

Les tests sont réussis si la suppression des réimplantations est correctement gérée.

## Etape 7 - Correction des symboles

### **Objectif**

Ces tests valident la gestion de symboles simples, de multiples sections aux adresses spécifiques et la détection d'erreurs intentionnelles.

### **Façon de Conclusion**

Les tests sont réussis si les sections sont correctement gérées.