b_printf

Because i'm tired of using putnbr and putstr.

Gaetan gaetan@42.us.org

Summary: Summary: By now, you must be fed up alternating the ft_putstr and ft_putnbr functions when doing your printing. Are you not allowed to use printf? You can recode yours! It will be a chance for you to discover a feature of the C language – variadic functions – and to practice detailed management of the printing options. You will then be allowed to use your b_printf in all your future projects.

Contents

Chapter I

Foreword

Here is a list of motivational posters you can find in the office of Barney Stinson in the various seasons of How I Met Your Mother:

- Awesomeness : "When I get sad, I stop being sad and be Awesome Instead. True Story. Barney Stinson"

- Conformity : "It's the one who is different that gets left out in the cold."

- Courage : "True greatness comes when you're tested. Theodore Roosevelt"

- Challenge : "We either find a way or we make one"

- Opportunity : "You will always miss 100

- Teamwork : "Coming together is the beginning. Keeping together is progress. Working together is success. Henry Ford."

- Teamwork : "The chain is only as strong as the weakest link"

- Perseverance : "Continuous effort is the key to unlocking your potential. Sir Winston Churchill"

- Strength : "What the mind can conceive, it can achieve"

Chapter II

Introduction

Regardless of the programming language considered, the printf function,(or its equivalents) is always highly useful. The main reason is the ease of its formatting, and the support of diverse types in variable numbers. Some variations even propose to be able to write the resulting string of characters either in a file descriptor or in a particular stream. Some also propose to recall this string without printing it. In short, undeniably, printf is a vital function. In this project, we ask you to recode it and add it to your libft so that you can use it in all your future projects.

Chapter III

Goals

The versatility of the printf function in C represents a great exercise in programming for us. This project is of moderate difficulty. It will enable you to discover variadic functions in C in a particularly relevant context as well as learn about a great example of a basic "dispatcher" in C via the use of an array of function's pointers.

The key to a successful b_printf is a well-structured and good extensible code. This is because as time goes by, you will be more and more tempted to continue to extend your b_printf, making your life easier for your future projects. So take your time to code properly while keeping in mind that you will have to read again your code in a few weeks or a few months to extend its functionality according to your needs. It would be a shame not to being able to do it because your can't read your own work. What do you think?

Chapter IV

General instructions

- Your function must be called `b_printf`.

- Your project must be written in accordance with the Norm.

- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).

- Your must submit a Makefile which will compile a libftprintf.a. This lib will be linked to our testing main to give you your results.

- You'll have to submit a file called author containing your username followed by a '\n' at the root of your repository

```
$>cat -e author
xlogin$
```

- You are allowed to use the following functions:

    ○ `write`

    ○ `setlocale`

    ○ `malloc`

    ○ `free`

    ○ `exit`

    ○ The functions of `man 3 stdarg`

- You can ask your questions on the forum.

Chapter V

Mandatory part

- You have to recode the libc's `printf` function.

- Your function will be called `b_printf` and will be prototyped similarly to `printf`.

- You won't do the buffer management in the `printf` function.

- You have to manage the following conversions: `scdipoux`

man 3 printf / man 3 stdarg

Chapter VI

Submission and peer-evaluation

Submit your work on your GiT repository as usual. Only the work on your repository will be graded.