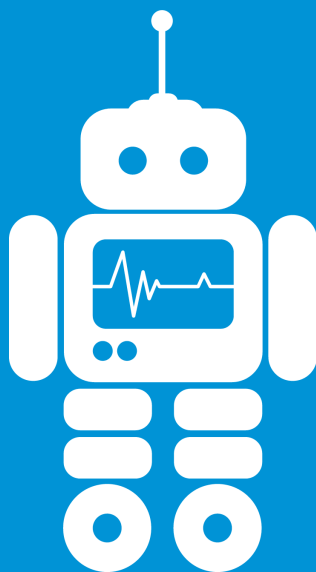




MINILIBC

GOING DEEPER INTO THE RABBIT HOLE



MINILIBC

Preliminaries



binary name: libasm.so
language: x86-64 Assembly
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.

The objective of this project is to create a dynamic ELF library to replace (to a certain extent) the standard C library you use every day on your system.

Thanks to the bootstrap, you are now able to replace some functions with your own implementations through a dynamic library (which is called *weak binding*).



`LD_PRELOAD`



We will consider an approximate understanding of your own as proof of cheating. Instead, we urge you to commit broken code instead of copying code you don't really understand.
You have been warned!

Here are the functions to be implemented in your MiniLibC:

- ✓ strlen
- ✓ strchr
- ✓ strrchr
- ✓ memset
- ✓ memcpy
- ✓ strcmp
- ✓ memmove
- ✓ strncmp
- ✓ strcasecmp
- ✓ strstr
- ✓ strpbrk
- ✓ strcspn



Refer to the respective **man** pages of the aforementioned functions for behavior specifications.



You must stick to them.

Boni

Here are some suggestions for potential bonus points:

- ✓ additional interesting functions: *ffs*, *syscall*, *strfry*, *index*, *rindex*, *strtok*, *memfrob*...;
- ✓ additional architectures (32-bit x86 or ARM, for example);
- ✓ anything **useful** that comes to mind.

{EPITECH}

