



Софийски университет "Св. Климент Охридски"
Факултет по математика и информатика

ПРОЕКТ

по

Функционално програмиране

спец. Информатика, 3 курс, зимен семестър,

учебна година 2023/2024

Тема: Файлова система

Изготвил: Анастасия Петрова

Фак. №: 45746

I. Основна идея на проекта:

Идеята на проекта е да се реализира проста файлова система. За нейната реализацията в програмата е използвано дърво. Също така за запис на промените по нея при затваряне на приложението, се използва специален файл `fileSystem.txt`, който също се използва и за нейното зареждане при стартиране на програмата.

Реализираните команди са следните:

- `pwd` – извежда пълния път на текущата директория
- `cd` – променя текущата директория
- `ls` – извежда съдържанието на директорията
- `cat` – конкатенира съдържанието на файловете
- `rm` – изтрива файлове
- `mkdir` – създава директория ако подадения път до нея съществува във файловата система
- `exit` – излиза от приложението

Горепосочените команди няма значение дали се изписват с главни или малки букви в приложението.

II. Как е реализирана идеята:

За целта е създаден собствен тип `FileSystem`, който ще ни представя файловата система. Той реално представлява дърво, дефинирано рекурсивно, където чрез:

`Empty` - представяме, че няма нищо (също се използва и за изтриване) ;

`File` – представяме файл;

`Directory` – представяме директория, която съдържа в себе си списък от дървета, които ще ни бъдат подпапките и файловете. Директориите и файловете също така имат `String`, който ще бъде тяхното име.

Текущата директория, в която се намира потребителя по време на работа, е представена като списък от имена на папки. По същия начин ще се представя кой да е път в проекта.

III. Функции:

Първо ще разгледаме някои основни функции от модула `FileSystem` (идеята му е да се реализира дървото за файловата система):

- Дефинираният тип `FileSystem` представлява дървото на файловата система в проекта.
- `getRootValue` – приема като аргумент дърво и ни връща стойността на корена му.

- `getChildrenList` – приема като аргумент дърво и връща списък от поддървета (децата на корена).
- `getAllInDir` – приема дърво и връща стринг от имената на децата на корена на дървото (ще се използва за командата `ls`).
- `extractSubtree` – приема дърво, лист от децата на корена, и лист от `String`, който ще ни представя път по дървото и ни връща поддърво с корен последен елемент от пътя.
- `insertSubTree` – приема две дървета, където първото ще ни представя файловата система, а второто ще ни бъде директория или файл, който искаме да добавим. Третия аргумент е пътя по дървото къде точно да добавим файла/директорията.
- `removeNodes` – приема дърво и път и връща файлова система, в която е премахнат файла, указан ни чрез пътя.

Основните функции от модула `Commands` са следните (идеята му е да се реализират командите на файловата система и чрез тях работата с нея):

- `changelist` – приема командата, дърво, аргументите на командата, текущата директория и връща нова директория (къде се намира потребителя) ако е необходимо да се смени иначе връща старата.
- `changeFS` – аналогично на горната функция само че връща променена файлова система ако е необходимо
- `pwd`, `ls`, `cat`, `rm` и `mkdir` – правят същото, което е написано в началото за командите, като приемат за аргументи : аргументите на командата (представени са чрез списъци от списъци от низове, тъй като пътищата в проекта са представени навсякъде като списъци от низове), текущата директория и дървото на файловата система.
- `cd` – идеята на тази функция е същата като описанието на същата командата, но връща наредена двойка от булева стойност дали е била успешна и път от новата директория (ако е била сменена)
- `executeCommand` – приема низ, който ни представлява командата, която потребителя е подал, дърво, лист от аргументите на командата и текуща директория. Функцията просто изпълнява командата на потребителя
- `enterComand` – приема дърво и текущата директория. Функцията приема командите на потребителя от терминала и ги преобразува в нужния формат за програмата, и извиква с тях `executeCommand`. След като `executeCommand` приключи чака потребителя да въведе следващата команда.

Някой от функциите от модула `HelperFunctions` (идеята му е да спомогне работата на модула `Commands`)

- `getWordsList` – превръща подадения път от потребителя в подходящ формат за програмата
- `getFullPath` – приема два пътя, като първия е подаден от потребителя, а втория е текущата ни директория. Връща ни пълния път на пътя подаден от потребителя.
- `convertToActualPaths` – приема списък от пътища и връща списък от низове, които представляват реалните пътища на локалната машина.
- `getInfoFromFilesInFile` – приема списък от низове и файл. Записва елементите на списъка във файла.
- `removeActualFiles` – приема списък от низове, който представляват различни пътища на файлове на локалната машина и ги изтрива

Във файла `source.hs` ни се изпълнява функцията `load`, която ни зарежда всичко необходимо, за да работим с нея.

IV. Как се стартира:

За да стартирате приложението, е необходимо да се стартира терминала на VS Code и да се намерите в папката на проекта. В терминала се пише командата `ghci` и `source.hs` и се изчаква да се заредят модулите. След това се пише `main` и вече можем да използваме приложението.

V. Упътване:

Във файла `fileSystem.txt` е зададена празна файлова система. Този файл служи единствено и само за работа с програмата, за да може, когато се излезе от приложението да се запазят промените по файловата система. Когато започнат да се създават папки и файлове, корена на дървото ще бъде директорията, от която сме пуснали файла.