{EPITECH}

FTRACE

ANALYZING EXECUTABLES

FTRACE

Preliminaries

binary name: ftrace
language: C
compilation: via Makefile, including re, clean and fclean rules

- ✓ The totality of your source files, except all useless files (binary, temp files, objfiles,...), must be included in your delivery.
- ✓ All the bonus files (including a potential specific Makefile) should be in a directory named bonus.
- ✓ Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

You must complete this project on at least x86-64/Linux.

The following libraries are allowed: **libc**, **libelf**, **libm**.

As you know, **ftrace** allows to list all of the different inputs and outputs of a program's function. Therefore, you must list the following:

- ✓ system calls,
- ✓ a program's internal function calls with their name and address,
- ✓ signals received from other programs,
- ✓ function calls contained in the shared libraries (.so).

This information must be displayed in the following way:

```
Terminal
~/B-PSU-400> ./ftrace ./a.out >&-
Entering function main at 0x42ff231
Entering function my_putstr at 0x42ff9fd
Entering function my_putchar at 0x43aa123
Syscall write(0x1, 0xff3210123, 0x1) = 0x1
Leaving function my_putchar
Entering function my_putchar at 0x43aa123
Syscall write(0x1, 0xff3210124, 0x1) = 0x1
Leaving function my_putchar
Entering function my_putchar at 0x43aa123
Syscall write(0x1, 0xff3210125, 0x1) = 0x1
Received signal SIGWINCH
Leaving function my_putchar
Entering function my_putchar at 0x43aa123
Syscall write(0x1, 0xff3210126, 0x1) = 0x1
Leaving function my_putchar
Leaving function my_putstr
Entering function printf at 0x877621fda31
...
```

According to the available elements, the display can limit itself, for example, if the executable that you call does not have a table of symbols.
However, you must follow the function calls and display a description.
For example: **func_0x8765FDE0@a.out**.

```
Terminal
~/B-PSU-400> ./ftrace -help
USAGE: ftrace <command>
```

Here is a list of possible **bonus** points:

- ✓ Display the code of the called functions, when debugging symbols allow (level of verbosity)
- ✓ Unmangling function names
- ✓ Explicit decoding of CALL addresses
- ✓ 32-bit (x86) comptability
- ✓ Compatible on different material structures (ARM, PowerPC, SPARC etc.)
- ✓ Compatible with different systems
- ✓ When the program sends a signal to another program, trace the other program too
- ✓ Generation of a call-graph

v 3.2