Assembled Developer Manual

5/10/2023

Contents

This document contains the full documentation of all underlying functions within the Assembled text editor.

Source Code Directory Structure

The directory structure in implementation currently looks like the following list:

- src/
    - editor/
        - config.c
        - functions.c
        - keyboard.c
    - include/
        - editor/
            - config.h
            - functions.h
            - keyboard.h
        - interface/
            - screens/
                - start.h
            - interface.h
        - global.h
        - util.h
    - interface/
        - screens/
            - start.c
        - interface.c
    - main.c
    - util.c

The program can be split into two parts: update and render. The editor part defines functions which are used for the functionality of the program, and interface defines functions which are used to display information to the user.

An example of functionality is the keyboard. The user presses a key, which the program should interpret, this interpretation is handleld by *keyboard.c*.

Likewise, an example of rendering is the screen interface. There are various different screens which the user can access and the program needs to somehow manage which screen is being rendered and updated.

main.c is used for concatenating the two halves into one functioning program.

Editor Half

The editor is responsible for handling non-visual tasks.

Interface Half

The interface manages all possible screens which can be displayed to the user at any time.

interface/interface.c contains two functions: register_screen and switch_to_screen.

int register_screen(name, render, update, local)

Calculates the index of the screen within the *screens* hashmap, and sets the attributes of the screen structure at that index to the given arguments.

Returns the index of the screen in the hashmap.

name – The name of the screen, used as the key within the hashmap of screens

char *

render – A function pointer to the code which renders the screen to the TUI

void (*)(struct render_context *)

update – A function pointer to the code which updates the screen

void (*)(struct render_context *)

local – A function pointer for handling special cases

void (*)()

Special cases occur for events like navigation keys. When a key mapped to a function such as “up” is pressed, the “up” function is called. This function is then told to call into the local handler of the currently active screen with the code LOCAL_ARROW_UP.

int switch_to_screen(name)

Switches the *active_screen* pointer to point to the specified screen.

Returns the indedx of the screen in the hashmap.

name – The name of the screen to switch to

char *