

Cahier des charges pour la réalisation d'un programme de *reverse engineering*

Jean-Michel Richer
M1 Informatique, Génie Logiciel

23 août 2007

1 But du projet

Etant donné un fichier exécutable sous Linux on désire faire du reverse engineering en le désassemblant et en étudiant son code source. On veut notamment obtenir un arbre d'appel des sous-programmes afin de comprendre la structure du programme. Pour obtenir le code assembleur d'un exécutable sous Linux, on peut utiliser la commande suivante :

```
objdump -d -M intel -r -j .text monexe >code.asm
```

On obtient alors le résultat suivant :

```
program:      file format elf32-i386
```

```
Disassembly of section .text:
```

```
8049e20: 55                push   ebp
8049e21: 89 e5            mov    ebp,esp
8049e23: 51              push   ecx
8049e24: 51              push   ecx
8049e25: 8b 15 54 1d 17 08 mov    edx,DWORD PTR ds:0x8171d54
8049e2b: 85 d2            test   edx,edx
8049e2d: 74 19            je     8049e48 <strcpy@plt+0xd0>
8049e2f: b8 00 00 00 00  mov    eax,0x0
8049e34: 85 c0            test   eax,eax
8049e36: 74 10            je     8049e48 <strcpy@plt+0xd0>
8049e38: 83 ec 0c        sub    esp,0xc
8049e3b: 68 54 1d 17 08  push   0x8171d54
8049e40: e8 bb 61 fb f7  call   0 <_init-0x80496c0>
8049e45: 83 c4 10        add    esp,0x10
8049e48: c9              leave
8049e49: c3              ret
...
```

On veut afficher :

- l'arbre d'appel des sous-programmes, indiquant pour un sous-programme donné, quels autres sous-programmes il appelle.
- pour chaque sous-programme on affichera :
 - l'adresse de début et de fin de sous-programme ainsi que la taille du sous-programme en nombre d'octets
 - les sous-programmes appelés ainsi que le nombre d'appels
 - s'il s'agit d'un sous-programme récursif ou non
 - si le sous-programme utilise des instructions x87
 - si le sous-programme utilise des instructions sse

On vérifiera notamment la cohérence des informations fournies avant l’affichage, c’est à dire qu’il n’existe pas de sous-programmes dont les adresses se chevauchent ou qu’un sous-programme est compris dans un autre.

2 Contraintes

Votre programme exécutable devra s’appeler `asmrengine` et devra accepter des paramètres optionnels en ligne de commande comme suit :

```
asmrengine [options] input-file
```

- `--exclude-size-over=n`, où n est un entier positif, permet de ne pas afficher les sous-programmes dont la taille est supérieure à n octets
- `--exclude-size-under=n`, où n est un entier positif, permet de ne pas afficher les sous-programmes dont la taille est inférieure à n octets
- `--order-by=m`, où m prend les valeurs `size` ou `address`, permet d’afficher les sous-programmes par ordre croissant de taille ou d’adresse
- `--leaves-only`, affiche uniquement les informations concernant les sous-programmes qui n’appellent pas d’autres sous-programmes

3 Organisation

le déroulement des opérations est le suivant :

3.1 rapport préliminaire

Avant le 15 décembre 2007, vous devrez rendre un *rapport préliminaire de conception* de votre application, c’est à dire que vous expliquerez comment vous allez concevoir votre programme. Vous devrez notamment faire figurer les points suivants :

1. architecture logicielle de votre application :
 - organisation modulaire
 - diagramme de classe
 - description de chaque classe (données membres) et rôle de chaque méthode
2. analyse spécifique :
 - expliquez comment vous allez identifier le début et la fin d’un sous-programme, comment procéderez vous si vous détectez deux fois consécutivement le début d’un sous-programme ?
3. planning de travail pour chaque membre de l’équipe
 - de septembre à décembre
 - puis planning prévisionnel de décembre à janvier

Vous recevrez un message de confirmation de ma part indiquant que j’ai bien effectivement reçu votre rapport

3.2 logiciel final

Pour le 20 Janvier 2008 dernier délai, vous devrez me rendre

- les sources de votre programme écrit en C++
- la documentation générée par doxygen

Votre programme devra pouvoir être compilé sous tout système Unix/Linux disposant de g++ en tapant simplement la commande `make`

4 Notation

Vous serez sanctionné sévèrement dans les cas suivants :

- non remise du rapport préliminaire en temps et en heures : -10 pts
- programme ne répondant pas aux exigences : -8 pts
- impossibilité d'obtenir à terme l'exécutable lors de la compilation : -5 pts
- programme ne fonctionnant pas (notamment segmentation fault) : -8 pts
- récupération des sources d'un autre groupe : vous obtiendrez la note de 0

5 Points divers

1. le programme doit bien évidemment être écrit en C++ et utiliser la STL pour la gestion des containers.
2. vous travaillerez sur un exemple comprenant plus de 307.000 lignes de code assembleur
3. vous devrez travailler par groupe de 5 personnes maximum, et vous élirez parmi vous un chef de projet qui sera chargé d'organiser le travail du groupe et me tiendra régulièrement au courant de l'avancement du projet (tous les 15 jours environ) en m'envoyant un mail :

To : `richer@info.univ-angers.fr`
Subject `M1INFO-GL:<identifiant-projet>`

Il aura également la responsabilité de m'informer des problèmes que rencontre le groupe et notamment des membres qui ne travailleraient pas, comme cela arrive tous les ans. L'appartenance au groupe implique la participation au travail!

May the force be with you!