Документация на проект – Файлова система

Изготвил : Дейвид Барух 0MI0600128

1. Принцип на работа

Проектът използва IO monad, за да може да има обратна връзка между потребител и програмата. Въвежда се стринг от потребителя, който се разделя на елементи , където за разделител се приема whitespace. Елементите се вкарват в лист от тип String . Първият елемент e функцията, а останалите елементи са аргументи от нея. Със case of структурата се извършва при въвеждане на функция се извършва очакваното от нея. При дадена несъществуваща функция се издава съобщение за грешка, а при подадени само space-ове програмата не променя състоянието си. Всяка функция използва множество от helper функции , които спомагат за тяхното изпълнение, като при подадени невалидни аргументи също се издават съобщения за грешка.

1. Функционалност и структура

Файловата система се позовава на дефинира структура **FileSystem** която се състои от създаден тип данна **File** или данна **Directory**.

**File** има конструктор, приемащ параметрите name и content, които са от тип String.

**Directory** има конструктор, приемащ като параметри name, който е от тип String и лист от тип FileSystem.

Функциите дефинирани в проекта са **cd**,**pwd**,**ls**,**cat**,**rm** и допълнително добавени са **mkdir** и **tree**.

* 1. **cd** – тя може да работи без аргументи, като в този случай те отпраща в root директорията. Когато и се подадат аргументи се очаква път, който да е релативен или пълен. При невалиден път издава съобщение за грешка. При валиден път те отпраща в желаната от потребителя директория.
  2. **pwd** – тази функция работи без аргументи и извежда пътят на текущата директория.
  3. **ls** - тя може да работи без аргументи, като в този случай връща съдържанието на текущата директорията. Когато и се подадат аргументи се очаква път, който да е релативен или пълен. При невалиден път издава съобщение за грешка. При валиден път връща съдържанието на желаната от потребителя директория.
  4. **cat** – тази фукция изисква като аргумент/аргументи файлове с релативни или пълни пътища като издава нейното/тяхното съдържание. При съдържане на символа „>“тя записва съдържание в нов файл подаден от дясната страна на символа „>“. Този изходен файл може да бъде с релативен или пълен път. От лявата страна на символа „>“ може да се подадат аргументи или не. При налични аргументи, които са файлове , тяхното съдържание се записва в изходния файл. При неналични аргументи потребителят въвежда съдържанието на файла до поставяне на символ „.“ на нов ред.
  5. **rm** – тази функция приема множество от файлове които се трият от файловата система. Допълнително е включен и аргумент „**-d**“,с който се указва , че ще се трият директории.
  6. **mkdir** – създава нова директория при подаден път.
  7. **tree** – показва съдържанието на текущата директория и нейните поддиректории при неподадени аргументи. При подаден аргумент път към директория показва съдържанието на тази директория и нейните поддиректории.

1. Инструкции за работа

Използване на „ghci“. Въвеждане на „ : main “, след което се стартира файловата система и може да се поддават функции и аргументи. За приключване на работа с програмата се въвежда „**exit**“.