Проект по Структури от данни и програмиране

(спец. Компютърни науки, 2015/16 г.)

Файлова система

Да се напише програма, която реализира Unix-подобна файлова система, и позволява работа с нея в интерактивен режим.

Файловите системи служат за организация на данни в именувани блокове данни, наречени файлове. Файловете от своя страна са организирани в йерархична дървовидна структура чрез използването на директории.

В Unix-базираните системи, коренът на файловата система се означава с `/`, като същият символ се използва и за разделител на директории в път към файл. Ако един път започва с `/`, то той е абсолютен, иначе е относителен (спрямо текущата директория).

В една Unix-подобна файлова система “всичко е файл”, т.е. не се прави разлика между различните видове файлове. Директориите са файлове, *символичните връзки* (файлове, съдържащи път към друг файл) също. Файловата система трябва да поддържа както обикновени файлове и директории, така и символични връзки (*symlink*-ове), като за символичните връзки операциите върху файловете трябва да действат върху файла, към който сочи връзката.

Освен това за всеки файл трябва да се съхраняват следните *метаданни* (допълнителна информация за самия файл):

- пореден номер на файла във файловата система
- размер на файла
- вид на файла
- време на последен достъп
- време на последна промяна на съдържанието
- време на последна промяна на метаданните
- (*част от бонуса*) брой на твърдите връзки към файла

Целта на проекта е да се реализира програма, която записва информацията за файловата система в подходящо избрана структура (в паметта или във файл). Програмата да предлага конзолен интерфейс за работа със системата, като поддържа командите:

- `pwd` — извежда пълния път до текущата директория (в началото `/`)
- `cd` — променя текущата директория
  - синтаксисът е **`cd`** <директория>
  - `cd /full/path` сменя текущата директория с тази, зададена като параметър
  - `cd relative/path` сменя текущата директория с тази, зададена с относителен път относно текущата директория
  - `.` означава текущата директория, например `cd foo/./bar` е еквивалентно на `cd foo/bar`, а `cd .` не прави нищо
  - `..` означава родителската директория, например `cd foo/../bar/../foo/../bar` е еквивалентно на `cd bar`
- `ls` — извежда съдържанието на директория
  - синтаксисът е **`ls`** <директория>
  - `ls` — извежда съдържанието на текущата директория
  - `ls relative/path` извежда съдържанието на директория, зададена с относителен път относно текущата
  - `ls /full/path` извежда съдържанието на директория, зададена с абсолютен път
- `cat` — конкатенира съдържанието на файл(ове)
  - общият синтаксис е **`cat`** <файл$_1$> <файл$_2$> … <файл$_n$> **`>`** <изходен-файл>
  - <файл$_1$> <файл$_2$> … <файл$_n$> са относителни или абсолютни пътища
  - <изходен-файл> е файл, в който се записва резултатът
  - съдържанията на <файл$_1$> <файл$_2$> … <файл$_n$> се слепват в реда, в който са зададени пътищата на файловете, и резултатът се извежда в изходен файл
  - ако има един <файл>, той просто се извежда
  - ако няма нито един <файл>, съдържанието се въвежда от стандартния вход, като въвеждането приключва при въвеждане на знак точка самостоятелно на ред
  - ако няма <изходен-файл>, съдържанието се извежда на стандартния изход
- `cp` — копиране на файл(ове)
  - синтаксисът е **`cp`** <файл$_1$> <файл$_2$> … <файл$_n$> <директория>
  - копира всичките подадени файлове в <директория>
- `rm` — изтриване на файл(ове)
  - синтаксисът е **`rm`** <файл$_1$> <файл$_2$> … <файл$_n$>
  - изтрива всичките файлове, подадени като аргументи
- `mkdir` — създаване на директория(и)

    - синтаксисът е **mkdir** <директория$_1$> <директория$_2$> … <директория$_n$>
    - създава всичките подадени директории
- `rmdir` — изтриване на директория(и)
    - синтаксисът е **rmdir** <директория$_1$> <директория$_2$> … <директория$_n$>
    - изтрива всичките подадени директории
- ln — създава връзки към файл(ове)
    - синтаксисът е **ln -s** <файл$_1$> <файл$_2$> … <файл$_n$> <директория>
    - създава връзки към всички подадени файлове в <директория>
    - (*част от бонуса*) ако има `-s`, връзите са символични, иначе са твърди
- `stat` — извежда метаданните за файл(ове)
    - синтаксисът е **stat** <файл$_1$> <файл$_2$> … <файл$_n$>

Пример

```
$ pwd
/
$ ls
folder1 folder2 file1 file2
$ cd folder1
$ pwd
/folder1
$ cd ..
$ rm file2
$ ls
folder1 folder2 file1
$ cat > file1
This is the content for file1
.
$ cat file1
This is the content for file1
$ cat file1 > file3
$ cat file3
This is the content for file1
$ cat file1 file3 > file4
$ cat file4
This is the content for file1
This is the content for file1
```

Бонуси

- да се реализира поддръжка за някои от другите типове файлове в Unix (*named pipe*, *socket*, *character*/*block device file*)
- да се реализира поддръжка за множество *hard link*-ове към един файл (т.е. повече от един път да сочи към един и същ файл, за разлика от symlink-овете, които са отделни файлове, които сочат към други файлове)
- да се използват за метаданни всички полета от POSIX *stat* структурата
- да се добави поддръжка за монтиране и размонтиране на файлови системи (цялото дърво на монтирана файлова система върху даден път в друга (носеща) файлова система може да се достъпи от носещата файлова система. Пътищата на файловете вече започват от точката на монтиране, като пътят към корената на монтираната файлова система е същият като пътя на монтиране.)
- да се имплементира VFS (като този в Linux)
- да се имплементират някои от функциите в POSIX за работа с файловата система

Насоки

- https://en.wikipedia.org/wiki/Unix_filesystem
- https://en.wikipedia.org/wiki/Unix_file_types
- ако решите да имплементирате линукския VFS или POSIX функциите:
  - https://en.wikipedia.org/wiki/Virtual_file_system
  - Един PDF с малко повече информация и източници за Unix-like файловите системи