

`get_next_line`

Generated by Doxygen 1.8.16

1 File Index	1
1.1 File List	1
2 File Documentation	1
2.1 get_next_line.c File Reference	1
2.1.1 Function Documentation	2
2.2 get_next_line.h File Reference	3
2.2.1 Macro Definition Documentation	4
2.2.2 Function Documentation	4
2.3 README.md File Reference	5
Index	7

1 File Index

1.1 File List

Here is a list of all files with brief descriptions:

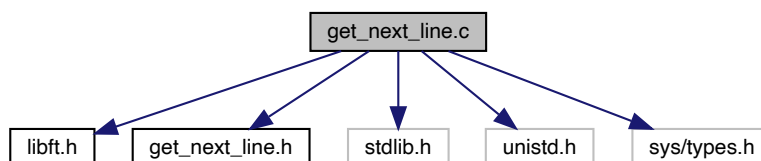
get_next_line.c	1
get_next_line.h	3

2 File Documentation

2.1 [get_next_line.c File Reference](#)

```
#include "libft.h"
#include "get_next_line.h"
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
```

Include dependency graph for `get_next_line.c`:



Functions

- `int get_next_line (const int fd, char **line)`

Reads and returns a line read from a line descriptor.

2.1.1 Function Documentation

2.1.1.1 `get_next_line()` `int get_next_line (`
 `const int fd,`
 `char ** line)`

Reads and returns a line read from a line descriptor.

Reads a line from a file descriptor, where a line is defined as a succession of characters that end with '`\n`' or with `EOF`. The function can read from a file, the standard output, redirection etc, as well as multiple file descriptors simultaneously.

Parameters

in	<i>fd</i>	The file descriptor used to read.
out	<i>line</i>	The address pointing to a string that is dynamically allocated by the function and used to save the line read from the file descriptor

Return values

<i>1</i>	a line has been read
<i>0</i>	reading has been completed
<i>-1</i>	an error happened

Note

The line is returned without '`\n`'.

Warning

The function has an undefined behaviour if, between two calls, the same file descriptor designs two distinct files although the reading from the first file was not completed, or if a call to `lseek (2)` was made.

Remarks

- The maximum number of file descriptors supported is controlled by the macro `MAX_FD`.
- The size of the reading buffer is defined by the macro `BUFF_SIZE`.

See also

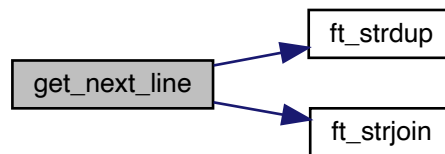
[BUFF_SIZE](#)

[MAX_FD](#)

Definition at line 90 of file `get_next_line.c`.

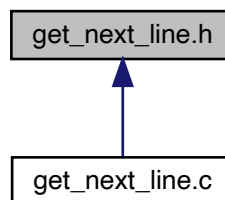
References [BUFF_SIZE](#), [ft_strdup\(\)](#), [ft_strjoin\(\)](#), and [MAX_FD](#).

Here is the call graph for this function:



2.2 `get_next_line.h` File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define MAX_FD 512`
- `#define BUFF_SIZE 32`

Functions

- `int get_next_line (const int fd, char **line)`
Reads and returns a line read from a line descriptor.

2.2.1 Macro Definition Documentation

2.2.1.1 **BUFF_SIZE** `#define BUFF_SIZE 32`

Controls the size of the read buffer. Increasing it can lead to a speed-up due to a reduced number of system calls, but will increase the function memory usage.

Definition at line 27 of file `get_next_line.h`.

2.2.1.2 **MAX_FD** `#define MAX_FD 512`

Controls the largest file description, as well the overall number of file descriptors, that can be simultaneously processed.

Definition at line 20 of file `get_next_line.h`.

2.2.2 Function Documentation

2.2.2.1 **get_next_line()** `int get_next_line (` `const int fd,` `char ** line)`

Reads and returns a line read from a line descriptor.

Reads a line from a file descriptor, where a line is defined as a succession of characters that end with '`\n`' or with `EOF`. The function can read from a file, the standard output, redirection etc, as well as multiple file descriptors simultaneously.

Parameters

in	<i>fd</i>	The file descriptor used to read.
out	<i>line</i>	The address pointing to a string that is dynamically allocated by the function and used to save the line read from the file descriptor

Return values

<i>1</i>	a line has been read
<i>0</i>	reading has been completed
<i>-1</i>	an error happened

Note

The line is returned without ' \n '.

Warning

The function has an undefined behaviour if, between two calls, the same file descriptor designs two distinct files although the reading from the first file was not completed, or if a call to `lseek(2)` was made.

Remarks

- The maximum number of file descriptors supported is controlled by the macro `MAX_FD`.
- The size of the reading buffer is defined by the macro `BUFF_SIZE`.

See also

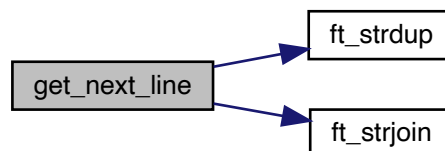
[BUFF_SIZE](#)

[MAX_FD](#)

Definition at line 90 of file `get_next_line.c`.

References `BUFF_SIZE`, `ft_strdup()`, `ft_strjoin()`, and `MAX_FD`.

Here is the call graph for this function:

**2.3 README.md File Reference**

Index

BUFF_SIZE
 [get_next_line.h](#), 4

[get_next_line](#)
 [get_next_line.c](#), 2
 [get_next_line.h](#), 4

[get_next_line.c](#), 1
 [get_next_line](#), 2

[get_next_line.h](#), 3
 BUFF_SIZE, 4
 [get_next_line](#), 4
 MAX_FD, 4

MAX_FD
 [get_next_line.h](#), 4

[README.md](#), 5