Elementary Programming

get_next_line

Teacher in charge
younes2.serraj@epitech.eu
Last update
26/11/2015_18h13

Contents

Administrative details

- Name of the turn-in repository: CPE_year_getnextline
  `Example for the school year 2015-2016 : CPE_2015_getnextline`

> ⚠ `Be careful: the norm will be checked for every single file you turn in`

Subject

- The goal of this project is to write a function that returns a line read from a file descriptor.

- You must define a macro called READ_SIZE in your get_next_line.h file as it is defined below (including the #ifndef protection).

    - This macro indicates the number of characters to read at each call of read().
    - The value 14 used below is given here just as an example. You can use the value you want as long as it's a strictly positive number.
    - The value of this macro will be changed (by another strictly positive value) during the evaluation to make sure that you are using it correctly.

```
# ifndef READ_SIZE
#  define READ_SIZE (14)
# endif /* !READ_SIZE */

```

- You must use one (or several) static variable(s) to save the characters that were read but not sent back to the calling function.

- You shall turn in two files get_next_line.c and get_next_line.h that must be located at the root of your repository.

- Your repository shall not contain any Makefile nor main function.

- The macro READ_SIZE, as well as the get_next_line function prototype, must be located in a file named get_next_line.h

- get_next_line shall return its result <u>without</u> the \n. If there is nothing more to read on the file descriptor, or if an error occured while reading, the function returns NULL.

- The function shall be prototyped as follows:

```
char    *get_next_line(const int fd);

```

- Turn in:
  get_next_line.h and get_next_line.c

- Example of how to use the get_next_line function:

```
/*
** main.c for get_next_line_main_test in
**
** Made by tek assistant
** Login   <astek@epitech.net>
**
** Started on  Mon Nov  5 14:59:09 2001 tek assistant
** Last update  Mon Nov  5 14:59:09 2001 tek assistant
*/

#include "my.h"
#include "get_next_line.h"

int     main()
```

```
{
  char  *s;

  while ((s = get_next_line(0)))
    {
      my_putstr(s);
      my_putchar(’\n’);
      free(s);
    }
  return (0);
}

```

Authorized functions

- read
- malloc
- free