# Файлова система

Автор: Мартин Георгиев Маринов
Специалност: КН
Група: 2
Факултетен номер: 0MI0800052

Информация за програмата:

Тази програма симулира файловата система. Симулацията на системата се прочита от файла tree.txt. Примерен файл съм оставил в пакета. Основните команди за работа са: pwd, ls, cd, cat, mkfile, rm, save. Описание на командите:

- pwd – Извежда пътя до текущата директория
- ls – Зададена без аргументи извежда съдържанието на текущата директория, има един опционален аргумент път, който се използва, ако искаме да изведем съдържанието на директорията указана от този път.
- mkfile – тази команда приема 2 аргумента: име и съдържание на файла и създава нов файл с тези свойства в текущата директория.
- rm – Премахване на файлове указани като параметри
- save – Запазва състоянието на файловата система във файла tree.txt
- cd – Сменя директорията. На тази команда може да се подаде като аргумент релативен и абсолютен път към директорията, в която искаме да отидем.
- cat – Използва се за конкатенация на файлове, чиито пътища са подадени като аргументи. Ако няма указано > file резултата се извежда на екрана, ако има резултата се записва в файл с име дадено в file в директорията в която сме в момента.

Информация за реализацията:

Файловата система се симулира с дърво, което трябва да се променя като се добавят и премахват нови върхове. За целта съм използвал свое представяне на дърво чрез 3 списъка: ParrentList, AdjacencyList и IntToNodeMap. Това представяне използва факта, че дървото е граф и едно класическо представяне на графите, а именно списък на съседство. На всеки връх в дървото се дава уникално ID и двойките връх във файловата система, ID се съхраняват в списъка IntToNodeMap. Съхранявам и двойките връх, родител в списъка ParrentList и реализацията на списъка на съседство е AdjacencyList в който за всеки връх има оказано ID и списък от ID-та на наследниците му. Предимствата на тази реализация са, че вместо да ъпдейтваме всичко до корена и после децата, е нужно само да ъпдейтнем 3-те списъка и AdjacencyList-а на съответния родител за да добавим или премахнем запис.

Серилизация на данните:

Данните се съхраняват във файла tree.txt по следния начин:

- 1 ред – IntToNodeMap – Всеки запис отговаря на 3 или 4 елемента в зависимост от типа си. Ако е директория - на 3, ако е файл – на 4. Структурата е id type name (content ако нещото е файл). Редът съдържа информация за всички записи в списъка.
- 2 ред – ParrentList – Всеки запис отговаря на наредена двойка връх, родител.
- 3 ред – AdjacencyList – Всеки запис отговаря на наредената двойка ID, списък на съседство.

Информацията от файла се прочита и се преобразува до съответните списъци чрез функциите: getIntToNodeMapFromString, getParrentListFromString и getALFromString. След това се подават на слушащата функция за команди. Слушането и изпълняването на команди е реализирано чрез непряка рекурсия

между функциите inputCommand и processCommand. Те си подават взаимно списъците и промените им. Всяка функция която оперира с даден списък го приема като параметър от processCommand или от съответната извикваща функция, която е била извикана от processCommand. След всяка операция на функцията inputCommand се подават като параметри най-новите версии на списъците и така са в сила ъпдейтите. Командата save серилизира и записва тези списъци във файла tree.txt в указания по-горе ред.