



Programmation Système Unix

fttrace



Contents

Description	2
Usage	3
Rendu	4
Autres	5
Idées de bonus	6



Description

Ftrace permet de lister les différentes entrées et sorties de fonction d'un programme. Ainsi, vous devez lister les :

- appels système ;
- appels de fonctions internes au programme avec leur nom et leur adresse ;
- signaux reçus des autres programmes ;
- appels de fonctions contenues dans les bibliothèques partagées (.so) ;

Ces informations doivent être affichées de la manière suivante :

```
1  Entering function main at 0x42ff231
2  Entering function my_putstr at 0x42ff9fd
3  Entering function my_putchar at 0x43aa123
4  Syscall write(0x1, 0xff3210123, 0x1) = 0x1
5  Leaving function mu_putchar
6  Entering function my_putchar at 0x43aa123
7  Syscall write(0x1, 0xff3210124, 0x1) = 0x1
8  Leaving function mu_putchar
9  Entering function my_putchar at 0x43aa123
10 Syscall write(0x1, 0xff3210125, 0x1) = 0x1
11 Received signal SIGWINCH
12 Leaving function mu_putchar
13 Entering function my_putchar at 0x43aa123
14 Syscall write(0x1, 0xff3210126, 0x1) = 0x1
15 Leaving function mu_putchar
16 Leaving function my_putstr
17 Entering function printf at 0x877621fda312
```

Suivant les éléments dont vous disposez, l'affichage peut se restreindre, par exemple si l'exécutable que vous invoquez ne dispose pas de table des symboles. Vous devez tout de même suivre les appels de fonctions et en afficher une description. Par exemple :

func_0x8765FDE0@a.out.



Usage

`./ftrace <commande>`



Rendu

- Les sources doivent être rendues dans le répertoire `PSU_year_ftrace`.
ex: `PSU_2015_ftrace` pour l'année 2015-2016
- L'intégralité de votre projet devra compiler avec un unique *Makefile* comportant (au moins) les règles `all`, `clean`, `fclean` et `re`.
- L'exécutable sera nommé `ftrace`.
- Ce projet est à réaliser en groupe de 2 à 3 personnes.
- Votre rendu devra contenir un fichier `auteur` avec les logins de chaque membre du groupe, séparés par un ;
- Le groupe Yammer `PSU_promo` est le groupe de discussion officielle du projet. Des informations officielles peuvent y être données.



Autres

- Vous devrez faire ce projet au moins sur *x86-64/Linux*.
- Bibliothèques autorisées : libc, libelf, libm.
- Langage autorisé : C.



Idées de bonus

- Afficher le code des fonctions appelées quand les symboles de débogage le permettent (niveaux de verbosité).
- Décoloration (*unmangling*) des noms de fonction.
- Décodage explicite des adresses de `CALL`.
- Fonctionner également en 32-bit (*x86*).
- Être portable sur différentes architectures matérielles (ARM, PowerPC, SPARC, etc.).
- Être portable sur différents systèmes.
- Lorsque le programme envoie un signal à un autre programme, vous pouvez commencer à tracer l'autre programme.
- Générer un call-graph