B4- x86_64 Assembler

B-PSU-360

MiniLibC

Going deeper into the rabbit hole

v0.3

MiniLibC

Going deeper into the rabbit hole

binary name:	libasm.so
repository name:	asm_minilibc
repository rights:	ramassage-tek
language:	x86-64 Assembly
group size:	1-2
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- 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).

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 `malloc` project and 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`
- `memset`
- `memcpy`
- `strcmp`
- `memmove`
- `strncmp`
- `strcasecmp`
- `rindex`
- `strstr`
- `strpbrk`
- `strcspn`

Refer to the respective `man` pages of the aforementioned functions for behavior specifications.
You must stick to them.

Bonuses

Here are some suggestions for potential bonus points:

- additional functions: `read`, `write`, ...;
- additional architectures (32-bit x86 or ARM, for example);
- anything **useful** that comes to mind.

As you know, no bonuses will be evaluated unless you have a fully functional mandatory part.