**Файлова подсистема**

Целта на този проект е да се имплементира подсистема, представляваща опростена версия на UNIX файлова система, заедно с приложение - *терминал*, предоставящо набор от прости команди за работа с нея.

**1. Файлова Система**

За целите на задачата ще трябва да реализирате проста файлова система. Тя представлява множество от директории и файлове. Всяка директория може да е празна, а може и да съдържа други директории и/или файлове. Файловата система има начална директория, наречена **root.**

Директориите и файловете разполагат задължително със следните атрибути:

* name - име на директорията/файла.
* creationDate - дата на създаване
* modificationDate - дата на последна модификация

От всяка директория трябва да може да се достигне до нейното съдържание (поддиректории и файлове), както и до нейната родителска директория.

Системата трябва да работи поне със следните три типа файлове:

* Текстови файл (.txt) - Файл, съдържащ данни в четим за хората вид (human-readable).
* Скриптов файл (.sh) - Файл, в който са записани последователности от команди.
* Връзка (.lnk) - Файл-връзка, чието съдържание е път към **друг файл** от файловата система, като той може да бъде абсолютен (absolute) или релативен (relative).

**2. Команден Интерфейс**

Нашата операционна система се нуждае и от терминал, който позволява работата с файловата система. Трябва да поддържате следният набор от команди:

* **cd <път>** - променя активната директория спрямо подадения път. Той може да бъде релативен или абсолютен, като при невалиден път се извежда подходящо съобщение за грешка на потребителя.
* **ls** - извежда имената на всички файлове и директории в текущата директория. Извеждането се състои в представяне на информация за името на файловете и директориите в текущата работна директория, информация за датата на създаване и за датата на модификация.
* **ls <път към директория>** - Извежда съдържанието на подадената директория.
* **pwd** - изписва абсолютния път на текущата работна директория.
* **mkdir <име на директория>** - създава нова поддиректория в текущата работна директория. Ако директория или файл вече съществува с това име, командата връща подходящо съобщение за грешка.
* **touch <име на файл>** - създава нов файл в текущата работна директория. Ако името на файла завършва на “.sh”, то той ще се третира като shell script (скриптов файл). Ако името на файла завършва на “.lnk”, то той представлява link. Ако разширението е “.txt” или липсва, файлът е обикновен текстов файл. Ако директория или файл вече съществува с това име, променя датата на последна промяна.
* **rm <път до файл>** - изтрива посочения файл. Ако такъв файл не съществува, командата връща подходящо съобщение за грешка.
* **rmdir <път до директория>** - изтрива посочената поддиректория. Ако не съществува, командата връща подходящо съобщение за грешка. Ако директорията съдържа файлове в себе си, командата връща подходящо съобщение за грешка.
* **exec <път до файл>** - изпълнява посочения файл в зависимост от неговия тип:
  + Текстов файл - извежда се съдържанието му на конзолата.
  + Shell script - изпълняват се последователно всяка команда в съдържанието на файла ред по ред.
  + Link - изпълнението се извършва върху файла, към който сочи връзката. Ако връзката е към директория се извежда подходящо съобщение за грешка.
  + При опит за изпълнение на директория или несъществуващ файл се извежда съобщение за грешка.
* **find <път> <низ за търсене>** - извежда имената и пътищата до всички намерени файлове и директории, които имат подниз подадения низ, започвайки търсенето от посочения начален път. При невалиден стартов път за търсене (директорията не е открита или подаденият път сочи към файл, а не директория) се извежда съобщение за грешка.
* **echo <text>** - Извежда текста на стандартния изход.
* **echo <text> > <file>** - Презаписва текста в подаден файл.
* **echo <text> >> <file>** - Добавя текста в края на файла.
* **exit** - запазва извършените промени върху файловата система и затваря приложението.

При изход от терминала, програмата трябва да записва информацията за цялата файлова система в един или повече файловев подходящ формат. След повторно стартиране на приложението, информацията за цялата файлова система трябва да бъде възстановена, като за текуща работна директория се задава последната работна директория преди изхода от приложението.

**Бонус**: може да имплементирате всякакви други команди, типове файлове и функционалности (например: копиране, местене, потребители, групи, роли, file permissions и други), както и да разширите функционалностите на тези, изброени в условието на проекта (например: добавяне на флагове и допълнителни настройки към командите).

**Бележка:** Директорията .. се счита за родителската директория на текущата работна директория. Командата cd .. ще се възприеме като cd <parent working directory>. Командата ls .. ще изведе съдържанието на родителската директория.

**3. Примерен интерфейс**

root#

root# mkdir user1

root# cd user1/

root/user1#

root/user1# mkdir temp

root/user1# cd temp

root/user1/temp# pwd

root/user1/temp

root/user1/temp# cd ..

root/user1# pwd

root/user1

root/user1# touch file1.txt

root/user1# ls

May 19 14:07 file1.txt

May 19 14:07 temp

root/user1# touch file2.sh

root/user1# ls

May 19 14:07 file1.txt

May 19 14:08 file2.sh

May 19 14:07 temp

root/user1# rm file1.txt

root/user1# ls

May 19 14:08 file2.sh

May 19 14:07 temp

root/user1# rmdir temp/

root/user1# ls

May 19 14:08 file2.sh

root/user1# echo hello

hello

root/user1# echo ls > file2.sh

root/user1# exec file2.sh

May 19 14:09 file2.sh

root/user1# touch file.txt

root/user1# echo this is text > file.txt

root/user1# exec file.txt

this is text

root/user1# cd ..

root# mkdir test\_dir

root# cd test\_dir

root/test\_dir# touch hello.sh

root/test\_dir# echo "echo hello links" > hello.sh

root/test\_dir# cd ..

root# cd user1/

root/user1# touch link.lnk

root/user1# echo ../test\_dir/hello.sh > link.lnk

root/user1# exec link.lnk

hello links