

# Lire une ligne d'un fichiers en C

par [Nicolas Joseph](#)

Date de publication : 19 Mars 2005

Dernière mise à jour : 10 Aout 2005

Cette fonction a pour but de lire la prochaine ligne du flux de sortie stream et de la placer dans une zone de mémoire allouée dynamiquement dont l'adresse est stockée dans la variable `pp_line`. En cas d'erreur, la fonction retourne la valeur `EXIT_FAILURE`. Si le la fichier du fichier est atteinte, `pp_line` est mis à `NULL` et la fonction retourne `EXIT_FAILURE`.



## getNextLine.h

```

#ifndef H_NJ_GETNEXTLINE_1903052321
#define H_NJ_GETNEXTLINE_1903052321

#include <stdio.h>

#define skipNextLine(file) getNextLine ((file), NULL)

int getNextLine (FILE *stream, char **pp_line);

#endif /* not H_NJ_GETNEXTLINE_1903052321 */

```

## getNextLine.c

```

#include <stdlib.h>
#include "getNextLine.h"

int getNextLine (FILE *stream, char **pp_line)
{
    int ret = EXIT_SUCCESS;

    if (stream)
    {
        int end = 0;
        char *line = NULL;
        size_t size = 1;
        int i = 0;

        while (ret == EXIT_SUCCESS && end == 0)
        {
            int c = fgetc (stream);
            switch (c)
            {
                case EOF:
                    if (feof (stream))
                    {
                        if (pp_line && !line)
                        {
                            *pp_line = NULL;
                            ret = EXIT_FAILURE;
                        }
                        end = 1;
                    }
                    else
                    {
                        ret = EXIT_FAILURE;
                    }
                    break;
                default:
                    if (pp_line)
                    {
                        if (i == size - 1)
                        {
                            void *tmp = NULL;

                            size <= 1;
                            tmp = realloc (line, sizeof (*line) * size);
                            if (tmp)
                            {
                                line = tmp;
                            }
                            else
                            {
                                free (line);
                                line = NULL;
                                ret = EXIT_FAILURE;
                            }
                        }
                    }
                    if (line)
                    {
                        if (c == '\n')
                        {
                            end = 1;
                        }
                        else
                        {
                            line[i] = c;
                            i++;
                        }
                    }
            }
        }
    }
}

```

## getNextLine.c

```
        } }
        } }
        break;
    }
}
if (pp_line && line)
{
    line[i] = '\0';
    *pp_line = line;
}
else
{
    ret = EXIT_FAILURE;
}
return (ret);
}
```

Voic un exemple d'utilisation de cette fonction dans une boucle pour lire l'intégralité d'un fichier :

```
char *p_line = NULL;
FILE *file = fopen (...);

while (getNextLine (file, &p_line) && p_line)
{
    ...
    free (p_line);
    p_line = NULL;
}
```