Fractal OS

Technical Support Manual

Table of Contents

1. System Overview
2. System Structure
   1. UML Class Diagrams
   2. File Contents
3. Function Descriptions
4. Data Structure Descriptions
5. Cross References
6. Index

I. System Overview

Fractal is a pretty simple and easy to learn system. The primary elements include a class called “Command” and its methods, the main() function, and the secondary functions that execute in main. The system is set up so that anytime a new command needs to be created in Fractal, one can simply add a new method to the Command class and call it in the main program when needed. This keeps things clear and hard to confuse as there is only one object handling all of the command execution in the system.

II. System Structure

1. UML Diagrams
2. File Contents
   1. Main.cpp
      1. displayWelcome(): Displays Fractal’s welcome message.
      2. wait(int sec): Pauses system for ‘sec’ number of seconds.
      3. clearScreen(int delay): Waits ‘delay’ number of seconds and then clears the entire screen.
      4. runSession(bool session): Runs Fractal session.
      5. displayMainMenu(bool session): Displays main menu of choices to the user.
      6. string userChoice: Used to handle user input and all menu decisions.
   2. Command.h
      1. Contains “Command” class and all included libraries for system. See UML Diagram for Command.
      2. struct dirent: Holds information about a file directory.
      3. char date[]: Holds date values.

III. Function Descriptions

1. Command()
   1. Just the default constructor for the Command class.
2. void displayVersion()
   1. Displays the Fractal welcome message using cout.
3. void displayDate()
   1. Allocates a char array called ‘date’ and then calls \_strdate() which retrieves the system date. It then displays the date.
4. void displayHelp()
   1. Simply displays a list of help items using cout and endl.
5. void displayDirectory()
   1. Allocates a DIR pointer and dirent pointer to access directory data. Then attempts to open directory. If it fails, it returns and exits the method. If it succeeds, it runs a while loop that prints the name of each file using the d\_name part of the dirent struct. Then it closes the directory and returns.
6. void displayWelcome()
   1. Uses cout to display Fractal welcome message. Returns void.
7. void wait(int sec)
   1. Uses elements from ctime library to ‘wait’ for ‘sec’ number of seconds. Returns void.
8. void clearScreen(int delay)
   1. Waits ‘delay’ number of seconds and then clears the screen by outputting endlines. Returns void.
9. void runSession(bool session)
   1. Is responsible for running Fractal in a while loop until ‘session’ proves to be false. Returns void.
10. bool displayMainMenu(bool session)
    1. Takes a bool parameter called ‘session’. Alters session to false if the user exits Fractal. Handles output of menu display and handles user decisions with if statements. Also creates a Command object called ‘control’ and uses it to activate commands. Returns a bool.

IV. Data Structure Descriptions

1. userChoice
   1. String. Used to handle user keyboard input that is then processed to activate commands/decisions.
2. date[]
   1. Size 10 char array that holds the elements of a date.
3. dirent
   1. Struct used to hold information on directories. (e.g. name)

V. Cross References

1. displayWelcome()
   1. Called by: main()
2. wait()
   1. Called by: clearScreen()
   2. Calls: clock()
3. clearScreen()
   1. Called by: main(), displayMainMenu()
   2. Calls: wait()
4. runSession()
   1. Called by: main()
   2. Calls: displayMainMenu()
5. displayMainMenu()
   1. Called by: runSession()
   2. Calls: clearScreen(), Command.displayVersion(), Command.displayDate(), Command.displayDirectory(), Command.displayHelp(), and itself.
6. Command.displayVersion()
   1. Called by: displayMainMenu()
7. Command.displayDate()
   1. Called by: displayMainMenu()
   2. Calls: \_strdate()
8. Command.displayHelp()
   1. Called by: displayMainMenu()
9. Command.displayDirectory()
   1. Called by: displayMainMenu()
   2. Calls: opendir(), readdir(), closedir()

VI. Index

Command (class), II-i-1

Command.h, II-ii-2

Command(), III-i

displayVersion(), III-ii, V-vi

displayDate(), III-iii, V-vii

displayHelp(), III-iv, V-viii

displayDirectory(), III-v, V-ix

displayMainMenu(), II-ii-1-e, III-x, V-v

displayWelcome(), II-ii-1-a, III-vi, V-i

\*\*index is incomplete

Github link: <https://github.com/DaveyNicewander/OS_Simulator>