File manager

v1.0.0

Generated by Doxygen 1.8.1.2

Thu Jun 29 2017 15:29:07

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 include/func.h File Reference

Functions

- void ∗∗ draw_menu (int start_col)
- void print (WINDOW ∗win, char ∗∗filenames, int count, int user_pos, int color_bg)
- void no_print_line (WINDOW ∗win, char ∗∗filenames, int count)
- char ∗∗ direct (char ∗dp, int ∗count)
- void color_pair (WINDOW ∗win, int color_bg)
- void init_win ()
- void window ()

2.1.1 Function Documentation

2.1.1.1 void color_pair (WINDOW ∗ *win,* int *color_bg*)

Color_pair - function that initializes the color palette. Specifies the color template used to decorate the windows.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.1.1.2 char∗∗ direct (char ∗ *dp,* int ∗ *count*)

Direct - the function reads the contents of the directory and stores it in an array (in lexicographical order).

Parameters

dp	- the parameter that stores the path of a directory whose contents need to be read
count	- parameter that is responsible for the number of files in the directory

Returns

an array of file names(filenames)

2.1.1.3 void** draw_menu (int *start_col*)

Draw_panel - creation of upper and lower panels

Parameters

start_col	- color panel parameter

Returns

EXIT_SUCCESS in case of success

2.1.1.4 void init_win ( )

Init_win - initializing library ncurses

2.1.1.5 void no_print_line (WINDOW * *win*, char ** *filenames*, int *count*)

No_print_line - the function displays a list of files without highlighting the user's position. Used for output in a passive window.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
filenames	- parameter that passes the elements of the array that are needed display
count	- number of array elements

2.1.1.6 void print (WINDOW * *win*, char ** *filenames*, int *count*, int *user_pos*, int *color_bg*)

Print - output files to the screen with the highlight of the position on which the user is located. Used for the active window.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
filenames	- tarameter that passes the elements of the array that are needed display
count	- number of array elements
user_pos	- the parameter responsible for the position in which the user is located
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.1.1.7 void window ( )

Window - the function is responsible for creating and drawing windows, and also calls the function to read directories, which displays all the files in it in the manager windows. Here you can also navigate the file manager (KEY_UP, KEY_DOWN, TAB, ENTER).

2.2 text_editor/func.h File Reference

Functions

- void cycle (WINDOW *my_win, int user_pos_X, int user_pos_Y, int row, int col, char *filename, int count)
- void write_file (char *filename, char **string, int count, WINDOW *my_win)

- char ** read_file (char *filename, int *count)
- void color_pair (WINDOW *win, int color_bg)
- void window (char *filename)

2.2.1 Function Documentation

2.2.1.1 void color_pair (WINDOW * *win*, int *color_bg*)

Color_pair - function that initializes the color palette. Specifies the color template used to decorate the windows.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.2.1.2 void cycle (WINDOW * *my_win*, int *user_pos_X*, int *user_pos_Y*, int *row*, int *col*, char * *filename*, int *count*)

Cycle - performs processing of keystrokes.

Parameters

my_win	- The parameter that passes the pointer to the window. Needed in order to use the move cursor command
user_pos_X	- the parameter responsible for the position of the user in the rows
user_pos_Y	- the parameter responsible for the position of the user in the column
row	- parameter, which stores the number of rows
col	- parameter, which stores the number of columns
filename	- the parameter contains the name of the file that you want to open
count	- the parameter that counts the number of lines in a file

2.2.1.3 char** read_file (char * *filename*, int * *count*)

Read_file - the function reads each line of the file and writes it to an array of strings.

Parameters

filename	- the parameter contains the name of the file that you want to open
count	- the parameter that counts the number of lines in a file

Returns

an array of strings

2.2.1.4 void window (char * *filename*)

Window - creates a window in which the contents of the file are output

Parameters

filename	- the parameter contains the name of the file that you want to open

2.2.1.5 void write_file (char * *filename*, char ** *string*, int *count*, WINDOW * *my_win*)

Write_file - function allows you to enter words from the keyboard that will be written to a file

Parameters

filename	- the parameter contains the name of the file that you want to open
string	- array of new lines to be written to the file
count	- the parameter that counts the number of lines in a file
my_win	- The parameter that passes the pointer to the window. Needed in order to use the move cursor command

2.3 lib/func.c File Reference

```
#include <stdlib.h>
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <curses.h>
#include <stdio.h>
#include <malloc.h>
#include <dirent.h>
#include <string.h>
#include "../include/func.h"
#include <unistd.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <sys/types.h>
```

Macros

- #define _SVID_SOURCE
- #define MAX_LENGTH 255

Functions

- void ** draw_menu (int start_col)
- void print (WINDOW *win, char **filenames, int count, int user_pos, int color_bg)
- void no_print_line (WINDOW *win, char **filenames, int count)
- char ** direct (char *dp, int *count)
- void color_pair (WINDOW *win, int color_bg)
- void init_win ()
- void window ()

2.3.1 Macro Definition Documentation

2.3.1.1 #define _SVID_SOURCE

2.3.1.2 #define MAX_LENGTH 255

2.3.2 Function Documentation

2.3.2.1 void color_pair (WINDOW * *win*, int *color_bg*)

Color_pair - function that initializes the color palette. Specifies the color template used to decorate the windows.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.3.2.2 char** direct (char * *dp*, int * *count*)

Direct - the function reads the contents of the directory and stores it in an array (in lexicographical order).

Parameters

dp	- the parameter that stores the path of a directory whose contents need to be read
count	- parameter that is responsible for the number of files in the directory

Returns

an array of file names(filenames)

2.3.2.3 void** draw_menu (int *start_col*)

Draw_panel - creation of upper and lower panels

Parameters

start_col	- color panel parameter

Returns

EXIT_SUCCESS in case of success

2.3.2.4 void init_win ( )

Init_win - initializing library ncurses

2.3.2.5 void no_print_line (WINDOW * *win*, char ** *filenames*, int *count*)

No_print_line - the function displays a list of files without highlighting the user's position. Used for output in a passive window.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
filenames	- parameter that passes the elements of the array that are needed display
count	- number of array elements

2.3.2.6 void print (WINDOW * *win*, char ** *filenames*, int *count*, int *user_pos*, int *color_bg*)

Print - output files to the screen with the highlight of the position on which the user is located. Used for the active window.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
filenames	- tarameter that passes the elements of the array that are needed display
count	- number of array elements
user_pos	- the parameter responsible for the position in which the user is located
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.3.2.7 void window ( )

Window - the function is responsible for creating and drawing windows, and also calls the function to read directories, which displays all the files in it in the manager windows. Here you can also navigate the file manager (KEY_UP, KEY_DOWN, TAB, ENTER).

2.4 text_editor/func.c File Reference

```
#include <stdlib.h>
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <curses.h>
#include <stdio.h>
#include <malloc.h>
#include <dirent.h>
#include <string.h>
#include "func.h"
#include <unistd.h>
```

Macros

- #define _SVID_SOURCE

Functions

- void cycle (WINDOW *my_win, int user_pos_X, int user_pos_Y, int row, int col, char *filename, int count)
- void write_file (char *filename, char **string, int count, WINDOW *my_win)
- char ** read_file (char *filename, int *count)
- void color_pair (WINDOW *win, int color_bg)
- void window (char *filename)

2.4.1 Macro Definition Documentation

2.4.1.1 #define _SVID_SOURCE

2.4.2 Function Documentation

2.4.2.1 void color_pair (WINDOW * *win,* int *color_bg*)

Color_pair - function that initializes the color palette. Specifies the color template used to decorate the windows.

Parameters

win	- the parameter that passes a pointer to the window into which it is necessary list files
color_bg	- the parameter responsible for the highlight color of the position in which the user is located

2.4.2.2 void cycle (WINDOW * *my_win,* int *user_pos_X,* int *user_pos_Y,* int *row,* int *col,* char * *filename,* int *count*)

Cycle - performs processing of keystrokes.

Parameters

my_win	- The parameter that passes the pointer to the window. Needed in order to use the move cursor command
user_pos_X	- the parameter responsible for the position of the user in the rows
user_pos_Y	- the parameter responsible for the position of the user in the column
row	- parameter, which stores the number of rows
col	- parameter, which stores the number of columns
filename	- the parameter contains the name of the file that you want to open
count	- the parameter that counts the number of lines in a file

2.4.2.3 char** read_file (char * *filename,* int * *count*)

Read_file - the function reads each line of the file and writes it to an array of strings.

Parameters

filename	- the parameter contains the name of the file that you want to open
count	- the parameter that counts the number of lines in a file

Returns

an array of strings

2.4.2.4 void window (char * *filename*)

Window - creates a window in which the contents of the file are output

Parameters

filename	- the parameter contains the name of the file that you want to open

2.4.2.5 void write_file (char * *filename,* char ** *string,* int *count,* WINDOW * *my_win*)

Write_file - function allows you to enter words from the keyboard that will be written to a file

Parameters

filename	- the parameter contains the name of the file that you want to open
string	- array of new lines to be written to the file
count	- the parameter that counts the number of lines in a file

my_win	- The parameter that passes the pointer to the window. Needed in order to use the move cursor command

2.5 src/main.c File Reference

```
#include <stdlib.h>
#include <termios.h>
#include <sys/ioctl.h>
#include <signal.h>
#include <curses.h>
#include <stdio.h>
#include <malloc.h>
#include <dirent.h>
#include <string.h>
#include "../include/func.h"
```

Functions

- int main ()

2.5.1 Function Documentation

2.5.1.1 int main ( )

Author

E.Maklashkina

Version

1.0.0

Date

June,2017

Warning

For the text editor to work, you must specify the environment variable "export PATH=$PATH:$PWD" Main - the function of the main lines makes a call to the main part of the program - initializing and drawing windows, reading directories and files, displaying them on the screen, etc.

Returns

0 if the program finishes successfully

2.6 text_editor/main.c File Reference

```
#include <stdio.h>
```

Functions

- int main (int argc, char **argv)

2.6.1 Function Documentation

2.6.1.1 int main (int *argc,* char ** *argv*)

Main - takes the name of the file that needs to be opened and passed to the function Windows() 0 if the program finishes successfully

Index