**Име:** Тихомир Каменов, **ФН:** 45431, **курс:** 3, **група:** 1, **специалност:** “Информатика”

# “Файлова система”
# Проект за курса Функционално програмиране 2019/2020г.
# Документация

Проектът е писан на Haskell.

За целите на проекта са създадени нови типове:
- RegularFile, съдържащ името и съдържанието на обикновен файл
- Dir, съдържащ името и съдържанието (директории и/или обикновени файлове) на директория
- FileSystem – представлява дърво, чиито “листа” са обикновени файлове или празни директории
- Command – съдържа основните команди със съответните допустими аргументи

Също така за улеснение е създаден синоним на String под името Path, за да е по-ясно от сигнатурата на дадена функция дали се разглежда път до директория/файл, съобщение за грешка или нещо друго.

Проектът съдържа следната функционалност:
- main функция – В началото се “зарежда” файловата система, като ако има съществуващ файл filesystem.txt в директорията, в която е hs файлът, файловата система се зарежда от него, в противен случай се създава празна файлова система, съдържаща единствено root директория. След това се създава наредена двойка от текущата директория (в началото това е root директорията на файловата система) и файловата система. Извършват се евентуални модификации на файловата система, като при приключване на работата с нея получаваме нова файлова система fs’, която записваме във файл filesystem.txt.
- parseCommand – при въвеждане на команда от конзолата тя се обработва чрез тази функция, при което ако е валидна команда, връща командата, в противен случай връща съобщение за грешка.
- loop – интерактивната част се извършва тук. При всяко извикване се въвежда някакъв вход, който се парсва към команда. Ако е валидна команда се изпълнява, като ако се въведе exit, приключва извикването на loop функцията и програмата приключва работа. В противен случай (невалидна команда) се въвежда съобщение за грешка, което се е получило от parseCommand.
- executeCommand – приема текуща директория и файлова система, в зависимост от командата ги модифицира и връща нови текуща директория и файлова система.
- Search – приема файлова система, абсолютния път на файл/директория, списък от абсолютния път (разбит е спрямо разделителя “/”), функция, която извършва съответна операция върху файловата система (извикана от някоя команда) и текущия файл/директория, който разглеждаме. Функцията връща наредена двойка от резултата от извикването на функцията и евентуално модифицираната файлова система. Идеята на search функцията е че обхожда файловата система спрямо списъкът [Path], представляващ абсолютния път на файла/директорията, като ако по някое време на обхождането попаднем на файл или директория от [Path], която не съществува, ще се извика search функцията с последен аргумент Nothing, при което ще се изведе подходящо съобщение за грешка на конзолата. В противен случай ако файлът/директорията съществува, ще се извърши съответната операция като накрая върнем резултата от операцията и новата файлова система.
- pwd, ls, cd, getContent, rm, checkIfExists са функции, които се подават на search, ако търсеният файл/директория съществува.
- modifyFileSystem – функция, която приема файлова система, “съдържание” content, абсолютен път на файл, разбит отново спрямо “/”, и функция g, която модифицира файла. Идеята е, че обхожда дървото и го копира, докато не стигне до търсения файл, при което го връща модифициран. Функцията се използва, за да се премахне файл от системата (при което функцията g не добавя търсения файл към системата) и когато се извиква cat командата с пренасочен изход (при което функцията g добавя търсения файл към системата със съдържанието content).
- catToFile и (вътрешната функция deleteFile на rm) – функциите, които се подават на modifyFileSystem.

**Пример:** В рамките на проекта съм работил със следната примерна файлова система: root директорията съдържа обикновен файл file2.txt със съдържание “Bye!” и директория home. Директория home съдържа обикновен файл file1.txt със съдържание “Hello” и директория etc. Директория etc съдържа обикновен файл file2.txt със съдържание “Bye!”.

**Примерни изпълнения на командите:**

- ls

```
/$ ls
home  file2.txt
/$ ls home
file1.txt  etc
/$ ls home/etc
file2.txt
```

- cd

```
/$ cd home
/home/$ ls
file1.txt  etc
/home/$ cd ..
/$ ls
home  file2.txt
/$
```

- rm

```
/$ rm file2.txt
/$ ls
home
/$
```

- cat

```
/$ ls
home  file2.txt  file3.txt
/$ cat file2.txt
Bye!
/$ cat file3.txt
hello
bye!
/$ cat file2.txt file3.txt > home/file5.txt
/$ cat home/file5.txt
Bye!
hello
bye!
/$ ls home
file1.txt  etc  file5.txt
/$
```